Comprehensive Stock Price Prediction

Rohit K Sharma
rsharma54@wisc.edu

Chao Hsiang (Sean) Chung
seanchung@cs.wisc.edu

Akshata Bhat
akshatabhat@cs.wisc.edu

Aditya Rungta
arungta@wisc.edu

Abstract

This report is an attempt to exploit the news articles and predict short term stock price movements. It shows how a combination of stock market history, and the relevance of the news articles to the financial context can be used to label the news articles. Then, these labels are used to perform adjustments on the trained time-series predictions of the future stock price. The time-series model is trained by using the historical stock prices as an input to a recurrent neural network. The main idea of the report is to show that the difference between the time-series predicted stock price and the actual stock price can be explained by the news articles. Therefore, the objective is to analyze and extract such information, and derive numerical indicators from the news text. For prediction of stock price for a day, the news articles of that day are analyzed and the combined effect of them is realized in the time-series prediction to compute the final prediction. It is also shown how the model can be used in a financial market setting to generate profitable results.

1 Introduction

Prediction of stock market trend is a very difficult problem in general because of its volatile nature. Another reason is that there are many reasons that are responsible to determine the price of a stock. News always accounts for the significant amount of importance when investors and stock analysts evaluate and trade stocks. Actually, news contains information which may also influence the confidence and expectations of markets. Namely, the news is an obvious and easily accessible resource for realizing and predicting market condition in the present and future. In addition, some of the news may not seem to be related to the market; however, they may somehow affect the market through a series of butterfly effects. Because it is very easy to obtain huge amount of unstructured news online, exploiting the data and analyzing it can be beneficial for the prediction of stock prices. Therefore, all the articles should be taken into consideration for analysis.

There has been a considerable amount of work done to predict stock prices in the past. Nonetheless, they predict merely based on financial news articles. In this project, we trained two independent models: one to analyze the importance of news articles and their impact on the stock market, and second to perform time-series prediction of the stock price. We adopt Long Short-Term Memory (LSTM) approach to analyzing textual statements in news articles and combine with time series to predict stock price movements. Finally, it is shown how these two models can be combined and be used in a trading simulation setting to earn profits.

2 Related Work

Previously, there have been attempts to perform stock price predictions using the financial news articles. But the existing literature typically relies on classifying financial textual news using very simple text representations such as bag-of-words [7], which are mostly created based on dictionaries. Others use simple classification algorithms such as Support Vector Machines to classify the news articles [6]. These approaches rely on statistical models such as frequency-based or minimum occurrence based tools to classify.

We argue that current techniques severely restrict the power of the text mining and do not exploit state-of-the-art methods. Thus, we expect potential for improvement with respect to the following areas: Extracting the semantics of the news articles by utilizing existing labelled data sets and quantifying the importance of news articles depending on their category and the mention of financially renowned influential people and advisers. Some state-of-the-art methodologies such as LSTM neural networks are used to perform these tasks.

In previous researches such as [1], a simple and intuitive technique to label the news articles as positive and negative is used. In this technique, the news articles are labelled automatically based on the market feedback. Even though the success rate of the overall system may be somewhat low when compared to manual labelling as not all news articles may be positive or negative on a particular day, this technique still works well in most of the scenarios. The combination of the labels with this methodology along with the methodologies discussed above give us fairly decent results. A trading simulation based on back-testing results clearly indicates that the trading strategy exploiting this model is profitable.

3 Model

3.1 Data Set

Broadly, two classes of data sets are used to build the model. First, unstructured news articles and second, historical Market trend.

News Articles

'*News Articles*' data set was collected from Kaggle data set. The data set consists of 143,000 articles dated from 2007 to 2017, from various publications - the New York Times, Breitbart, CNN, Business Insider, the Atlantic, Fox News, Talking Points Memo, Buzzfeed News, National Review, New York Post, the Guardian, NPR, Reuters, Vox, and Washington Post.

To extract the financial relevance of the news articles, two labelled data sets were used:

'*Reuters 21578*' is a benchmark data set used for text classification, where the documents are Reuters newswire stories. It is a multi-class and multi-label data set. It has 90 classes, 7769 training documents and 3019 testing documents. Each of these categories belongs to one of the 5 category set (Exchanges, Orgs, People, Places, Topics).

