Stock Market Prediction from Sentiment and Financial Stock Data Using Machine Learning

Mohammad Al Ridhawi† and Hussein Al Osman‡
Faculty of Engineering, University of Ottawa, Ottawa, ON, Canada
{malri039, Hussein.AlOsman}@uottawa.ca

Abstract
Forecasting stock values is challenging due to market volatility and numerous financial variables, such as news, social media, political changes, investor emotions, and the general economy. Predicting stock value using financial data alone may be insufficient. By combining sentiment analysis from social media with financial stock data, more accurate predictions can be achieved. We use an ensemble-based model employing multi-layer perceptron, long short-term memory, and convolutional neural network models to estimate sentiment in social media posts. Our models are trained on AAPL, CSCO, IBM, and MSFT stocks, using financial data and sentiment from Twitter between 2015-2019. Results show that combining financial and sentiment information improves stock market prediction performance, achieving a next-hour prediction performance of 74.3%.

Keywords: Stock Market, Social Media, Twitter, Financial Stock Data, Machine Learning, Neural Networks

1. Introduction
Historically, stock market predictions relied on traditional techniques, while recent approaches employ machine learning to recognize influencing variables [1]. Traditional methods overlooked investor sentiment, which significantly affects stock prices [2]. Combining sentiment analysis from social media with financial data may improve prediction performance [3]. Various studies explored predicting stock market movements using sentiment scores and deep learning models, considering factors such as news information and technical indicators [4].

Integrating sentiment analysis with financial data enables more comprehensive stock market predictions by considering investor sentiment from social media platforms like Twitter alongside traditional financial metrics, capturing market nuances.

In this paper, we predict stock market movements using sentiment and financial data. We extract sentiment estimates from social media with an ensemble-based model comprising CNN, LSTM, and MLP member models. The extracted sentiment scores and financial stock features are input to a fully connected network to predict the stock price in one hour. The paper is organized into sections on prediction approach, experimental setup, results, evaluation, and conclusion.

2. Methodology
This paper presents the Sentiment and Financial Stock Data (SFSD) model for next-hour stock value prediction. The SFSD model employs the Ensemble Sentiment Estimation (ESE) model to derive sentiment scores from social media posts mentioning stocks, which are then combined with financial stock dataset features for stock price prediction.

The ESE model is a stacked ensemble consisting of 2 MLPs, a CNN, and an LSTM. The MLP Feature Driven (MLP FD) model uses stock, linguistic, and sentiment lexicon features, while the MLP Simple Word Embedding (MLP SWE), CNN, and LSTM models process word-embedding representations of social media posts, utilizing Google Word2Vec. We limit input
to 50 words, padding shorter messages. The ensemble members are complementary: MLP FD extracts information from widely used NLP features, MLP SWE identifies patterns in text vector representations, CNN captures local information, and LSTM models global text information. The ESE outputs a sentiment score, which the four ensemble members forward to a sentiment stacked fusion model, an MLP network producing the ESE output.

The ESE generates a sentiment score for each social media post. Features such as standard deviation, mean, median, minimum, and maximum sentiment scores are extracted from a set of posts about a stock. These features, along with financial stock dataset features, are input into a fully connected sentiment and financial fusion model to predict the final stock value. Refer to Section 2.1 for details on the financial stock dataset features.

We use the cosine similarity metric to assess the performance of the ESE model. We use the Mean Absolute Percentage Error (MAPE) (Equation (1)) and accuracy metrics to assess the performance of the SFSD model.

The MAPE metric has been widely employed to evaluate stock prediction models [5]. It is especially popular given how simple it is to interpret its significance. For instance, MAPE value of 10% signifies that the difference between the actual and the forecasted value is 10%. The model’s performance is computed using MAPE by calculating the absolute percentage error.

\[
MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{x_i - \hat{x}_i}{x_i} \right| * 100 \tag{1}
\]

Where \( n \) is the number of samples, \( x_i \) is the stock price, and \( \hat{x}_i \) is the predicted stock price.

Given that stock prediction is a regression problem, we assess accuracy based on the movement of stock prices. Hence, the model produces a correct output if it correctly predicts that the stock price will increase or decrease and a wrong prediction otherwise.

