Analyzing the Impact of Presidential Tweets on S&P 500 Close Prices Using Sentiment Analysis and Prediction Models

Kerem Bekmez
Dept. of Computer Engineering
Ozyegin University
Istanbul, Turkey
kerem.bekmez@ozu.edu.tr

Basar Yumakogullari
Dept. of Computer Engineering
Ozyegin University
Istanbul, Turkey
basar.yumakogullari@ozu.edu.tr

***Abstract*—This paper is a result of the CS440-Time Series in Finance class project, it looks at the relationship between presidential tweets and their influence on the S&P 500 close prices during Donald Trump's presidency. By applying sentiment analysis models such as VADER and FinBERT sentiment models, this study tries to find how sentiment scores affect predictive models. Both ARIMA and LSTM were applied to the data under different feature combinations, revealing key insights about the linear and non-linear dependencies between sentiment and market performance.**

***Index Terms*—Sentiment Analysis, Financial Time Series, ARIMA, LSTM, S&P 500, Presidential Tweets**

I. Introduction

Social networks have become a powerful tool that can have an affect on financial markets. This paper aims to analyze this relationship using sentiment analysis models and predictive modeling techniques.

We collected S&P 500 close prices and Trump's tweets during his presidency, cleaned the data, applied sentiment analysis using VADER and FinBERT models, and integrated these scores into ARIMA and LSTM models. This study aims to determine the predictive power of sentiment analysis in stock market movements.

II. Related Work

Review and research of:

- Tweet data for presidential periods.
- Sentiment analysis tools (e.g., VADER and FinBERT).
- Time-series forecasting with ARIMA and LSTM models.

III. Data Collection and Preprocessing

A. Tweet Gathering

We analyzed tweets, corresponding to the presidencies of Donald Trump and Joe Biden. Based on the data distribution shown in Fig. 1, we chose to focus on Donald Trump's tweets, as they provided a larger and more comprehensive dataset for sentiment analysis.

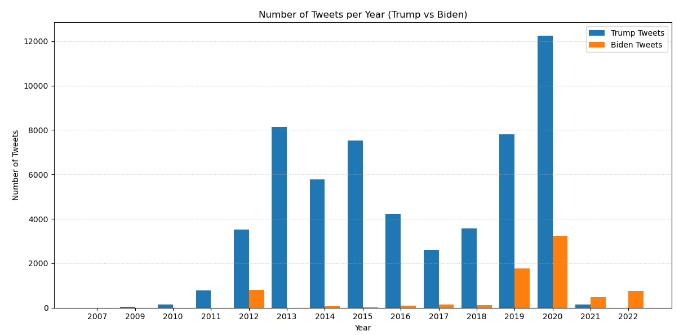

Fig. 1. Comparison of tweet counts during Donald Trump and Joe Biden's presidencies.

B. S&P 500 Data Gathering

After selecting the presidential period in Section 3.A, we knew the time we required for gathering S&P 500 data. This timeframe was going to align with Donald Trump's presidency, spanning from 01/03/2017 to 31/12/2020. During this period, we specifically focused on the S&P 500 closing indexes as our primary financial metric for analysis. The S&P 500 data was gathered using the `yfinance` library.

C. Data Cleaning

Data cleaning was a critical step to ensure the reliability and accuracy of our analysis. The process involved cleaning both the tweet data and the S&P 500 financial data. Below, we explain the steps taken for both tasks.

1) Tweet Data Cleaning: The raw tweet data, sourced from Donald Trump's presidency, contained lots of different characters that are not welcomed by the sentiment analysis models, just like other tweets. Such as URLs, hashtags, mentions, special characters and etc. To prepare the data for sentiment analysis, the following preprocessing steps were applied:

- Removed URLs, mentions, and hashtags using regular expressions.

- Removed punctuation and converted all text to lowercase for consistency.
- Filtered out tweets with missing or empty content.
- Standardized date formats and aligned tweets by date.

The pseudocode for the tweet cleaning process is provided in Algorithm 1.

Algorithm 1 Tweet Cleaning Algorithm

```
Input: Raw tweets dataset
Output: Cleaned tweets dataset
for each tweet in dataset do
    Remove URLs, mentions, and hashtags
    Remove punctuation and special characters
    Convert text to lowercase
    Drop rows with missing or empty text
    Standardize date formats
end for
Save cleaned dataset
```

After applying this code to raw tweet data, the following tables Table I and Table II show a sample of before and after for the tweets.