'*BBC*' data set consists of 2555 documents from the BBC news website corresponding to stories in five categories – business, entertainment, politics, sport and tech.

Stock Market History

'*Dow Jones Industrial Average*' (DJIA) price history. The data set was obtained as a comma-separated values (csv) file from Yahoo Finance. The data set consists of various Dow Jones market parameters such as opening price, closing price, high, low, and so on for each week day since 29th January 1985.

Below is a sample of the first few lines of the file:

```
Date,Open,High,Low,Close,Adj Close,Volume
1985-01-29,1277.719971,1295.489990,1266.890015,1292.619995,1292.619995,13560000
1985-01-30,1297.369995,1305.099976,1278.930054,1287.880005,1287.880005,16820000
1985-01-31,1283.239990,1293.400024,1272.640015,1286.770020,1286.770020,14070000
1985-02-01,1276.939941,1286.109985,1269.770020,1277.719971,1277.719971,10980000
1985-02-04,1272.079956,1294.939941,1268.989990,1290.079956,1290.079956,11630000
...
```

3.2 Architecture

The high-level architecture of the system is shown in Figure 1.

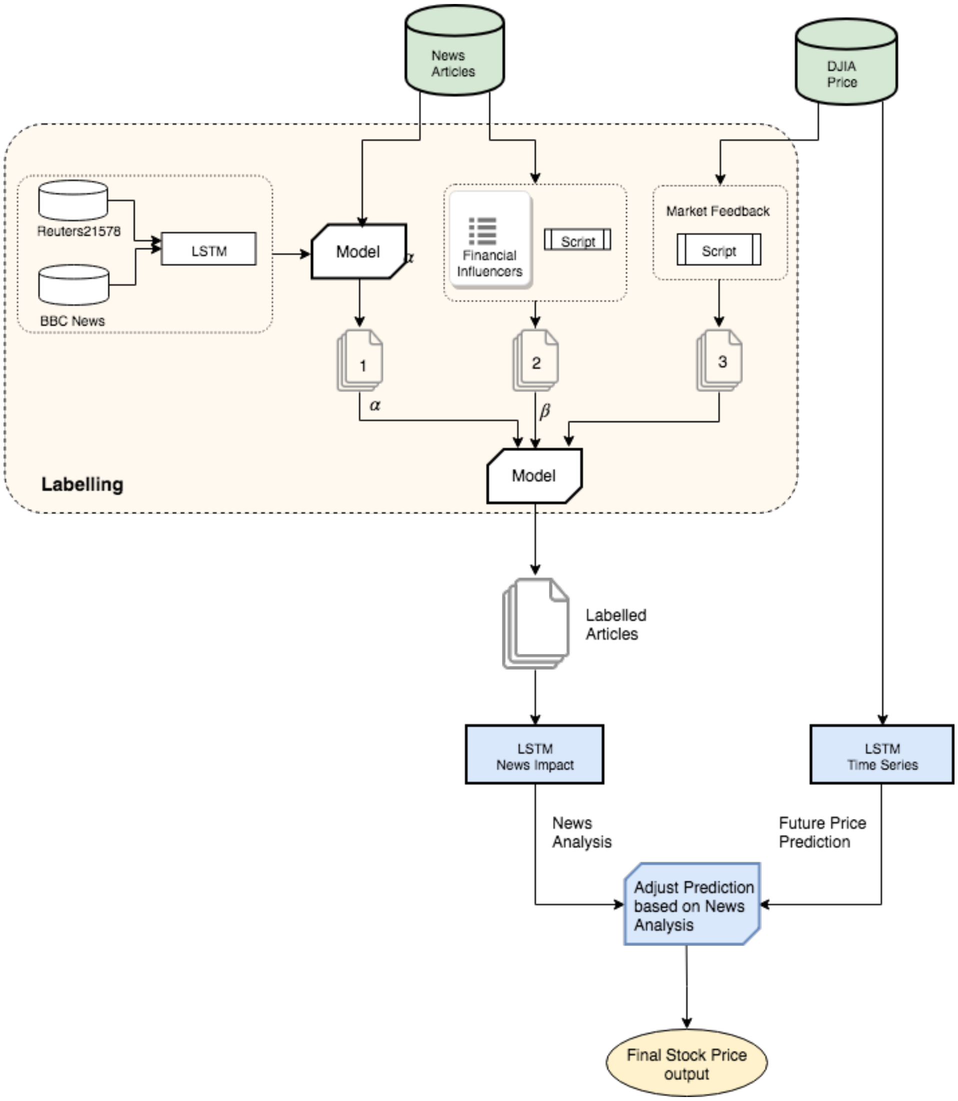

Figure 1: High-level architecture

'*Labelling Module*' - The news articles are labelled using three methods - '*Relevance of news articles*', '*Influential People*' and '*Market Feedback*'. '*Relevance of news articles*' predicts the output as 0 or 1 (to indicate high relevance or less relevance). If the article is highly relevant, α is assigned as its value, $1 - \alpha$ otherwise. '*Influential People*' module labels the articles as β or $1 - \beta$. If the article is highly relevant, β is assigned as its value, $1 - \beta$ otherwise. Market Feedback module outputs '-1', '0' or '1' (to indicate negative impact, neutral or positive impact). Further, output of these three models are multiplied, and this forms the label for the News Articles data set.

'*News Impact*' Module - Labelled News Articles are used to train the model to classify the articles into positive, neutral or negative.

'*Time series*' Module - Using DJIA price history, a time series model is trained to predict future stock price.

Using the output of above the modules, the final stock price is predicted.

3.3 Pre-processing tasks

Before feeding data from raw data set (news articles) into the LSTM model, it has to be pre-processed by:

