Pattern analysis of stock market trend based on news title

Amit Dutta
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
amitdutta121@gmail.com

A. A. Noman Ansary
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
showrav.ansary.bd@gmail.com

Adib Muhammad Amit
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
amitadib121@gmail.com

Saif Shahnewaz Saad
Department of Computer Science and Engineering
Brac University
Dhaka, Bangladesh
dhrubosaad@gmail.com

Md Sabbir Hossain
Department of Computer Science and Engineering
Brac University
Dhaka, Bangladesh
md.sabbir.hossain1@g.bracu.ac.bd

Annajiat Alim Rasel
Department of Computer Science and Engineering
Brac University
Dhaka, Bangladesh
annajiat@bracu.ac.bd

***Abstract*—In today's world Stock market is seems to be a lot shifty. Common people go through a lot of hassle and find the uneven fluctuation of stock values very complicated. Data on stock market prices are generated in large quantities and varies every second. The stock market is a complex and difficult system in which people can make money or lose their whole life savings. But then again they don't have to put themselves through that if only a proper analysis is done based on past findings. And that's where our insights will come in handy. We have analysed and calculated Risk factors of stocks from past thirteen years of news data available in online market from 2008 to 2021. We scraped the news of World News Subreddit data to help companies and common people decide if they should buy and sell a particular stock at a particular situation. Stock market depends a whole lot on the news since that is the outlook which is apparent to everybody. In the past researches, we have found that Arima Sarima algorithm are used to find trading patterns. But they are not meaningful nor seem to provide enough insight to be able to trade. We will improve that accuracy and result. Our models are built using supervised machine learning methods, to be specific NLP is used to merge and process news of specific dates. Deep learning and its variant recurrent neural network RNN and GRU is implemented to predict stock trend. So far we have come to a satisfying result and it has been able to predict trend of stock market of specific domain. We found this improved result due the utilization of Bidirectional LSTM.**

***Index Terms*—Neural Network, Data Scraping , NLP, RNN, Preprocessing, NLTK**

I. Introduction

Market sentiment is determined by using sentiment analysis to measure the effect of unorganized and unfamiliar information and news on investor sentiments. Previous research has found that the predictability of news influence the market. The goal of this research is to capture stock market fluctuations as soon as a price change occurs. The results of asking for bids are based on the information available. We perform supervised research in this project. Sentiment analysis is done using high-frequency, real-time news data. The goal of this research is to predict the trend of the stock market. When it comes to sentiment analysis, the types of approaches are unsupervised dictionary based and supervised dictionary based approach. Market sentiment is retrieved from news material using an unsupervised dictionary-based technique. To count, a lexicon of sentiment words is needed and the various written representations of sentiments or emotions hints. The term "supervised sentiment analysis" refers to a type of sentiment analysis that is done with the help of past market patterns [6]. Training is necessary for the supervised method. Data can be manually labeled or can be done through an automatic process. Historical information is gathered. Despite the fact that hand labeling has its advantages this approach cannot be used for larger projects due of its higher precision. Sets of data Automatic labeling entails matching the labels in time. With the news, there are some interesting or important market trends that could have contributed to the observed patterns. In this study, we examine the predictability of dictionary-based and non-dictionary-based news predictions based on the sentiment metrics. We assume that markets are inefficient and that adjustment takes time. We've seen a tendency toward being relevant to regularly published news items. According to previous studies on market mood, it takes approximately

20 minutes for information to be reflected in the stock market in the event of a new market.As a consequence, the effect of breaking news on stock prices is much quicker in the context of index attitude. This study employs a variety of methods to investigate this notion. A literature review on sentiment is presented in the next section. In a financial market setting, analysis studies are carried out. This study's approach was followed by data analysis. Lastly, you'll find the results. To the best of our ability research on sentiment analysis is done in the context of foreign stock market.The findings of this study could be valuable in high-tech industries. In order to understand buying and selling market behavior is necessary for predicting future market trend.

II. Literature review

The stock market is perhaps one of the most volatile and ambivalent entities on this planet. So, predicting or finding a way to predict its trends and which way it will go in the future is probably one of the most sought after topics in the field of research. We observed that in most research paper pertaining to this topic, we see many methods and techniques that are innovative and efficient to find the stock market trends. Some of the popular methods are data mining, stemming, text mining of corporate web and time series data, using quotes and financial news, sentiment analysis, RNN, deep learning, LSTM, GRU and etc.There are a myriad of ways to predict the stock market but the research field is leaning in the way of neural networking and deep learning. Hence, the research papers related to this topic is very popular nowadays.

Although there are many techniques and many computational capabilities involved, the most ubiquitous way to predict the stock market trends is parsing through a database containing years of data of the stock market. That database is combed through for certain quotes and certain words those are a dead giveaway of the predictability of the stock market behavior. The database mainly consists of a corpus of news articles and quotes regarding financial news mainly [3]. The combination of news and quotes will be used by the simulation server for the purpose of prediction of the stock price. Then the model is trained via RNN to recognize anomalies in the stock price and the stock market behavior by observing certain keywords, quotes and re-iterate the process over and over again to improve the accuracy of the recognition and prediction of the stock market trend [4].

