2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)

Stock Price Prediction Using News Sentiment Analysis

Saloni Mohan[1], Sahitya Mullapudi[1], Sudheer Sammeta[1], Parag Vijayvergia[1] and David C. Anastasiu[1,*]

***Abstract*—Predicting stock market prices has been a topic of interest among both analysts and researchers for a long time. Stock prices are hard to predict because of their high volatile nature which depends on diverse political and economic factors, change of leadership, investor sentiment, and many other factors. Predicting stock prices based on either historical data or textual information alone has proven to be insufficient.**

Existing studies in sentiment analysis have found that there is a strong correlation between the movement of stock prices and the publication of news articles. Several sentiment analysis studies have been attempted at various levels using algorithms such as support vector machines, naive Bayes regression, and deep learning. The accuracy of deep learning algorithms depends upon the amount of training data provided. However, the amount of textual data collected and analyzed during the past studies has been insufficient and thus has resulted in predictions with low accuracy.

In our paper, we improve the accuracy of stock price predictions by gathering a large amount of time series data and analyzing it in relation to related news articles, using deep learning models. The dataset we have gathered includes daily stock prices for S&P500 companies for five years, along with more than 265,000 financial news articles related to these companies. Given the large size of the dataset, we use cloud computing as an invaluable resource for training prediction models and performing inference for a given stock in real time.

***Index Terms*—stock market prediction, cloud, big data, machine learning, regression.**

I. Introduction

There are many factors that influence stock market prices. One of those factors is investor's reaction to financial news and day to day events. Nowadays, news availability has increased dramatically. It is hard for investors to decide the trend of stock prices based on the huge amount of news. So, an automated system to predict future stock prices will be helpful for investors. An automated system can gather financial news related to the companies of interest in real time and can execute a machine learning model on those data, along with historical stock price information, to predict price.

For years, research has been done on predicting stock prices either based on historical stock price data alone or by using textual data and historical data [1], [2], [3], [4], [5], [6]. Some of the previous works used Twitter sentiments, financial blogs, or news articles as the textual data. In our work, we use financial news articles from well-known sources to avoid fake news that may be prevalent on social media. We have used past stock prices and current day financial news to predict current day closing stock price. We believe that this approach is better because financial news related to the company has a significant effect on its stock price. Hence, taking into consideration the financial news of a company instead of only considering the past stock prices can lead to better prediction results.

II. Related Works

Most of the stock prediction approaches have been built on technical and fundamental analyses of stocks. In recent studies, it has been evident that there is a strong correlation between news articles related to a company and its stock price movements.

Alostad and Davulcu [1] used hourly stock prices of 30 stocks and online stock news articles from the NASDAQ website. They collected tweets related to those 30 stocks for a period of six months. Li et al. [2] collected five years of Hong Kong stock exchange data. They gathered financial news articles over the same time period in order to draw a correlation between the news articles and stock market trends. In a particular trading day, they collected the open, high, close, and low prices of stock for each company.

Collected articles have been processed in different ways to extract features. Alostad and Davulcu [1] extracted N-gram features, after removing stop-words, whitespace, punctuations, and numbers. They built final features into a document matrix and used OpenNLP to extract sentences from each document. They then used the SentiStrength library with Loughran and McDonald Financial Sentiment Dictionaries on those sentences to detect sentiment. Removal of HTML tags [3], tokenization of sentences [3], noun phrasing [7], document weighting [3], TFIDF [3], and extraction of named entities are some of the text pre-processing steps used in various papers.

Alostad and Davulcu [1] solved the stock price trend prediction problem as a classification problem. They used logistic regression to the n-gram document matrix, stock price direction for each hour, and weight of each document. Later, they used SVM to perform the classification. The experiments also showed that extracting document-level sentiment does not significantly increase the prediction accuracy. In various other works, random forest, naive Bayesian, and genetic algorithms were used for stock price/direction prediction.