a. Tokenization, which breaks down the sentence into unique words. E.g., "AI is the future and ML is the future" becomes ["AI", "is", "the", "future", "and", "ML"]
b. Indexing, which makes the words in a dictionary like structure and assigns each of them an index. E.g., {1: "AI", 2: "is", 3: "the", 4: "future", 5: "and", 6: "ML"}
c. Text to Sequence, which represents the sequence of words in the comments in the form of index, and feed this series of index into LSTM model. E.g., the original sentence becomes [1, 2, 3, 4, 5, 6, 2, 3, 4]

After converting articles from text to sequence, they are represented using word embedding to reduce the model size and high dimensionality. The output of the Embedding layer is a list of coordinates of words in the (word) vector space. Thus, the distance of these coordinates can be used to detect relevance and context.

3.4 Pre-training tasks

3.4.1 Labeling the news articles

In previous research work done in this area, two approaches have been majorly used in labeling the news articles. One approach is to label to articles manually. But this approach is very time consuming and is not practical for a large data set of news articles. The other approach is to label articles based on their effects on stock market. We have used this approach for labeling articles in our project.

Relevance of news articles To classify the news articles into more relevant and less relevant, we trained a model using two data sets - Reuters 21578 and BBC News.

For Reuters 21578 data set, among the 5 categories we have picked those that seem relevant – all the categories from ‘Exchange’ set, and some of the categories from ‘Orgs’ and ‘Topics’ set. All the articles that contain these tags are labelled as more relevant and the rest as less relevant, and this binary labelled data set is used to train the model (Model architecture is mentioned in Figure 2).

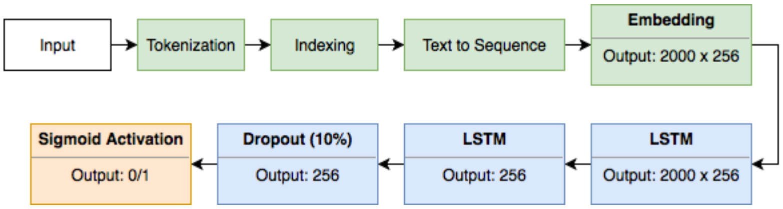

Figure 2: Architecture - Classifying into financially relevant articles

Pre-processing (Tokenization, Indexing, Text-to-sequence and Embedding - size 256) is done on the input data. The model consists of two layers of LSTM followed by a Dropout Layer(10%). The final layer consists of a sigmoid activation function that outputs label ‘0’ or ‘1’. We use this trained model to label our news data set as ‘0’ or ‘1’.

