Can News Headlines Predict Stock Market Volatility?

Max Sterman
Department of Computer Science
New York University
New York City, USA
ms8705@nyu.edu

Abstract: I present a study on the effects of news headline sentiment on stock market volatility. Stock market volatility is the variance of stocks or an index. The VIX, the Chicago Board Option Exchange (CBOE) Volatility Index, is a popular index for expected volatility. In this study, I examine if news headline sentiment can predict the VIX. Unlike previous studies, I try to predict the intraday changes of the VIX. I perform sentiment analysis on these news headlines using two different Python packages, TextBlob and Vader Sentiment. The resulting sentiment scores are inputs to the predictive models. To gather the articles, I use a Python package, News API, which provides metadata on the articles. Then I evaluated the sentiment scores using Pearson Correlation, Granger Causality, ARIMAX, and Recurrent Neural Networks. Results from Granger Causality tests show that Vader's neutral sentiment provides some predictive power. The ARIMAX modeling shows that the best predictive sentiment had 57 percent directional accuracy, 5 points higher than the best predictive AR VIX Model. The Recurrent Neural Networks of lags of sentiment and VIX have 54 percent accuracy, 2 percent better than a Recurrent Neural Network of just VIX lags.

I. INTRODUCTION

In the 21st century, many computer scientists have examined how text sentiment can affect various financial stock market and economic indicators. Researchers have rationalized such insights as information necessary for the Efficient Market Hypothesis (EMH). [3] The EMH states that investors "rationally" want to maximize their profits and will extract all possible publicly available information [3]. News, especially about the stocks, themselves, are considered to be such information. In the last decade, many papers have explored how sentiment from different sources of text can affect the stock market. Specifically, Twitter has become a common source for similar projects, such as those of Arias and Arriata and Bollen, Counts, and Zheng, [1][10] to examine market data.

While tweets can provide real-time reactions to important events, they are not the primary source of information, but rather the response to it. News articles, especially those from top news organizations, are. In addition, many news organizations write headlines with a lot of emotion in an attempt to garner readers' attention.

In this paper, I test whether or not news headlines can have a significant impact in predicting the Chicago Board Option Exchange's Volatility Index (VIX). The VIX is a significant topic of interest as it provides the predicted spread of the market over the following 30 days. [2] Using the analogy of a probabilistic distribution, if a stock market index could be thought of the mean value of stock prices, the VIX can be thought of as the variance.

I use Pearson Correlation Analysis, Granger Causality, ARIMAX, and Recurrent Neural Network models to test for both linear and non-linear relationships between news headline sentiment and the VIX.

II. RELATED WORK

John Bollen, Huino Mao, and Xao-Jun Zeng's "Twitter mood Predicts the Stock Market" provided an innovative result of how twitter sentiment could affect the results of the stock market [1]. They collected tweets from February to December 2008, filtering out only tweets that exhibited emotion (including phrases such as "I feel" and "I don't feel"). They provided sentiment analysis on these tweets using two tools Opinion Finder (OF) and Google-Profile of Moods States (GPOMS). OF is a sentiment analyzer that predicts the polarity of the sentiment, negative or positive. GPOMS was a sentiment analyzer that provided scores for six moods: Calm, Alert, Sure, Vital, Kind, and Happy. The study used Granger Causality to check if augmenting the sentiment scores to an autoregressive (AR) model had a more significant effect than the non-augmented AR model. To address non-linear results, the authors used a Self-Organizing Fuzzy Neural Network (SOFNN). The Granger Causality test showed that there was a significant effect of "Calm" twitter sentiment on Dow Jones returns. The SOFNN proved a non-linear effect between this mood and the Dow Jones [1]. There were a couple of limitations to their approach. First, the test set they used only accumulated for 15 of the 207 days, about 7 percent of their combined training and test data set. While the SOFNN with Dow Jones lags and Calm lags outperformed a SOFNN with just Dow Jones lags by about 14 percentage points (87 percent vs 73 percentage points), only two more days were predicted correctly. Secondly, they used Granger Causality as feature selection on the SOFNN. However, Granger Causality predicts correlation on linear models, while a SOFNN is a non-linear model, hence disregarding variables that may not have a linear relationship with the VIX, but could have a non-linear relationship.