The cleaned data was later saved as *SP500_IndexCloses.csv* for future use.

TABLE I
RAW TWEETS

Date	Raw Tweet
2011-08-02	Republicans and Democrats have both created our economic problems. Time to take America back.
2020-03-03	I was thrilled to be back in the Great city of Charlotte. Tremendous crowd for @FoxNews!
2020-01-17	RT @CBS_Herridge: READ: Letter to surveillance court obtained by @CBSNews.

TABLE II
CLEANED TWEETS

Date	Cleaned Tweet
2011-08-02	republicans and democrats have both created our economic problems time to take america back
2020-03-03	i was thrilled to be back in the great city of charlotte tremendous crowd for foxnews
2020-01-17	rt read letter to surveillance court obtained by cbsnews

2) S&P 500 Data Cleaning: The S&P 500 financial data was collected using the `yfinance` library. The following steps were taken to clean and process the data:

- Data for all S&P 500 tickers was downloaded within the timeframe from 01/03/2017 to 31/12/2020, corresponding to Donald Trump's presidency.
- Only the `Close` prices were retained as the primary financial metric.
- Missing values and incomplete records were filtered out.

The cleaned data was later saved as *SP500_IndexCloses.csv* for future use. A sample of the data is shown in Table III.

TABLE III
SAMPLE OF CLEANED SP500_INDEXCLOSES.CSV

Date	Close Price (USD)
2017-01-03	2257.83
2017-01-04	2270.75
2017-01-05	2269.00
2017-01-06	2276.98
2017-01-09	2268.90
2017-01-10	2268.90
2017-01-11	2275.32
2017-01-12	2270.44
2017-01-13	2274.64
2017-01-17	2267.89

After this cleaning process, both the tweet and the S&P500 datasets were ready.

D. Sentiment Analysis

Sentiment analysis was conducted using two models: **VADER** and **FinBERT**, which are 2 different sentiment analysis techniques, Vader being a more general sentiment analysis technique where as FinBERT is more finance focused.

1) VADER Sentiment Analysis: VADER (Valence Aware Dictionary and sEntiment Reasoner) is a rule-based model that specializes in social media text. It generates a compound sentiment score ranging from -1 (most negative) to +1 (most positive). The compound score for each tweet was calculated, and daily average sentiment scores were derived by grouping tweets by date.

2) FinBERT Sentiment Analysis: FinBERT is a transformer-based model pre-trained on financial texts, specifically designed to analyze market-related sentiment. Tweets were processed in batches, with each tweet classified into `Positive`, `Negative`, or `Neutral` sentiment categories with a score. Sentiment scores were mapped as follows:

- `Positive`: The score is retained as is.
- `Negative`: The score is negated.
- `Neutral`: Assigned a score of 0.

Daily average FinBERT sentiment scores were then calculated just like in the VADER sentiment by grouping the tweets and the scores by date.

3) Comparison of Models: While VADER provides general sentiment insights, FinBERT focuses more on financial contexts, offering a domain-specific perspective. This dual approach captures both the generic emotional tone and financial implications of tweets. This paper will be inspecting the results later on in the results section.

4) Daily Sentiment Averages: The daily averages for both models were calculated using the following steps:

1) Group tweets by `Date`.
2) Count the number of tweets for each date.
3) Compute the average sentiment scores for VADER and FinBERT models for that date.

The process can be summarized in the following pseudocode:

Algorithm 2 Daily Sentiment Averages Pseudocode

1: **Input:** Dataset of tweets with sentiment scores (`tweets`)
2: **Output:** Daily sentiment dataset (`daily_sentiment`)
3: Group `tweets` by `Date`.
4: Compute:
- `number_of_tweets`: Count of tweets for each date.
- `avg_vader_sentiment`: Mean VADER sentiment score for each date.
- `avg_finbert_sentiment`: Mean FinBERT sentiment score for each date.

5: Save the result as `daily_sentiment` for future use.

A sample of the dataset that resulted by the algorithm 2 is shown in Table IV.

TABLE IV
SAMPLE OF DAILY SENTIMENT AVERAGES
(DAILY_SENTIMENT_WITH_BATCHES.CSV)

Date	Number of Tweets	Avg. VADER	Avg. FinBERT
2017-01-03	12	0.45	0.32
2017-01-04	15	0.39	0.28
2017-01-05	10	0.50	0.30

IV. DATA ALIGNMENT

The alignment of sentiment scores, which were already combined, with S&P 500 data was essential to ensure robust analysis. By aligning data on the `Date` column, we achieved a coherent dataset that links sentiment scores with market performance.