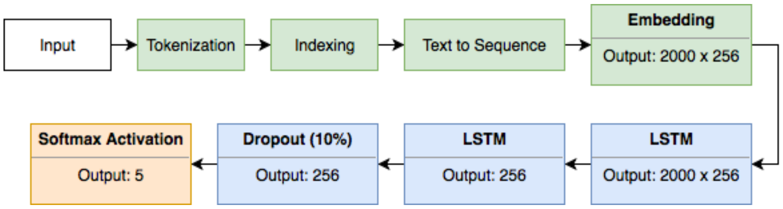

Figure 3: Architecture - Classifying into financially relevant articles

The model architecture as mentioned in Figure 3 is used for training on BBC data set. It is similar to the one described above with one difference. The final layer here uses softmax activation instead and predicts probability of each class. Using this trained model, we classify our news data set (class with highest probability is selected). If the article is classified as business or politics we label it as '1' and otherwise '0'.

The predictions using both the above models are combined to label our news data set (i.e. for a given article, if either one predicts '1', it is labelled as '1').

Influential people We created a dictionary of influential people in the world from various sources. If the article refers to any person from this dictionary, the article is given more importance. We assign a high weight to the article in this case. By doing this, we are assigning some relevance to the articles which are important but not captured by the approach above.

Market feedback For labelling the effect of news articles on stock market, we compared the closing prices of the stock market for the previous two days. If the closing price is increasing, we set the label for news articles on that day as positive. Otherwise we set the labels as negative. The logic behind this is that the news articles that are published will discuss the current trend of the stock market. Such news articles will influence people's actions and consequently the stock market.
Example:
Let's say today is t_d and yesterday is t_{d-1} and day before yesterday is t_{d-2}.
We compute the difference between closing stock prices of yesterday and day before yesterday.
$diff = price_{t_{d-1}} - price_{t_{d-2}}$
If diff is positive the articles on t_d are labeled as positive, else negative.

Combining labels with relevance We have the {positive, negative} label for each article and the relevance of articles for predicting stock market movement based on different criteria. We combine the labels of articles with their relevance to come up with the weighted label. For combining, we multiply the label of the article with the weight assigned to the article based on both the criteria. Once we have the weighted label, we use it for training our LSTM. After a lot of tuning, we chose the weight value(α) of 0.7 for relevant articles and 0.3 for irrelevant articles, and weight value(β) of 0.8 for relevant articles and 0.2 for irrelevant articles. Table 1 gives an example for calculating the label.

Label	Relevance1	Relevance2	Weighted label
1	0.7	0.8	0.56
-1	0.3	0.8	-0.24

Table 1: Example illustrating the Labelling

Let's consider the first row in the table 1 above. The article was labelled positive based on market feedback. In order to know the relevance of the article for stock market movement, we used our approaches based on news category and influential people to get the relevance weights. The final weighted label is computed by multiplying label with relevance.

3.5 News Data Analysis

In this section, as input data (news articles) is natural language which is similar to time series, LSTM is an ideal model dealing with it. Also, natural languages are usually inter-weaved, instead of being purely chronological, we decide to apply bi-directional LSTM [8].

When LSTM is done, we will receive a vector $y_0, y_1, ..., y_t$, where t is the size set to be the embedding size. This vector is the input for a neural network, and the output of the neural network will be the multiplication of importance, market feedback and influential people.