III. Data Collection

We have collected two different datasets for this research.The daily stock price dataset consists of closing stock prices of the Standard and Poor's 500 companies, from February 2013 to March 2017. We also collected news articles for the S&P 500 companies from February 2013 to March 2017 from international daily newspaper websites. The total number of articles collected is 265463. The challenging aspect of stock price prediction is making use of available data to make an informed decision. A lot of data is being generated for many companies and, if these data were to be processed manually,

[1] Computer Engineering, San José State University, San José, CA
[*] Corresponding Author, david.anastasiu at sjsu.edu

978-1-7281-0059-3/19/$31.00 ©2019 IEEE
DOI 10.1109/BigDataService.2019.00035

it would be difficult to arrive at a decision in time. So, we make use of deep learning models to process the data as it is being generated. We created a web scraper to collect the data, textual information and stock prices, from the web. This raw data is fed to a data pipeline which processes the data and sends them to the machine learning engine, which identifies sentiments in the provided texts and their effect on stock prices or predicting the future price of the stock.

IV. Data Processing

This section will provide details about the pre-processing steps followed before inputting data to the models. For stock prices, we explored the data to check for the presence of duplicate records and null values and did not find any. We applied a log transformation on the stock prices to reduce the difference between high and low stock prices. We also used differencing to transform the stock price series into a stationary series of price changes.

We removed HTML tags from textual data. In order to capture the context effectively from news articles, we extracted five sentences surrounding the name of the company in articles in which the name of the company appeared. For articles only containing the name of the company in the title, we considered the whole article.

V. Evaluation Methodology

Mean Absolute Percentage Error (MAPE) is a popular measure of the prediction accuracy of a forecasting model. It has a very intuitive interpretation in terms of relative error, represented mathematically as:

$$MAPE = \frac{100}{n} \sum_{j=1}^{n} \left| \frac{y_j - y'_j}{y_j} \right| \%,$$

where y_j is the observed value and y'_j is the prediction for the corresponding observed value. It reports the average relative error of the predictions, as a percentage.

MAPE is useful while evaluating prediction models where only the magnitude of the difference between predicted values and observed values is important to consider while the direction of the difference can be ignored [8]. Evaluation using MAPE overcomes the large deviation bias present in Root Mean Square Error (RMSE) and shows robustness for datasets containing long tails.

VI. Experiments

In this section, we discuss the models we trained, their optimization and parameter tuning. In our experiments, we divided the dataset into 90% train, 5% validation, 5% test.

A. ARIMA

ARIMA is the most commonly used model for time series analysis. It consists of three parts: autoregressive (AR), integrated (I), and moving average (MA) models. $ARIMA(p, d, q)$ is the notation of ARIMA model, where p, d, and q are parameters for the AR, I and MA models, respectively. The main application of ARIMA models is short term forecasting, which makes it one of the best models to apply on the stock price prediction problem.

We performed a meta-parameter grid search to find the best values for parameters p, d, and q. We used the Akaike information criteria (AIC) to choose the best trained ARIMA model. AIC is widely used to measure the performance of statistical models. It quantifies goodness of the fit of a model. Models with lower AIC are better. We passed a range of (0, 2) to the grid search function and finally considered the parameters which gave the lowest AIC value. We tried different models by changing the number of observations. Finally, we predicted stock prices for different lengths of the test dataset, namely 5 and 10 days.

B. Facebook Prophet

In this experiment, we fit the model with the data consisting of stock prices of each company for the years 2013–2016 in order to predict stock prices of each company for the next year. The model is a single-variate time series regression model that is able to capture seasonality very effectively but cannot use any side-information in the prediction. We dealt with overfitting and underfitting in Facebook Prophet with the help of a parameter called as *changepoint prior* scale, which is set at a default value of 0.005. Facebook prophet model while getting trained detects many changepoints at which the rate is allowed to change. It then puts a limit on the magnitude of the rate change, which is called as sparse prior. Decreasing the changepoint prior scale below a certain value can lead to underfitting. The Facebook prophet model delivers the best result when the yearly and daily seasonality is enabled and changepoint prior scale is maintained at its default value of 0.005. The performance of the model declines if the seasonality factor is disabled or the value of changepoint prior scale is increased or decreased.