1) Pseudocode for Data Alignment: The pseudocode below demonstrates the process of aligning the S&P 500 data with the sentiment analysis results:

Algorithm 3 Data Alignment Pseudocode

1: Load the `cleaned_sp500_index_prices` dataset.
2: Load the `daily_sentiment_with_batches` dataset.
3: Perform a left join on `Date` to merge `SP500` data with `VADER` sentiment scores.
4: Perform another left join on `Date` to merge the result with `FinBERT` sentiment scores.
5: Replace any missing sentiment scores with 0.
6: Save the resulting aligned dataset as `aligned_data`.

2) Aligned Data Preview: After alignment, the dataset included S&P 500 close prices and daily sentiment scores. A preview is shown in Table V.

TABLE V
ALIGNED DATA SAMPLE (ALIGNED_DATA.CSV)

Date	Close Price (USD)	Number of Tweets	Avg. VADER	Avg. FinBERT
2017-01-03	2257.83	12	0.45	0.32
2017-01-04	2270.75	15	0.39	0.28
2017-01-05	2269.00	10	0.50	0.30

V. PREDICTIVE MODELS

A. Approach

This study examined the impact of sentiment analysis on stock market predictions using three different approaches and two predictive models. The approaches were defined as follows:

1) **Close Prices Only**: Using only the historical S&P 500 close prices as input features.
2) **Close Prices + VADER Sentiment Scores**: Combining historical close prices with daily average VADER sentiment scores.
3) **Close Prices + FinBERT Sentiment Scores**: Combining historical close prices with daily average FinBERT sentiment scores.

The rationale behind this setup was to use the "Close Prices Only" model as a benchmark to evaluate the added value of sentiment scores in predicting market behavior.

Two predictive models, ARIMA and LSTM, were then chosen for this analysis:

- **ARIMA**: A linear time series model used to capture linear relationships in the data. Its simplicity provides a baseline for evaluating linear dependencies.
- **LSTM**: A deep learning model designed to handle sequential data and capture non-linear relationships. LSTM's ability to learn from complex patterns in time-series data makes it suitable for assessing non-linear dependencies introduced by sentiment scores.

The combination of these approaches and models ensured a comprehensive analysis of how sentiment scores influence the predictability of stock market trends.

1) ARIMA: The ARIMA (AutoRegressive Integrated Moving Average) model was applied to forecast S&P 500 close prices using three distinct feature sets. ARIMA is a popular statistical time series model that captures linear dependencies in data. The implementation involved the following scenarios:

1) **Close Prices Only**: In this scenario, only the historical S&P 500 close prices were used as input features to establish a baseline model.
2) **Close Prices + VADER Sentiment Scores**: This approach incorporated the daily average VADER sentiment

scores as exogenous variables to evaluate their influence on market predictions.

3) **Close Prices + FinBERT Sentiment Scores**: Here, the daily average FinBERT sentiment scores were included as exogenous variables to assess their contribution to predictive performance.

a) Implementation Details: The following steps were performed for each scenario:

- The dataset was divided into training (80%) and testing (20%) subsets.
- The `auto_arima` function was used to automatically select the optimal ARIMA parameters (p, d, q) based on the training data. This function is doing a cross-validation within the given range of p,d and q values. This is crucial for having the best model possible.
- For scenarios involving sentiment scores, the sentiment features (`avg_vader_sentiment` or `avg_finbert_sentiment`) were treated as exogenous variables.
- Forecasts were generated for the testing subset.
- Model performance was evaluated using metrics such as:
  - Mean Squared Error (MSE)
  - Root Mean Squared Error (RMSE)
  - Mean Absolute Error (MAE)
  - Coefficient of Determination (R^2)

b) Pseudocode: The algorithm for applying ARIMA in these scenarios is summarized below:

Algorithm 4 ARIMA Implementation

1: Split dataset into training (80%) and testing (20%) subsets.
2: **for** each scenario **do**
3:     **if** Sentiment scores are included **then**
4:         Use *avg_vader_sentiment* or *avg_finbert_sentiment* depending on the scenario as exogenous variables.
5:     **end if**
6:     Fit `auto_arima` to the training data with appropriate features.
7:     Generate forecasts for the testing subset.
8:     Compute performance metrics: MSE, RMSE, MAE, and R^2.
9: **end for**

The results will be discussed in section VI.