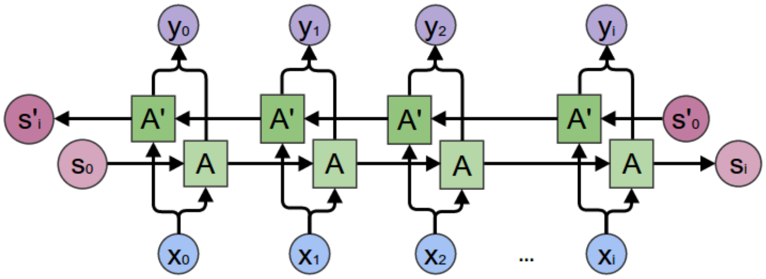

Figure 4: Unlike one-directional LSTM which only depends on previous result, outputs of bi-directional LSTM y_t are influenced by x_{t^*}, where $t^* \neq t$. That is, for an output of a word, it is affected by both past and future words

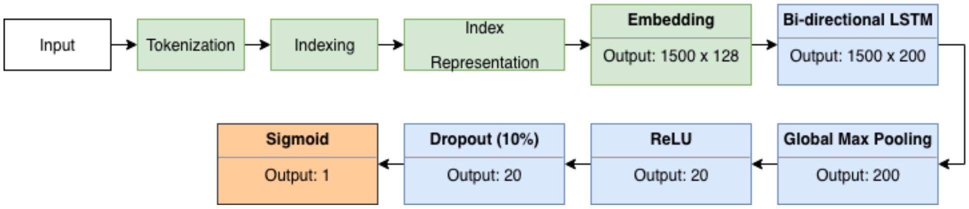

Figure 5: The model structure for News Data Analysis. Pre-processing (Tokenization, Indexing, Text-to-sequence and Embedding) is done on the input data. The model consists of one layer of LSTM, one layer of Global Max Pooling (to convert from 2-D to 1-D), one Dropout Layer (10%), and followed by a ReLU layer in between. The final layer uses sigmoid activation and predicts probability of each class.

3.6 Stock Market Analysis

In this section, we describe the analysis of DJIA price history. The task was to predict the closing price of DJIA for the future days given the past history.

Pre-processing The time-series prediction requires a data-point for every date in the sample. Since we don't have data for weekends, we perform some data imputations such as linear interpolation for all the missing data-points. We also discard all the other columns as we only need to predict the closing price in the future. Now our csv has two columns, namely, date and closing price of DJIA.

As there is a sequence dependence among the input variables (today's price depends on the history of prices in the past), time series modeling is performed to learn this dependence. An LSTM network which is a recurrent neural network is a powerful tool designed to handle sequence dependence, especially when there are time lags of unknown size and duration between important events, and hence we will be using the same.

The architecture of the model used is shown in Figure 6. (Its a double stacked LSTM layers with output from first LSTM at each time step being fed to the second LSTM)

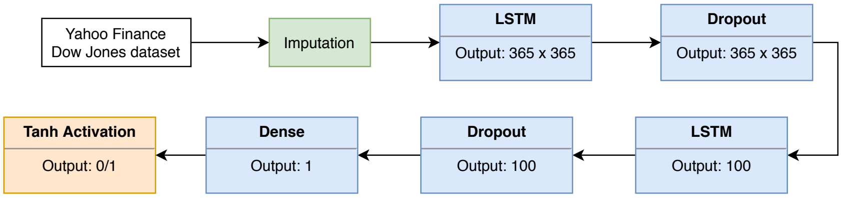

Figure 6: Time-series architecture

The plot of predictions vs actuals almost overlaps with each other to the extent that we cannot distinguish the green curve and red curve as seen in the plot 7

The above is usually not a realistic way in which predictions are done, as we will not have all the future window sequences available with us. So a more realistic way is to predict one time step at a time and feed that prediction back into the input window at the rear. The farther we predict in time, the more the error builds up. But, in our scenario, we can still do this as we only need to predict next days closing price based on the history and so we don't need to go very further in time.

4 Experiments and Results

4.1 System Details

All the models were developed in Python using the Keras deep learning library with Tensorflow backend. The device that was used to perform training had hardware specifications of an octa-core 1.80 GHz i7-8550 Intel CPU.

In the labelling task, using method-1: For Reuters data set, we used 80-20% split for training and cross-validation. It took around 6 hours to train the LSTM model with 4 epochs. The validation accuracy was around 0.9265. For the BBC data set, we used 80-20% split for training and cross-validation. It took around 5.5 hours to train the LSTM model with 3 epochs. The validation accuracy was around 0.854. The News Articles data set was classified into high and low importance based on Method-1 and Method-2 (Influential People), and the number of articles classified into each category are tabulated in Table 2.