In Forecasting with Twitter Data, Arias, Arratia, and Xuriguera also use twitter data to forecast the Stock market and a volatility measure similar to the VIX, known as the VXO [10]. In this model, not only do they find keywords, but they also train a sentiment analyzer using a Naive Bayes classifier to creating sentiment indices. Just like "Twitter Mood Predicts the Stock Market", they use Granger Causality to find linear relationships between stock market mood and neural networks and support vector machines to find non-linear relationships between those two variables. They found that the non-linear methods above did the best in predicting the VXO.

Bollen and Mao also wrote a follow up paper with Scott Counts of Microsoft Research to explore sentiment of tweets, Google searches, investor surveys, and above all, news headlines [3]. Furthermore, they not only attempted to predict Dow Jones returns, but also, gold prices, trading volumes, and the VIX. The tweets and Google terms contained words from a pre-defined list of financial topics, while the news headlines were from 9 different financial news sources. The paper examines the correlations between both the sentiment scores and the financial indicators themselves. They use both the Pearson Correlation Coefficient and Granger Causality tests to examine correlation. From there, they calculate the forecast errors of a multiple regression ARIMAX model. Overall, they find that google searches, twitter sentiment, and negative financial news sentiment significantly affect the returns of the VIX.

While many of these sources limited their search to financial topics, I do not limit my search to these topics. As Mao, Counts, and Bollen note, social mood can affect people's financial behavior. In addition, all these studies calculated returns of daily data for each of their financial instruments. I take a look at intraday (30 minute) changes in the VIX.

III. Business Understanding

A. VIX

The VIX is a measure of implied volatility derived by the CBOE. It measures the predicted spread in the stock market over the next 30 days [2]. This measure has often been associated with market risk and, thus, has been commonly referred to as the "fear index" [3].

The VIX is one of the common indicators of volatility and has been used in similar studies such as that of Mao, Counts, and Bollen. The VIX is calculated using all options on the S&P 500 with realizations in the upcoming 30 days as its input to a mathematical function of implied volatility. Because of this, the VIX has been found to be (negatively) correlated with S&P 500 returns [3].

Since the time period is limited, I will use intraday data on the VIX to generate more observations. With the Alpha Vantage package, I gather VIX data on 30 minute intervals. Like much economic and financial literature [3], I calculate the VIX using its log growth from the same time a workday earlier:

$$\%\,\Delta\left(VIX_t\right) = \log\left(VIX_t\right) - \log\left(VIX_{t-1\,day}\right)$$

The purpose of the daily growth is to smooth out the series, such that small spikes and drops in 30 minute periods do not affect the predictability of my results.

IV. Data Understanding

A. Data Sources

To gather the news articles for this study, I use the News API Python package [7]. The News API Python package allows for 250 queries (API call) per six hours. It also allows for more in-depth search categories such as keywords, sources, and the dates. Articles can be queried for any date or time up to 30 days before the day of the query. In my searches, I queried the top 100 headlines (maximum number of articles per query) from all general and business sources, as categorized by News API, that are in the United States and have an English title for each source 12:01 AM-4:30 PM of that day. This accounts for all articles that could affect VIX trading at the opening of the business day (9:30 AM) and accounted for all public information released during the day [2]. Many of these news sources did not have 100 articles for that period, but for those which did, I sorted them by popularity, another parameter in the News API query. I decided not to query by any keyword and, instead, using the articles from the query, I find the top topics for the time period used by performing NLP methods learned in class.

In a News API query, metadata is provided in a Python dictionary with the following attributes: Title, Source, URL, Description, Published Date, and a snippet of the article body [7]. From this, I pull the title, source, description, date, and URL. From there I append it to a large database of all the articles that I have collected. I have collected over 80,000 news headlines from September 20th to November 21st. While News API provides a lot of article metadata, the API only provides articles up to 1 month before the date of the query. As a result, I was only able to start my experiment with data from September 20th to November 14th. To compensate for the lack of daily data, I aggregated news articles that occurred over a 30 minute period. The one exception is that all articles occurring before 9:30 AM are aggregated as one to account for all important news information that could be important in trading at the beginning of the day.

B. Sentiment Analyzers:

To calculate the sentiment scores, I used two Python sentiment packages: TextBlob and Vader Sentiment.