2.1. Financial Stock Dataset

Using Python’s ‘yfinance’ package, we collected a financial stock dataset from Yahoo Finance for tickers $AAPL, $CSCO, $IBM, and $MSFT from 2015-01-01 to 2019-12-31, resulting in 28,224 records. The dataset is divided into 70% training, 20% validation, and 10% testing. Features include Open, High, Low, Close, Volume, Date, time, weekday, true range, Bollinger Indicator, stock closing price differences from Bollinger bands, and simple moving average indicator [7].

2.2 Social Media Dataset

We tested the ESE model on SemEval-2017 Task 5 (SET5) and the Apple, Cisco, IBM, and Microsoft (ACIM) datasets. SET5, comprising financially relevant microblog messages with sentiment scores, was used to train, validate, and test the ESE model. ACIM, collected from Twitter API and featuring tweets about four stocks from 2015-01-01 to 2019-12-31, was used
to train, validate, and test the sentiment-financial fusion model, integrating ESE outputs with financial stock features. ACIM is divided into 70% training, 20% validation, and 10% testing, labeled for stock value but not sentiment.

2.1.1. Text Cleaning

We clean social media text before extracting features for the ESE members. User ID and sensitive information are removed. Retweet symbols are eliminated, and URLs are replaced with '_url'. We correct abbreviations using a slang dictionary and standardize word contractions. Spelling and word elongation are fixed, and numeric values and symbols are cleaned. Symbols are joined with their respective words, unnecessary apostrophes and punctuation are removed, and ordinal words are converted to their ordinal number counterparts.

2.1.2. Feature Engineering

The MLP FD model of the ESE considers numerous engineered features such as the time in which the social media message was created, total likes, the number of capital letters, and counting hashtags. In fact, 64 features are taken from the initial data. We extract three types of features, namely, stock, linguistic, and sentiment lexicon. We describe these features in the following sub-sections.

2.1.2.1. Stock Features

Financial domain data often contain numbers indicating bullish or bearish sentiment, impacting stock sentiment scores. We extract 20 binary features from social media data, common in machine learning classification problems [8]. These binary features include '+number%', '-number%', '$number', 'number%', and ordinal numbers, representing positive/negative percentages, dollar amounts, percentages, and ordinal values.

2.1.2.2. Linguistic Features

The linguistic features consist of 22 features: i) seven integer linguistic features, ii) six Part-of-Speech (POS) tags, iii) one pointwise mutual information, iv) four Term Frequency-Inverse Documentation Frequency (TF-IDF) n-gram features, and v) four Relevance Frequency (RF) n-gram features. The Stanford NLP python package [9] is used to extract POS tags. Integer features include counts of '!', '?', '$', continuous '!', continuous '?', capitalized words, and hashtags. Verb features are extracted as Bag-of-Words from the sentence using POS tags (VB, VBD, VBG, VBN, VBP, VBZ). For generating TF-IDF N-gram features, we filter social media messages, using n-grams widely in NLP tasks. The four TF-IDF n-gram features are the average TF-IDF 1-gram, 2-gram, 3-gram, and 4-gram.

2.1.2.3. Sentiment Lexicon Features

We utilize publicly available sentiment lexicons, including AFINN, Bing Liu opinion Lexicon, NRC Hashtag Sentiment Lexicon [10], General Inquirer Lexicon, and SentiWordNet [11], which are crucial for sentiment analysis tasks. For each word in the text, we calculate five features: ratio of positive words, ratio of negative words, maximum sentiment score, minimum sentiment score, and sum of sentiment scores. Examples include AFINN sum score, Bing Liu positive word ratio, and General Inquirer negative word ratio. We use a total of 19 sentiment lexicon features, as shown in Table 1.

<table>
<thead>
<tr>
<th>Aggregation Strategy</th>
<th>AFINN</th>
<th>Bing Liu</th>
<th>NRC Hashtag</th>
<th>General Inquirer</th>
<th>SentiWordNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Min</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pos Ratio</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Neg Ratio</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 1. Sentiment Lexicon Features
2.1.3. Feature Selection

We used four analysis methods to select the most relevant engineered features for the MLP FD model: Missing Value, Constant Variable, Duplicated Variables, and Correlated Variables analyses. Missing Value analysis identifies features with missing values. Constant Variable analysis removes features that are almost constant. After applying this method, six features were removed. Duplicated Variable analysis removes duplicate features; however, our dataset had unique features. Lastly, we assessed feature correlation and removed one feature due to a 0.99 correlation with another. This process reduced the engineering features to 58.