	Importance	
	High	**Low**
Method-1	28329	114238
Method-2	72546	70021

Table 2: Frequency based on Method 1 & 2

Using the market feedback, we labeled the news articles as positive or negative, and the number of articles classified into each category is tabulated in Table 3 .

	Market Feedback	
	Positive	**Negative**
Method-3	76994	65573

Table 3: Frequency based on Market Feedback

For training the LSTM model on labelled news articles, we partitioned 15% of training data as the validation set, and it took our model around 6 hours to finish training on 5 epochs, with the validation loss (mean squared error) = 0.03.

The LSTM time-series model was trained for 4 epochs. 60-20-20% training-validation-test split was used for the training. The model took around 30 minutes to train and the validation loss was 1132.9.

4.2 Analysis

We used the stock data from 1984 to 2012 for training and the data from 2012 to 2018 as the test set. Mean squared error was used as loss to evaluate the performance of the model.

Following graph (7) shows the plot of predicted and actual stock price against time.

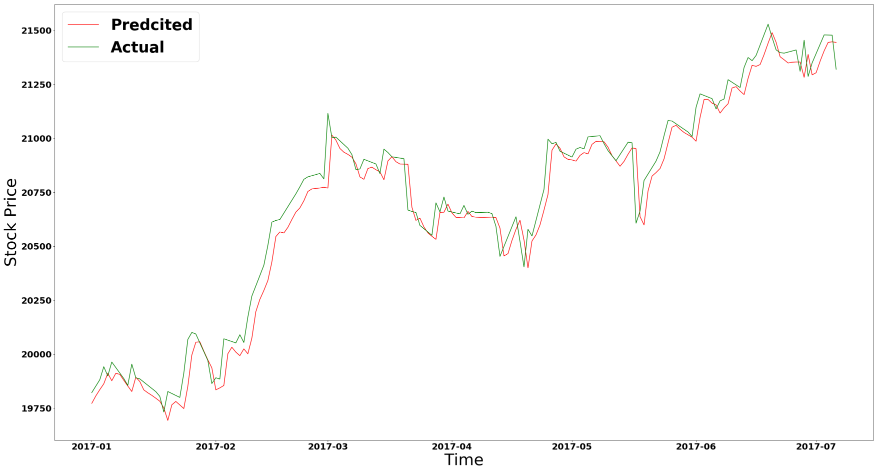

Figure 7: Actual stock prices and predicted stock prices based on historical data

From the above plot, it can be seen that our LSTM model has learned the seasonality of stock market from training data and does a fairly good job on the test data.

We combined the predictions that we obtained from historical stock data and the analysis of news articles. In the following bar graph (8), we have plotted the results after combining. The green bars indicate that the difference between the predicted and the actual stock price is actually reflected by the impact of news articles of that day. That is, both of them either predicted positive or negative. The red bars indicate that the difference between the predicted and actual stock price based on news articles and stock data is contradictory. As we can see from the graph 8, the stock market is mostly in agreement with what news articles have to say. That is, the difference in the predicted and actual values is in fact being realized by the impact of news articles.

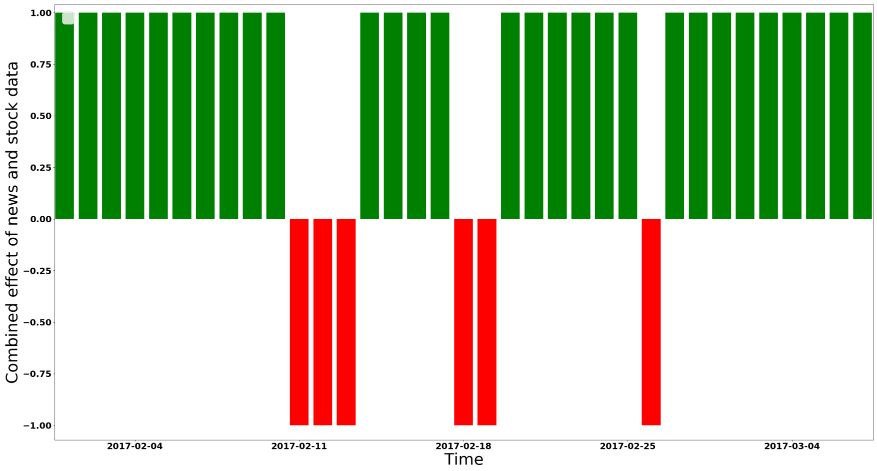

Figure 8: Combination of prediction using news articles and historical stock data