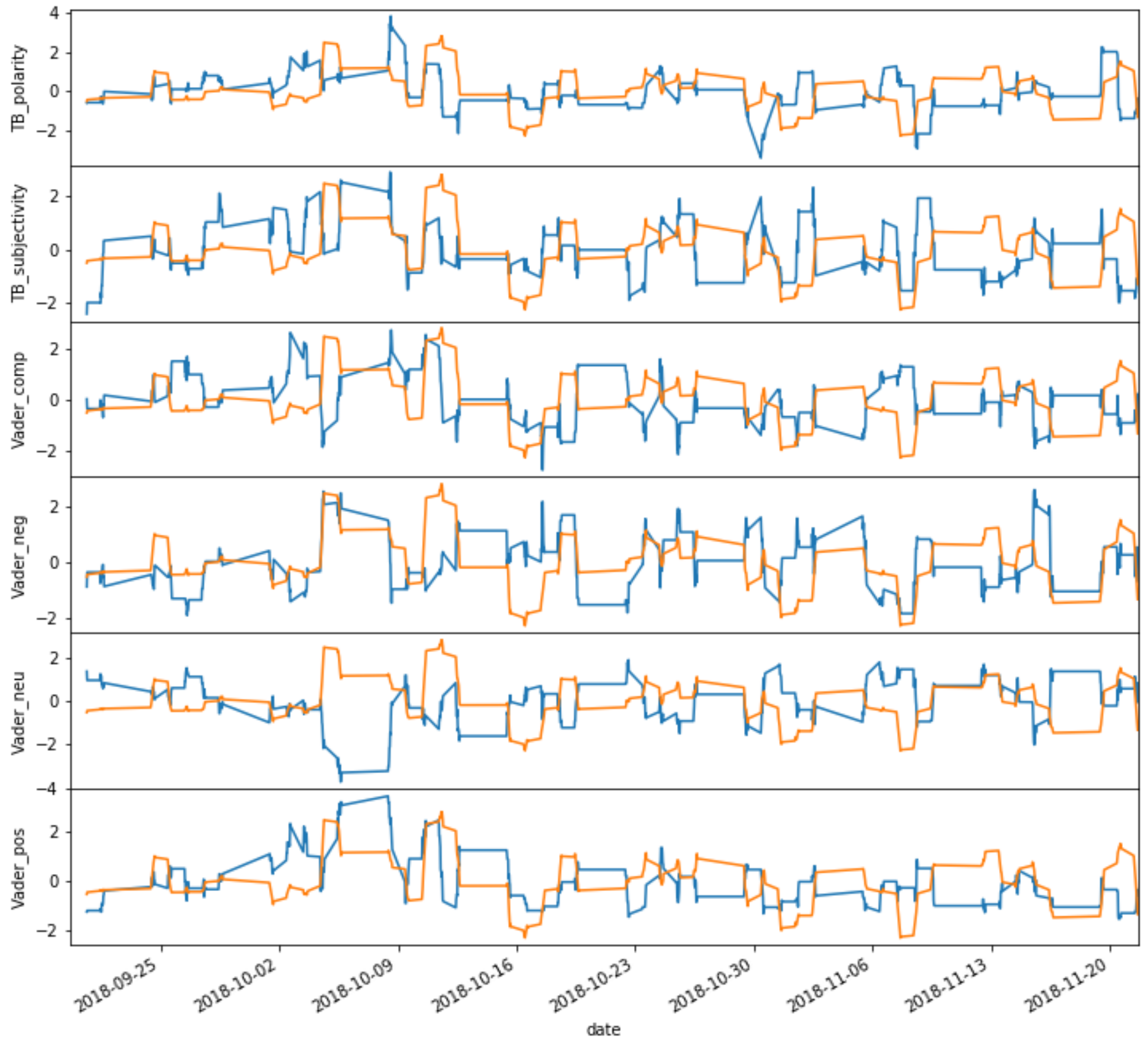

Fig. 1 Time Series of VIX and Sentiment. Sentiment Score in Blue, VIX In Orange. Normalized with Z-Score

TextBlob is an NLP Package licensed by the Massachusetts Institute of Technology (MIT) [4]. As Bari and Saatcioglu write, TextBlob is a lexical-based analyzer that takes into word association rules and context. From there, the analyzer will then calculate a score for each word and divide each word [5].