2) LSTM: The LSTM (Long Short-Term Memory) model was employed to analyze and predict stock market trends using three different feature sets. This method leverages LSTM's capability to capture complex, non-linear patterns in sequential data. The three feature sets used for the LSTM model were:

1) **Close Prices Only**: Historical S&P 500 close prices as input.
2) **Close Prices + VADER Sentiment Scores**: Close prices combined with VADER daily average sentiment scores.
3) **Close Prices + FinBERT Sentiment Scores**: Close prices combined with FinBERT daily average sentiment scores.

a) Data Preprocessing and Normalization:: To prepare the data for the LSTM model:

- The `Close` prices and sentiment features (*avg_vader_sentiment* and *avg_finbert_sentiment*) were normalized using Min-Max scaling to ensure values are within the range [0, 1]. Min-Max scaling is applied to ensure that the finbert scores, which are between -1 and 1 are not interpreted as noise because they are much smaller in scale compared to the close indexes which are around 3000 - 5000.
- Sequential input data was created with a sequence length of 30, meaning the model uses the last 30 time points to predict the next day's value.

b) Model Architecture:: The LSTM model architecture was designed as follows:

- A single LSTM layer with 50 units and ReLU activation.
- A dense output layer with one neuron to predict the next `Close` price.
- The Adam optimizer and Mean Squared Error (MSE) loss function were used for training.
- Early stopping was implemented to prevent overfitting by monitoring validation loss with a patience of 10 epochs.

c) Model Training and Scenarios:: Three models were trained corresponding to each feature set:

1) **Scenario 1**: Trained using only `Close` prices.
2) **Scenario 2**: Trained using `Close` prices and VADER sentiment scores.
3) **Scenario 3**: Trained using `Close` prices and FinBERT sentiment scores.

d) Evaluation:: The trained models were evaluated on the test set using the following metrics:

- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- Mean Absolute Percentage Error (MAPE)
- R^2 Score

e) Pseudocode for LSTM Training and Evaluation:: The overall process for training and evaluating the LSTM model is summarized below:

Algorithm 5 LSTM Training and Evaluation Pseudocode

```
Load and normalize features (Close, avg_vader_sentiment,
  avg_finbert_sentiment) based on the curent scenario.
Create sequences for input data with a sequence length of
  30.
for each feature set do
    Split data into training and testing sets.
    Train the LSTM model using the training set.
    Evaluate the model on the testing set.
    Compute MSE, RMSE, MAPE, and R^2 score.
end for
```

f) Results:: The results for each scenario are discussed in detail in Section VI, showcasing how sentiment scores impact the predictive capability of the LSTM model.

VI. Results and Discussion

A. ARIMA Results

The ARIMA models were tested using three feature sets: close prices only, close prices combined with VADER sentiment scores, and close prices combined with FinBERT sentiment scores. The results for all scenarios were identical, indicating no additional predictive value from sentiment scores.

TABLE VI
ARIMA Results for Different Scenarios

Scenario	MSE	RMSE	MAE	R²
Close Only	773,893.15	879.71	815.64	-5.81
Close + VADER	773,893.15	879.71	815.64	-5.81
Close + FinBERT	773,893.15	879.71	815.64	-5.81

The lack of improvement is expected from ARIMA, since it is unable to capture non-linear relationships. The weak correlation between close prices and sentiment scores further supports this limitation, as shown in Table VII.

TABLE VII
Correlation Between Close Prices and Sentiment Scores

Variable Pair	Correlation Coefficient
Close and avg_vader_sentiment	-0.0908
Close and avg_finbert_sentiment	-0.0759

These low correlations indicate that sentiment scores have minimal linear influence on stock prices, hence ARIMA model acts as if they are just noise.

B. LSTM Results

LSTM models, which excel at capturing non-linear relationships, showed performance differences across the three scenarios, as shown in VIII.

TABLE VIII
LSTM Results for Different Scenarios

Scenario	MSE	RMSE	R²	MAPE (%)
Close Only	5,848.80	76.48	0.9382	1.93
Close + VADER	6,432.94	80.21	0.9320	2.08
Close + FinBERT	4,785.91	69.18	0.9494	1.73

a) Discussion:: Incorporating VADER sentiment scores slightly worsened the model's performance, as indicated by higher MSE, RMSE, and MAPE compared to the baseline (Close Only). On the other hand, using FinBERT sentiment scores slightly improved performance across all metrics, highlighting its efficiency in capturing meaningful sentiment signals for stock price predictions.