C. RNN LSTM

Unlike ARIMA, the Recurrent Neural Nework (RNN) model does not impose restrictions, like stationarity, on the input. The RNN Long Short Term Memory (LSTM) architecture is useful for modeling arbitrary intervals of information.

1) Approach 1 - RNN LSTM with Stock Prices: To model a regression problem, we predicted the stock price at time t as a function of stock prices at $t-1$, $t-2$,..,$t-m$, where m is the window size of past stock prices. We used a sliding window with a step size of 1 across the training set to learn model parameters. We built LSTM model by experimenting with the history window size, number of LSTM layers, units in each layer and dropout percentage. During model fitting, we analyzed the results for different batch size and epoch parameters and using the ADAM or RMSprop optimizers during the training. For all approaches, we predicted stock price differentials for a given time period and applied inverse transformation to generate the predicted prices.

2) Approach 2 - RNN LSTM with Stock Prices and Textual Polarity: After preprocessing the textual data collected for S&P500 companies, we used natural language toolkit (NLTK) libraries to derive the sentiment or polarity of each document. The sentiment library gives positive, negative, and neutral values as output. In this experiment, we considered positive and negative values only. The final polarity of a company was calculated by identifying the maximum absolute positive or negative sentiment in each text and averaging the identified polarities across all texts identified for a company, i.e.,

$$Polarity_i^c = (+/-)\max(\text{abs}(N_i^c, P_i^c)),$$

$$Polarity^c = \frac{1}{k}\sum_{i=1}^{k} Polarity_i^c,$$

where N_i^j and P_i^j are negative and positive values corresponding to words in the ith of k documents about company c.

We trained the model based on current polarity and previous stock prices, i.e., we predicted the closing stock price at a given time t (Price$_t$) by considering pairs (Price$_{t-1}$, P$_t$), (Price$_{t-2}$, P$_{t-1}$), ..., (Price$_{t-m}$, P$_{t-m+1}$). We transformed and normalized data as described in Section VI-C1. We tuned various parameters of the RNN LSTM model, such as the number of LSTM layers, the number of units in each layer, the batch size, and the number of epochs. We used the RMSprop optimizer and linear activation to fit the model.

3) Approach 3 - RNN LSTM with Stock Prices and Textual Information: Instead of just identifying the sentiment of the text, as in the previous method, we processed the whole text and fed it as input to the neural network along with the price. We computed a linear combination of tf-idf weights with the word2vec representation of words in the documents, which we then provided as input to a convolutional neural network. The output of this neural network is a 10 dimensional vector which is in turn given as input to the recurrent neural network (RNN) along with the normalized price. We designed the convolutional network in the following sequence: two convolutional layers, max pooling, one convolutional layer, global average pooling2D, dense layer, dropout, and finally two dense layers with ReLu as the activation function and a dropout rate of 0.4.

4) Approach 4 - RNN LSTM Multivariate model: In this experiment, we processed the textual data the same way as in Approach 2, using NLTK libraries, but constructed combined daily samples that included stock prices and sentiment polarity for all companies. We used four consecutive days of samples as a window with the fifth day's stock prices as expected output. Stock prices were normalized across the companies. We built this model in order to verify if there is any effect or influence on the stock price of a particular company due to the stock price changes of other companies. The output of this model is stock prices of all companies in the dataset in a single day.