The package includes a sentiment analyzer that provides two scores: polarity and subjectivity. Polarity is a general strength of how positive or negative a scoring system is. The score has a range from -1 to 1, with a negative score inferring negative sentiment and a positive score inferring positive sentiment. The subjectivity score ranges from 0 to 1 with 0 being very objective and 1 being very subjective [4]. Subjectivity can be a very critical score as many of the sources I use have a partisan lean, such as conservative ones from Fox News and Breitbart and liberal ones from MSNBC and the Huffington Post.

Vader Sentiment is a parsimonious sentiment analyzer that measures both polarity and the intensity of a specific sentence. The sentiment was created by C.J. Hutto and Eric Gilbert of the Georgia Institute of Technology [6]. Instead of using machine learning methods such as Naive Bayes or neural networks to provide analysis on text, it uses many gold-standard rule based methods to calculate the sentiment. From there, they create estimates of the sentiment for each word. Vader Sentiment has worked particularly well in predicting social media text. Using news headlines as a proxy for social media text, Vader Sentiment may be a useful tool for analyzing the sentiment of news headlines.

In Python, Vader Sentiment has four scores: positive, negative, neutral, and composite. While the composite provides a similar score to polarity score (has the same -1 to 1 range), positive, negative, and neutral all show the intensity of text, hence providing a decomposition of the polarity [6].

Vader Sentiment and TextBlob rely on the case of the text, the associations, and the punctuations [4][6], so I do not remove any stop words or punctuation before estimating the sentiment scores on the headlines. I show the comparison of each of the sentiment analyzers against the change of the VIX for the period of the training and test sets in Figure 1.

TABLE I
CORRELATIONS OF SENTIMENT AND VIX CHANGES

	TB Polarity	TB Subjectivity	Vader Negative	Vader Positive	Vader Neutral	Vader Composite	VIX
TB Polarity	1	0.21	-0.1	0.4	-0.16	0.37	0
TB Subjectivity	0.21	1	0.42	0.36	-0.53	-0.12	0.02
Vader Negative	-0.1	0.42	1	0.06	-0.8	-0.75	0.07
Vader Positive	0.4	0.36	0.06	1	-0.65	0.58	0.07
Vader Neutral	-0.16	-0.53	-0.8	-0.65	1	0.23	-0.09
Vader Composite	0.37	-0.12	-0.75	0.58	0.23	1	-0.02
VIX	0	0.02	0.07	0.07	-0.09	-0.02	1

TABLE II
CORRELATIONS OF SENTIMENT AND FUTURE STEPS AHEAD LAGS

	VIX t+1	VIX t+2	VIX t+3	VIX t+4	VIX t+5	VIX t+6	VIX t+7
TB Polarity	0	0.02	0.06	0.04	0.04	0.02	0.02
TB Subjectivity	0.04	0.07	0.08	0.07	0.06	0.07	0.1
Vader Negative	0.07	0.11	0.12	0.11	0.09	0.08	0.08
Vader Positive	0.06	0.08	0.11	0.12	0.12	0.13	0.15
Vader Neutral	-0.09	-0.13	-0.16	-0.16	-0.14	-0.14	-0.15
Vader Composite	-0.02	-0.04	-0.02	-0.01	0.01	0.02	0.03

V. DATA PREPARATION

A. NLP Methods For Headline Filtering

To generate top topics for a thirty minute time period, I used many natural languages processing methods introduced in Professor Bari's Predictive Analytics HW2 assignment [8]. The first step was to remove all stop words and punctuations from the sentence. I removed all stop words using the list that I had used in that HW2 assignment. For my next step, I used NLTK, a Python Natural Language Processing toolkit to tokenize every News Headline and derive the lemmas from these words.

To find articles that were similar, I did not use the term-frequency, inverse document frequency (TF-IDF) measure. Since many headlines will only mention their top term once, the term-frequency will most likely be (1/number of terms in the headline). Therefore, longer headlines would be weighted down significantly, as well as common words, which would be detrimental to my analysis. Therefore, the value of a term for a headline only depended on if the term was in the headline (a binary score).