4.3 Trade Simulation

Using the model that was trained, we performed a trading simulation for the period from 01/01/2017 to 07/06/2017. The prediction of the closing prices of the DJIA index using the time series model was used as a baseline price and the adjustment to it was made based on the news analysis. The LSTM model was used to predict the impact of the news articles for each day and their average labelling was taken to determine if the news of that day will have a positive impact or negative. Based on the adjusted predicted price, the decision to buy or sell the stocks is taken.

For the simulation, let the initial number of stocks possessed by the simulator be 100. The strategy used by the simulator is to sell 10% of stocks whenever the prediction of the stock price is lower than the current days price, and recording the amount of profit made on them. If the prediction is greater than the actual, the strategy is to purchase 10% of stocks currently possessed. Finally, the value of stocks possessed on the last day of the simulation are realized into the profit thereby computing the total profit/loss during the simulation.

With the above strategy, our model was able to earn profits of $1,458. Therefore, the strategy is profitable, and the model is useful.

5 Conclusion

The project was based on the hypothesis that the movements in stock prices can be predicted by analyzing the historical stock price trends and news articles. As we see from the results, there is a strong correlation between the news articles and the stock price movement. The predictions closely reflect the real conditions and can in fact be used to earn profits. This implies that a thorough textual analysis of the news articles is beneficial in predicting the stock prices.

In this study, two independent models for time series prediction of stock price and news analysis classification were combined in a very simplistic approach of linear combination with static weights. Instead, it may be possible to improve the performance of the system by using dynamic weights which depend on several factors such as volume of stock traded, importance of news articles vs time series, etc. The trade simulation can be done in a much more complicated way to reflect the actual trading conditions. For example, the risk aversion of the investor could be one of the factors to determine when to sell or buy the stock. There are few other factors as well that could impact the stock prices - public sentiment from social media (Twitter, Facebook, ...), etc. Success quotient of the system can be increased by taking such factors into account.

References

[1] Kaya, M.I., & Karsligil, M.E. (2010). Stock Price Prediction Using Financial News Articles. 2010 2nd IEEE International Conference on Information and Financial Engineering, 478-482.

[2] Hagenau, M., Liebmann, M., Hedwig, M., & Neumann, D. (2012). Automated News Reading: Stock Price Prediction Based on Financial News Using Context-Specific Features. 2012 45th Hawaii International Conference on System Sciences, 1040-1049.

[3] Jahan, I., & Sajal, S. (2018). Stock Price Prediction using Recurrent Neural Network (RNN) Algorithm on Time-Series Data. 2018 Midwest Instruction and Computing Symposium.

[4] Ariyo, A.A., Adewumi, A.O., & Ayo, C.K. (2014). Stock Price Prediction Using the ARIMA Model. 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, 106-112.

[5] Gidófalvi, Győző. (0004). Using news articles to predict stock price movements.

[6] Vakeel, K.A., & Dey, S. (2014). Impact of News Articles on Stock Prices: An Analysis using Machine Learning. I-CARE 2014.

[7] Groth, S.S., & Muntermann, J. (2011). An intraday market risk management approach based on textual analysis. Decision Support Systems, 50, 680-691.

[8] Kiperwasser, E., & Goldberg, Y. (2016). Simple and Accurate Dependency Parsing Using Bidirectional LSTM Feature Representations. TACL, 4, 313-327.

[9] Sundermeyer, M., Schlüter, R., & Ney, H. (2012). LSTM Neural Networks for Language Modeling. INTERSPEECH.