2.1.4. Dataset Partition

We split the SET5 dataset into training (70%), validation (20%), and test (10%) sets. All ESE members are trained and validated using the training set through the stratified 10-fold cross-validation. The member models are then tested on the validation and test sets. The sentiment stacked fusion model (see Figure 1) is then trained on the predictions of the member models for the validation set. It is validated on the predictions of the member models for the test set.

3. Experiments

We use the Yahoo Finance package (yfinance python package) to obtain the financial stock dataset, with timestamps in one-hour increments. The ACIM dataset has finer timestamps, so we aggregate sentiment scores for tweets during each hour. Sentiment scores during closed hours are associated with the opening hour of the next day. The SET5 dataset is used to train, validate, and test our ESE model, while the ACIM and financial stock datasets are used for the sentiment and financial fusion model.

3.1. ESE Member Models’ Architectures

The MLP Feature Driven and MLP Simple Word Embedding models both have 3 hidden layers, but with different node configurations: 50, 30, and 15 nodes for MLP FD, and 30 nodes in each layer for MLP SWE. Both employ ReLU for hidden layers and TanH for output layers. They share dropout layer values (0.5), input dropout (0.25), batch size (32), optimizer (Adam), and loss function (MSE). Differences lie in hidden dropout, L2 regularization, epochs, and learning rate.

In Section 2.2, the CNN model processes cleaned text (up to 50 words) using Google’s Word2Vec, converting words into 300D vectors. The CNN has a Gaussian noise layer and four Tanh-activated convolutional layers with output dimensions of 300, 299, 298, and 297. After applying 1-max pooling, the model generates 100 univariate vectors which are concatenated into a 100D feature vector, inputted into a fully connected network for further processing.

The LSTM model captures the global behavior of the text. Similar to the CNN model, the LSTM model takes the cleaned text as explained in Section 2.2 as the input to the model, with a maximum of 50 words in each text segment. We use Google’s Word2Vec pre-trained model to convert each word into its 300-dimensional vector representation. The vectors are then passed sequentially to the LSTM model.

For the CNN and LSTM models, the architectures and hyperparameters differ. The CNN has two hidden layers with 15 nodes each, while the LSTM has two hidden layers with 50 and 10 nodes. The CNN uses the TanH activation function, a 0.45 dropout rate for hidden layers, 0.006 L2 regularization, and runs for 75 epochs. In contrast, the LSTM utilizes the ReLU activation function, a 0.3 dropout rate for hidden layers, 0.05 L2 regularization, and is trained for 350 epochs. Both models share a batch size of 32, with the CNN having a learning rate of 0.0005 and the LSTM a rate of 0.0001. They both employ the Adam optimizer and the Mean Square Error (MSE) loss function.

3.2. ESE Member Models’ Performance

ESE member models’ stratified cross-validation performance: MLP FD (avg: 0.514, SD: 0.076), MLP SWE (avg: 0.568, SD: 0.086), CNN (avg: 0.572, SD: 0.082), LSTM (avg: 0.542, SD:
3.3. Sentiment Stacked Fusion Model

We employ the sentiment stacked fusion model, an MLP network, to fuse the sentiment estimates of the four ensemble member models, namely MLP FD, MLP SWE, CNN, and LSTM. Stacking is an ensemble machine learning approach that uses a meta-learning algorithm to optimally combine the predictions of the ensemble members [12]. The benefit of using stacking is to harness the capabilities of models that perform moderately well in classification and regression tasks to produce a model that exhibits better performance than any model used in the ensemble.

The architecture and hyperparameters for the MLP stacked ESE fusion model consist of one hidden layer with four nodes, ReLU activation function for the hidden layer, TanH activation function for the output layer, a dropout layer with a rate of 0.05, L2 regularization with a value of 0.02, 300 epochs, a batch size of 32, a learning rate of 0.00075, the Adam optimizer, and the Mean Square Error (MSE) loss function.