B. Top News Algorithm

To find similar headlines, I created the "Top News" clustering algorithm. The algorithm was based off of DBSCAN [12], an algorithm that finds similar items after removing "noisy points," articles that do not necessarily contain the top topics. My algorithm works as follows:

1) Similar to DBSCAN, I defined a metric, epsilon, and minpts parameters. For the metric, I used the discrete metric used in the k-modes algorithm [13]:

$$d(x,y) = \sum_{(x \in D_1, y \in D_2)} \delta(x,y)$$

$$\delta(x,y) = \begin{cases} 1 & if\ x = y \\ 0 & if\ x \neq y \end{cases}$$

Where x is an n-gram in Headline 1, D_1, and y is an n-gram in Headline 2, D_2. Epsilon is 3, and minpts is four. This means that for a document to be similar to share 3 key n-grams with another headline in order to be considered and has to be similar to four headlines to be a core point.

2) Define noise, border, and core points [12]. These have the same definitions from the general DBSCAN algorithm. From above, a core point will be similar to four other points and the border points are non-core points that are similar to a core point.

3) Cluster together all the border and core points to create one large cluster of "top headlines".

4) Average the sentiment scores of all of the headlines that were used.

As the algorithm would suggest, the "Top News" algorithm is an approximation algorithm for finding the top news headlines for the time period by only selecting headlines with similar phrases. The "Top News" algorithm makes the assumption that if multiple sources are writing articles about the same key topic, that topic is a "top headline" and can be influential information. To further ensure that the top topics for that half hour were selected, I chose to narrow down, before applying DBSCAN, to the top 30 n-grams for that half hour period. The reason for the choice of the top 30 n-

TABLE III
F-SCORES FOR GRANGER CAUSALITY TESTS

Lags	TB Polarity	TB Subjectivity	Vader negative	Vader positive	Vader neutral	Vader composite
1	0.779	0.331	0.686	0.976	0.766	0.726
2	0.508	0.132	0.048*	0.323	0.023*	0.508
3	0.146	0.231	0.084	0.105	0.017*	0.388
4	0.151	0.356	0.14	0.198	0.037*	0.479
5	0.216	0.504	0.188	0.267	0.072	0.443
6	0.278	0.547	0.245	0.252	0.077	0.542
7	0.362	0.365	0.355	0.364	0.156	0.646

*$p < 0.05$ **$p < 0.01$ ***$p < 0.001$

grams is that there are roughly 30 sources, thus a one-to-one correspondence for an n-gram per source. If no such top news story exists, then there is no eye-catching headline. Therefore, this has similar effect to headlines which are neutral and objective. Thus, if no "top headlines" exist for the half hour, I enter 0 for all sentiment scores, except for neutral, which I enter 1.

VI. MODELING

To model the results, I take four steps: correlation analysis, Granger Causality analysis, ARIMAX analysis, and recurrent neural networks.

A. Correlation Analysis

Correlation analysis is a quick method to evaluate if a relationship exists between two variables. I calculated the Pearson correlation coefficient of the aggregated sentiment scores and compared them to the changes in the VIX. In Table I, the other sentiment scores do have higher correlations with each other than the VIX. This is not surprising as the scores are not independent of the other sentiment scores.

In Table II, I calculated the sentiment scores up to seven lags of the current VIX Change. Noticeably, the magnitudes of the neutral, negative, and positive sentiments, which were already the three most correlated sentiments, seem to increase as the lags increase indicating that the VIX may not respond immediately to the news, but instead within one to three hours after the story was reported.

B. Granger Causality

Granger causality is an econometric tool for determining the influence of one time series or another. Specifically, a time series X granger causes Y if for two models,

$$M_0: Y_t = Y_{t-1} + \epsilon_t$$

$$M_1: Y_t = Y_{t-1} + X_{t-1} + \epsilon_t$$

M_1 significantly performs better than M_0 [13].

I performed the Granger Causality tests for 1 to 7 lags. The results showed significant effects for 2, 3, and 4 lags for Vader Neutral as shown in Table III. The interpretability of this seems rather intuitive when combined with its negative Pearson Correlation Coefficient. The less neutral sources are, the more likely people VIX investors will react. A slightly less significant effect exists with Vader Negative at lag of 2.