VII. Results and Analyses

In this section, we discuss the results obtained using our models. Table I shows the labels we use to represent models in the result graphs and table. RNN models performed well when compared to traditional models like ARIMA and Facebook Prophet. The RNN-pt model performed well for stable stocks and was able to follow their trends, but performed poorly for low price and high volatile stocks. The RNN-pp model gave the best results across all experiments. This model performed well for companies for which we had more textual data. Fig. 1 shows the stock price predictions for Apple for all models. Additionally, Fig. 2 shows the average MAPE across all companies for each model. Most RNN approaches perform very well, with MAPE values between 2.03 and 2.17. Classical time series models like ARIMA and Facebook Prophet obtain much lower scores, between 7.39 and 7.98. The models that used text (financial news articles) as part of the input have performed very well, while models that predicted future stock prices only on the basis of historical stock prices lead to high percentage error. Table II shows a comparison of these results for five major companies. There were more documents collected for these companies as compared to other companies, and the high number of news articles contributed to achieving low MAPE for these models. Finally, the global RNN-mv model performed poorly overall, indicating that the individual signal for each company is more important towards predicting its performance. More work is needed to incorporate global stock price information into a local company-specific prediction model.

TABLE I
LABELS TO REPRESENT VARIOUS MODELS

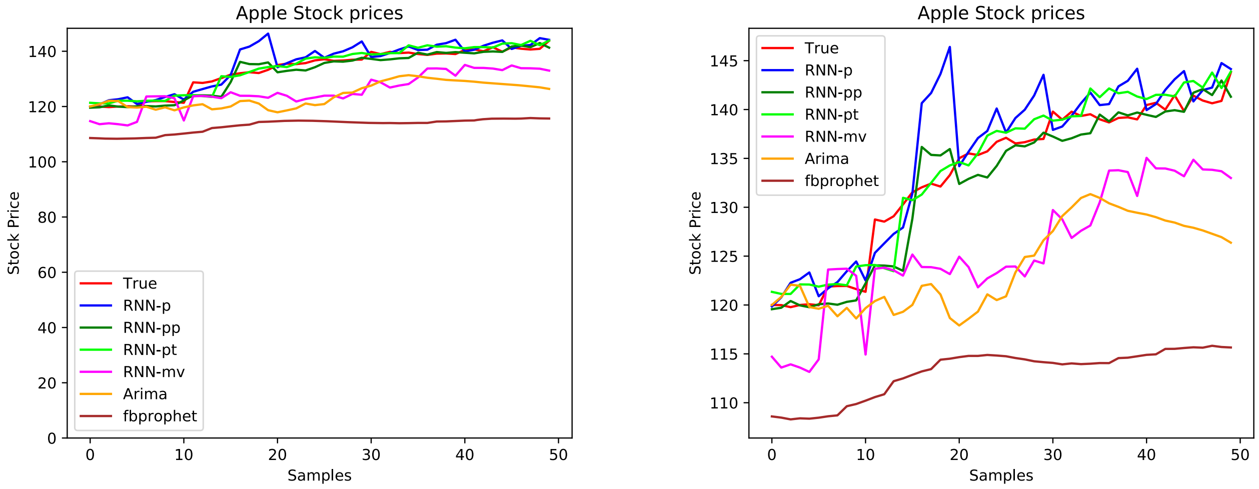
label	Model
fbprophet	Facebook Prophet
RNN-p	RNN-LSTM model with prices as input
RNN-pp	RNN-LSTM model with prices and text polarity as input
RNN-pt	RNN-LSTM model with prices and text as input
RNN-mv	RNN-LSTM multivariate model

VIII. Conclusion and Future Work

In this work, we predicted stock prices using time series models, neural networks, and a combination of neural networks and financial news articles. The results suggest that there is a strong relationship between stock prices and financial news articles. We built prediction models based on time series forecasting models, such as ARIMA, RNN, and Facebook Prophet. We achieved better results with RNN and found that there is a correlation between the textual information and stock price direction. The models did not perform well in cases where stock prices are low or highly volatile. There are still different ways to build stock prediction models, which we leave as future work. Some of these include building a domain-specific model by grouping companies according to their sector, considering adverse effects on the stock price of a company due to news about other related companies, and considering more general industry and global news that could indicate general market stability.

References

[1] H. Alostad and H. Davulcu, "Directional prediction of stock prices using breaking news on twitter," in *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, vol. 1, Dec 2015, pp. 523–530.