Each of the four models, MLP FD, MLP SWE, CNN, and LSTM, achieved a cosine similarity average score around 0.5. By combining these models using the MLP stacked ESE fusion model, a higher cosine similarity of 0.674 (67.4%) was achieved, demonstrating the effectiveness of the ensemble approach.

3.4. Sentiment and Financial Fusion Model

The objective of this paper is to combine the financial stock and sentiment information to predict stock market prices. Hence, we propose fusing the output of the ESE model and the financial stock dataset features to generate a prediction that leverages the sentiment and the financial data. We employ the sentiment and financial fusion model, a fully connected network, to combine the sentiment and financial stock dataset features and produce the final stock prediction. To assess the performance of the proposed solution, we train and validate the sentiment and financial fusion model on the training and validation sets of the financial stock dataset described in Section 2.1 and ACIM dataset described in Section 2.2.

The SFSD fusion model takes the five sentiment score outputs from the ESE model (standard deviation, mean, median, minimum, and maximum sentiment score) and the financial stock data features. The input to the model is considered time-series data, requiring time-series cross-validation for the combined data while optimizing the hyperparameters.

The architecture and hyperparameters for the SFSD fusion model include two hidden layers with 128 and 64 nodes, ReLU activation function, no input dropout layer, a dropout layer with a rate of 0.175, L2 regularization with a value of 0.0025, 275 epochs, a batch size of 25, a learning rate of 0.0004, the Adam optimizer, and the Mean Square Error (MSE) loss function.

4. Evaluation

4.1. Evaluating the Sentiment and Financial Fusion Model on the Test Set

The SFSD model is evaluated using the test set composed of the test sets from the ACIM dataset and FSD. The results achieved for individual stocks and the overall performance of the model on the MAPE and accuracy metrics are as follows: Prediction accuracy for $AAPL is 74.5%, for $CSCO is 74.4%, for $IBM is 74.1%, and for $MSFT is 74.2%. The average prediction accuracy is 74.3%, with a standard deviation of 0.0012. The overall MAPE value is 1.079.

4.2. Comparing the SFSD Model to the State of the Art

We evaluate the sentiment and financial fusion model using the test sets of the ACIM dataset (see Section 2.2) and financial stock dataset (see Section 2.1).

First, we compared the SFSD to the model presented in Xiang et al. [13] to predict the S&P 500 index fund. The Xiang et al. [13] model achieved an accuracy of 50%. We test the SFSD
model using a dataset that correspond to the same timeframe specified in Xiang et al. [13]. The SFSD model achieved a 67% average accuracy. When we reduced the timeframe to the years 2015-2019, the SFSD accuracy rises to 81%. Second, we evaluated the SFSD model to the solution proposed by Dey et al. [14] where achieved an accuracy between 85-99% on the Yahoo and Apple stocks [14]. The study did not specify which dates were chosen. However, they used a timeframe of 28 days, 60 days, and 90 days. To obtain a dataset for this comparison, we randomly selected a starting date and collected data for a 28-, 60-, and 90-days time period. The proposed SFSD model achieved an average accuracy of 94% accuracy for all timeframes and both stocks.

Third, we compared the SFSD model to the approach proposed by Pagolu et al. [13]. Pagolu et al. [13] achieved an accuracy of 70.1 percent on the prediction of the MSFT stock price. The data was collected between the dates of August 2015 and August 2016. We tested the SFSD model using a dataset that correspond to the same timeframe specified in Pagolu et al. [13]. The SFSD model achieved an accuracy of 84.7%.

Comparing our SFSD model with state-of-the-art models from highly cited papers, such as Xiang et al.'s machine learning and deep learning techniques, Dey et al.'s eXtreme Gradient Boosting, and Pagolu et al.'s Random Forest, demonstrates its technical prowess and potential for significant impact in stock market prediction.

5. Conclusion

This paper presents a stock market prediction method using sentiment analysis and financial data, employing an ensemble model of MLP, LSTM, and CNN. Feature engineering and selection were applied to social media and financial data from $AAPL, $CSCO, $IBM, and $MSFT (2015-2019). The proposed SFSD model achieved 74.3% accuracy, outperforming other models in the field.

References