C. ARIMAX

In addition to Granger Causality showing significant effects between the Vader Neutral results, I used an ARIMAX model for predicting the linear relationship between Sentiments and VIX Changes. In Bollen, Mao, and Counts, the authors set I=0 and MA=0 (no differencing or moving average terms used) to estimate the model [3]. Following their format, I estimate VIX Change as the following:

$$Y_t = \alpha + \sum_{i=1}^{n} \beta_i \cdot Y_{t-i} + \sum_{i=1}^{n} \gamma_i \cdot X_{t-i}$$
$$where\ Y_t = \%\ \Delta VIX_t\ and\ X_t = Sentiment_t$$

Y_t is the log change of the VIX at time t. X_t, is any of the six sentiment variables: Text Blob Polarity, Text Blob Subjectivity, Vader Composite, Vader Negative, Vader Neutral, or Vader Composite at time, t. The results are compared to the AR model for the VIX Change.

D. Recurrent Neural Network

I model non-linear relationships with a Recurrent Neural Network on the sentiment scores and the changes of the VIX. A recurrent neural network is similar to a feed-forward neural network, as the network has input, hidden, and output layers. However, the weights in the hidden layer of a recurrent neural network is a function of not only transformation of input layers, but also of weights of previous sequences of the model. In time series models, the sequence of neural networks is that the weights of the hidden inputs at time t-1 are fed into the weights of the t [15].

Just like the ARIMAX, I estimated a non-augmented neural network (lags of the VIX Change) as well as an augmented

TABLE IV
RSMEs OF ARIMAX FORECASTS

Lags	No X	TB Polarity	TB Subjectivity	Vader Negative	Vader Positive	Vader Neutral	Vader Composite
1	**0.0395**	0.0397	0.0395	0.0396	0.0398	0.0398	0.0395
2	**0.0391**	0.0396	0.0395	0.0395	0.0397	0.0394	0.0398
3	**0.0391**	0.0398	0.0399	0.0402	0.0396	0.0400	0.0402
4	**0.0388**	0.0396	0.0395	0.0419	0.0391	0.0406	0.0408
5	**0.0387**	0.0399	0.0395	0.0423	0.0391	0.0407	0.0404
6	**0.0392**	0.0403	0.0402	0.0432	0.0397	0.0412	0.0414
7	**0.0409**	0.0420	0.0419	0.0463	0.0402	0.0435	0.0435
Best	**0.0387**	0.0396	0.0395	0.0395	0.0391	0.0394	0.0395

TABLE V
RSMEs OF RNN FORECASTS

Lags	No X	TB Polarity	TB Subjectivity	Vader Negative	Vader Positive	Vader Neutral	Vader Composite
1	0.0382	**0.0358**	0.0372	0.0368	0.0361	0.0374	0.0367
2	0.0387	0.0365	0.0373	0.0362	0.0364	0.0380	**0.0357**
3	0.0386	0.0374	0.0377	0.0370	0.0371	0.0394	**0.0366**
4	**0.0353**	0.0396	0.0376	0.0373	0.0372	0.0418	0.0393
5	0.0382	0.0383	0.0394	0.0385	0.0389	0.0396	**0.0368**
6	**0.0365**	0.0395	0.0404	0.0412	0.0390	0.0437	0.0402
7	0.0423	0.0414	0.0411	0.0409	**0.0396**	0.0463	0.0418
Best	0.0353	0.0358	0.0372	0.0362	0.0361	0.0374	0.0357

Lowest RMSE per lag in Bold

neural network (lags of the VIX Change and a Sentiment Score).

VII. EVALUATION

I use the five open market days between November 15th and 21st as a test set for my data. Bollen, Mao, and Zeng and Mao, Counts, and Bollen use the Mean-Absolute Predictive Error (MAPE) to evaluate the results of the model [1][3]. However, since I used the log-change of the VIX, near-zero log changes, can exaggerate the error for the prediction, as the denominator is divided by the near-zero log change. Instead, I use the Root-Mean Squared Error (RMSE) to compare predictions of the VIX for the ARIMAX and RNN models. According to Hyndman, the RMSE is a valid measure when all predictive measures are on the same scale. Since the VIX Change is the dependent variable in all models, the RMSE is a sufficient measure for this case. The RMSE is calculated as follows:

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

With y_i =actual value of the VIX change for observation i and $\hat{y}_i$=predicted value of the VIX change for observation i.