C. Comparison Between VADER and FinBERT

FinBERT outperformed VADER in the LSTM models due to its domain-specific design for financial data. Unlike VADER, which provides a general sentiment score, FinBERT analyzes text with a financial context, making its sentiment scores more relevant to stock market behavior. The difference can be better seen in table IX, which is a small sample from the *daily_sentiment_with_batches.csv*.

TABLE IX
Comparison of Sentiment Scores for Selected Dates

Date	VADER Sentiment	FinBERT Sentiment
2009-06-26	0.5106	0.9942
2009-06-30	0.4215	0.0000
2009-07-04	0.8689	0.9996
2009-07-13	0.3818	0.0000
2009-07-15	-0.0516	0.0000
2009-07-22	0.9169	0.0000
2009-07-28	0.0000	0.0000
2009-08-04	0.5719	0.9852
2009-08-11	0.0000	0.0000
2009-08-14	0.4767	0.9947
2009-08-20	0.0000	0.0000
2009-08-21	-0.1531	-0.8572

The differences between VADER and FinBERT sentiment scores are obvious. FinBERT, being a finance-specific model, better identifies financial context, whereas VADER often misinterprets non-financial text as having financial sentiment value. This is evident in several cases:

- On **2009-08-21**, FinBERT assigns a strongly negative sentiment score of `-0.8572`, while VADER assigns a weaker score of `-0.1531`.
- On **2009-07-04**, FinBERT identifies a strongly positive sentiment with a score of `0.9996`, whereas VADER assigns a lower score of `0.8689`.
- Many cases (e.g., **2009-07-13**, **2009-07-22**, and **2009-08-11**) where FinBERT assigns a score of `0.0000`, indicate that these texts were not financial in nature, while VADER provides non-zero scores.

These observations highlight FinBERT's ability to distinguish financial relevance from general sentiment, making it more reliable for financial sentiment analysis.

These differences make FinBERT more suitable for analyzing the financial implications of tweets, as reflected in the improved LSTM performance in Scenario 3.

VII. Conclusion and Future Work

This study explored the impact of presidential tweets on the S&P 500 close prices through sentiment analysis and predictive modeling using ARIMA and LSTM models.

A. Findings

- **ARIMA's Limitations:** The ARIMA models, tested with close prices alone and with sentiment scores as exogenous variables, showed no improvement in predictive performance. This outcome aligns with ARIMA's inherent inability to capture non-linear relationships, as evidenced by the weak correlations between sentiment scores and close prices.
- **LSTM's Performance:** The LSTM models demonstrated superior predictive capability, particularly when FinBERT sentiment scores were incorporated. FinBERT's domain-specific focus on financial contexts resulted in improved performance metrics (MSE, RMSE, MAPE, and R^2), underscoring its relevance for financial sentiment analysis. In contrast, VADER sentiment scores slightly degraded the model's performance due to their general sentiment analysis focus.

B. Future Work

The findings from this study pave the way for several future research directions:

- **Hybrid Models:** Explore the integration of ARIMA with LSTM or other machine learning models to capture both linear and non-linear patterns in financial time series data.
- **Expanded Sentiment Analysis:** Investigate additional domain-specific sentiment analysis models, such as FinSentS or other transformer-based architectures, to enhance predictive power.
- **Alternative Financial Indices:** Extend the analysis to other financial indices (e.g., Dow Jones, Nasdaq) or commodity prices to validate the findings across diverse markets.
- **Granular Data Analysis:** Incorporate more granular tweet-level data (e.g., retweets, likes, or time of day) to assess intra-day market movements influenced by sentiment.
- **Real-Time Predictions:** Develop real-time predictive systems integrating live sentiment analysis from social media feeds and stock market data.

The insights gained from this research show that there is a relation between social media sentiment and financial market trends, with promising applications for both academic research and practical applications in finance and economics.

References

- Donald Trump Tweets Retrieved from https://www.kaggle.com/datasets/codebreaker619/donald-trump-tweets-dataset
- Joe Biden Tweets Retrieved from https://www.kaggle.com/datasets/rohanrao/joe-biden-tweets