(a) Apple: Stock prices plotted on zero axis.

(b) Apple: Closer perspective of stock prices.

Fig. 1. Stock price predictions using multiple models

TABLE II
MODEL PERFORMANCE FOR A FEW COMPANIES AND ALL MODELS

Company	ARIMA	Facebook Prophet	RNN-p	RNN-pp	RNN-pt	RNN-mv
Apple	7.37	14.08	1.98	1.12	1.17	5.93
Amazon	4.88	5.12	1.52	1.25	1.6	1.58
American Airlines	17.38	10.9	3.18	3.11	3.02	9.8
Facebook	15.0	8.05	1.17	2.43	1.13	2.54
Microsoft	3.89	6.64	1.34	1.27	1.53	3.9

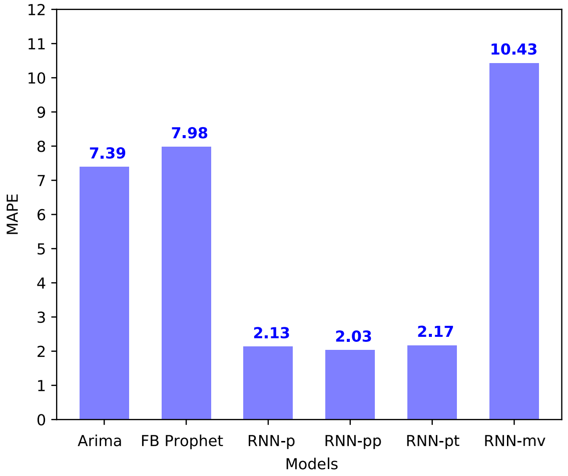


Fig. 2. MAPE for all models

[2] X. Li, H. Xie, Y. Song, S. Zhu, Q. Li, and F. L. Wang, “Does summarization help stock prediction? a news impact analysis,” *IEEE Intelligent Systems*, vol. 30, no. 3, pp. 26–34, May 2015.

[3] D. Duong, T. Nguyen, and M. Dang, “Stock market prediction using financial news articles on ho chi minh stock exchange,” in *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication*, ser. IMCOM ’16. New York, NY, USA: ACM, 2016, pp. 71:1–71:6. [Online]. Available: http://doi.acm.org/10.1145/2857546.2857619

[4] Y. Wang, D. Seyler, S. K. K. Santu, and C. Zhai, “A study of feature construction for text-based forecasting of time series variables,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ser. CIKM ’17. New York, NY, USA: ACM, 2017, pp. 2347–2350. [Online]. Available: http://doi.acm.org/10.1145/3132847.3133109

[5] Y. Shynkevich, T. McGinnity, S. A. Coleman, A. Belatreche, and Y. Li, “Forecasting price movements using technical indicators: Investigating the impact of varying input window length,” *Neurocomputing*, vol. 264, pp. 71 – 88, 2017, machine learning in finance. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231217311074

[6] H. D. Huynh, L. M. Dang, and D. Duong, “A new model for stock price movements prediction using deep neural network,” in *Proceedings of the Eighth International Symposium on Information and Communication Technology*, ser. SoICT 2017. New York, NY, USA: ACM, 2017, pp. 57–62. [Online]. Available: http://doi.acm.org/10.1145/3155133.3155202

[7] R. P. Schumaker and H. Chen, “Textual analysis of stock market prediction using breaking financial news: The azfin text system,” *ACM Trans. Inf. Syst.*, vol. 27, no. 2, pp. 12:1–12:19, Mar. 2009. [Online]. Available: http://doi.acm.org/10.1145/1462198.1462204

[8] A. de Myttenaere, B. Golden, B. L. Grand, and F. Rossi, “Mean absolute percentage error for regression models,” *Neurocomputing*, vol. 192, pp. 38 – 48, 2016, advances in artificial neural networks, machine learning and computational intelligence. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231216003325