In Bollen, Mao and Counts, the VIX AR model augmented with Survey, Twitter, and News Sentiment performed better in the test set than the VIX AR model [3]. For the ARIMAX modeling in my study, the AR model had a better RMSE at all lags than all the models augmented with the sentiment variables, as shown in Table IV.

For the RNN, the test performance improved greatly with Vader Sentiment's composite score at lags 2, 3, and 5 , positive score at lag 7 and Text Blob's polarity score at lag 1 as shown in Table V. This is similar to Bollen, Mao, and Zeng in which their SOFNN performed better than the lags of Dow Jones when augmenting the calm and happy moods [1].

In addition to calculating the RMSE's I calculated the direction of the change. The direction is the first difference in the VIX Change from one 30 minute time period to the next. I calculated the accuracy of the direction by determining if the predicted difference in the change is equal to the actual difference in the change.

For the ARIMAX, as listed in Table VI, I find that the Vader negative measure has a 5 percentage point increase in directional accuracy from the VIX change itself in lags 4 and 5, which are the highest lags. Vader neutral, which was found to significantly granger cause the VIX Change at the 4th lag is also 2 percentage points higher. These results exhibit similar effects to that of Bollen, Mao, and Counts who saw a 7 percentage point increase using the combination of news, twitter, and survey sentiment [3].

For the recurrent neural network, as listed in Table VII, at lag 5, we find that there is a 5 percent increase (49 to 54 percent) for Vader Negative and a 4 percent increase for Text Blob Polarity. Direction for Vader Neutral and Text Blob Polarity.

TABLE VI
DIRECTION OF ARIMAX FORECASTS

Lags	VIX	TB Polarity	TB Subjectivity	Vader Negative	Vader Positive	Vader Neutral	Vader Composite
1	49.2	47.6	49.2	**50.8**	49.2	50.8	49.2
2	**47.6**	**47.6**	46	**47.6**	46	**47.6**	**47.6**
3	46	46	47.6	47.6	**49.2**	47.6	**49.2**
4	52.4	50.8	49.2	**57.1**	54	54	50.8
5	52.4	54	49.2	**57.1**	52.4	55.6	52.4
6	50.8	50.8	49.2	**55.6**	52.4	54	49.2
7	52.4	**54**	**54**	**54**	47.6	54	**54**
Best	52.4	54	54	**57.1**	54	55.6	54

TABLE VII
DIRECTION OF RNN FORECASTS

Lags	VIX	TB Polarity	TB Subjectivity	Vader Negative	Vader Positive	Vader Neutral	Vader Composite
1	47.6	49.2	50.8	**52.4**	47.6	50.8	46
2	**49.2**	47.6	46	47.6	44.4	46	47.6
3	**52.4**	46	46	49.2	50.8	49.2	46
4	**52.4**	50.8	49.2	**52.4**	49.2	50.8	46
5	49.2	50.8	49.2	47.6	**54**	47.6	44.4
6	50.8	50.8	50.8	49.2	49.2	**52.4**	49.2
7	50.8	47.6	47.6	49.2	49.2	**54**	47.6
Best	52.4	50.8	50.8	52.4	**54**	**54**	49.2

Highest Accuracy Per Lag in Bold

VIII. CONCLUSION AND DISCUSSION

This study has provided some substantial preliminary analysis on how sentiment analysis can be used to predict stock market volatility. In this paper, I collected VIX quotes using Alpha Vantage's package and news headlines using News API's package. I preformed sentiment analysis and Natural Language Processing filtering on the news headlines and created my own data clustering algorithm to derive sentiment indexes for the 30 minute time period. Like Twitter News Predicts the Stock Market, of the six sentiments, I found that only one, Vader Sentiment's Neutral Score, Granger Cause the VIX Change. When evaluating its accuracy, the neutral score preformed 2 percentage points more accurate at its most accurate lag than the VIX, while the negative score performed 5 percentage points better than the VIX's most significant model at lags 4 and 5. Similar results occurred for the recurrent neural network where the negative score had the best out of sample. Therefore, like Bollen, Mao, and Counts, using other sentiment can improve the accuracy by a modest amount [3].

However, the biggest conclusions made from these results that modelling intraday data is a challenging task. Bollen, Mao, and Zeng, increased their accuracy from 73 percent with a SOFNN of just VIX lags to 87 percent with a SOFNN of VIX Lags and Calm Sentiment[1]. Bollen, Mao, and Counts show an improvement from a VIX AR model to a VIX ARIMAX model with X being public sentiment, increasing the accuracy from 60 percent to 67 percent [3]. At its best lag, the accuracy only increases from 52 to 57 percent. In essence, the intraday lags of the VIX Change cannot sufficiently predict the direction of the difference in the VIX change.

Moreover, there were significant limitations to this analysis. The data was only collected over an 8-week period between September and November of 2018 due to the inability to query for news articles more than a month back. Many of the other studies such as those conducted by Bollen, Mao, and Zeng, Arias, Arratia, and Xuriguera, and Bollen, Mao, and Counts [1][3][10] contained data samples large enough (over 6 months) to provide statistical analysis on the effects of current sentimental text on a daily basis. In order to use a relatively large sample over eight weeks, the changes had to be calculated in intraday periods. Since my Python code has been written to incorporate new daily data, with proper future upkeep, this model can be tested on a larger dataset, which could hopefully provide even more robust results. As a result, there may be sufficient data to estimate a daily model. Furthermore, expanding the VIX time window from 30 minutes to 60 or 90 minutes could provide a larger window for breaking news to ensue, thus increasing the information for that time period. Therefore, while the preliminary results provide some improvement over models of the VIX on lags of itself, more data and different preparation methods might show even better improvement in predicting stock market volatility.

REFERENCES

[1] Bollen, Johan, Mao, Huina and Zeng, Xiaojun. (March 2011). Twitter Mood Predicts the Stock Market. Journal Of Computational Science. Vol 2. Issue 1. Accessed: https://www.sciencedirect.com/science/article/pii/S187775031100007X

[2] Chicago Board of Exchange. Cboe Volatility Index® (VX) Futures. Accessed: http://cfe.cboe.com/cfe-

products/vx-cboe-volatility-index-vix-futures/contract-specifications
[3] Mao, Huina, Counts, Scott, and Bollen, Johan (Dec 5, 2011). Predicting Financial Markets: Comparing Survey, News, Twitter and Search Engine Data. Accessed: https://arxiv.org/abs/1112.1051. arXiv:1112.1051
[4] Textblob Github Documention. Accessed: https://github.com/sloria/TextBlob
[5]Bari, Anasse, and Saatcioglu, Goktug (2018). Emotion Artificial Intelligence Derived from Ensemble Learning. 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications.
[6]Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.
[7] News API Documentation. Accessed: https://newsapi.org/docs
[8] Bari, Anasse. (2018) Predictive Analytics: Descriptive Modeling and Clustering of Textual Data.
[9] Bari, Anasse (2018). Predictive Analytics: Chapter One. Introduction to Predictive Analytics And Related Disciplines.
[10] Arias, Marta, Arratia, Argimiro, and Xuriguera, Ramon (April 2012). Forecasting with Twitter Data. ACM Transactions on Intelligent Systems and Technology, Vol. V, No. N, Article A. Accessed: http://www.lsi.upc.edu/~argimiro/mypapers/Journals/aax2012.pdf
[11] NLTK 3.4 Documentation. Accessed: https://www.nltk.org/
[12] Bari, Anasse (2018). Predictive Analytics: Chapter Seven. Data Clustering Algorithms.
[13] Zhexue Huang, Joshua (2009). Clustering Categorical Data with k-Modes. IGI-Global. Accessed: http://www.irma-international.org/viewtitle/10828/
[14] Hyndman, Rob. Measuring Forecast Accuracy (2014). https://pdfs.semanticscholar.org/af71/3d815a7caba8dff7248ecea05a5956b2a487.pdf
[15] Lewis, N.D. Neural Networks For Time Series Forecasting with R (2017).
[16] Connor, J.T., Martin, R.D., Atlas, L.E. Recurrent Neural Networks and Robust Time Series Prediction. IEEE Transactions on Neural Networks. Volume 5. Issue 3. Mar. 1994. Pgs. 240-254. Accessed: https://ieeexplore.ieee.org/document/279188
[17] Alpha Vantage Documentation. Accessed: https://alpha-vantage.readthedocs.io/en/latest/