In some research papers, sentiment analysis has been heavily emphasized upon which seemed to recognize the sentiment of the investors when buying stock using natural language processing sentiment analysis via differentiating positive, negative and neutral statements. In summary, from all the research papers that we have gone through, the most common goal was to find the most efficient way to find and predict the stock market trend by the means of financial news either by data scraping or mining [5]. This elusive topic will always be of interest and will constantly be improved upon in the days to come. We welcome the innovative approaches to predict the stock market trends faster and in a more efficient way. As a result, this research pertaining to the topic of predicting stock market will improve by leaps and bounds.

In most research papers applicable to this topic, we also have determined that machine learning strategies were used to become aware of non-linear dependency in the stock market fee sequences. Although the problem was dealt with upfront, due to the high volatility of the stock market, forecasting the trend of the stock market stays an undertaking [1]. To meet these demanding situations, methods like making use of a convolutional neural network to seize the spatial structure of time collection had been used. This method is extra correct than maximum sign processing methods and frequency trading styles modeling methods with deep gaining knowledge of in-stock market prediction.

There are also morphological approach in many research papers which bring nuance to research related to predicting stock market trends. In these type of research papers, we observed that grammar and multi-word syntax is to be applied alongside the sentiment analysis so that one can find the corresponding relationship between the financial news and stock market volatility [2]. This method is optimal because there are a myriad of news articles to choose data or words form in order to train the model and ultimately increase the F1 score as well as the accuracy.

III. Preprocessing

A. Stopword Removal

Stop word removal is one of the integral steps in preprocessing in a myriad of NLP based applications. In short, stop words are those words which if removed, will have no significant impact on training the model. Stop words are either generally pronouns, articles and etc. or handpicked words i.e. some names and etc. We remove such words to make the data less noisy and to decrease the time it takes to train the model. In our paper, some of the stop words were: i, me, my, myself, we, our, ours, ourselves and etc. This step drastically changes the outcome and therefore this step is intrinsically included.

B. Frequent Words Removal

This step is almost the same as the previous step mentioned but there is a little contrast between these two steps. In this step we count the most frequently used words and least frequent words and then remove them. The advantage of removing them are similar to those of the previous step. Some of the most frequent words that were sighted in our paper are: US, new, says, news, world, police, Russia.

C. Stemming

Stemming is one of the most integral part of preprocessing. In summary, stemming is the process of turning words into their root or base form. Stemming is basically considered an algorithm which reduces words Like "retrieval", "retrieved", "retrieves" reduced to the stem "retrieve". In our paper, we experimented with the PORTER stemmer. This particular stemmer has the function of removing suffixes from words to revert them to the original word. One example is removing

“ing” from “Flying”. This is advantageous in a sense of information retrieval and this obviously shortens the vocabulary space and thus dramatically changes the size of the index.

D. Lemmatization

Lemmatization is the process of grouping together” lemmas” or inflected word forms so they can be scrutinized together as a single item. It is similar to that of stemming but the logic is different and in this algorithm, having detailed dictionary plays a key role in grouping the words and analyzing them in a singular form. Through this algorithm, the trained model learns to link a word to its lemma. For example: “The boy is eating rice” and “The boy eats rice”; here the lemma is “eat”. The verbs will be linked to this particular lemma and will be grouped together for analysis. For out paper, we chose to experiment with the Wordnet which is a well-defined lemmatizer or lexical database enriched with English vocabulary for the reasons mentioned above.

1) URL Removal: There were URLs embedded in the links we chose to extract data from and fortunately the removal of the URLs does not affect the outcome or the training of the model that much. And the removing irrelevant data from all the relevant data is always a welcome approach to make the preprocessing faster and more efficient.

E. Punctuation Removal

Punctuation do not have any role to play in the data that we used to train the model with. Therefore, punctuation are redundant to be kept in the database. Removal of punctuation have the same effect as the effects of removing URLs as mentioned above.

F. Tokenization

Simply put, tokenization is the process of separating pieces of text into smaller chunks of texts or” Tokens” where said “Tokens” can be characters, words or subwords (also known as n-gram characters). We utilized this technique to separate words which are relevant and which are not by assigning weights to the words and turning them into tokens so that the model can analyze and recognize certain keywords much faster. For our paper, we made use of tf-idf algorithm. Word embedding is a mastered portrayal of text where words those have the same meaning have an indistinguishable representation. We utilized word embedding so that we get words that have similar meaning and get similar representations for them. For example, if we have swimming, cycling, jogging, hiking etc. in the same context, in each of their definitions, the common meaning would be “exercise”. Then all these words would be categorized under the same category. It makes the preprocessing way more efficient and it save a lot of time in training the model. That is why this is an integral part of preprocessing. We used tf-idf algorithm to convert the English words into numerical values.

IV. Model Design

A. Input Layer

Input layer delivers the initial data into the system for processing by succeeding layers of artificial neurons. The input layer is the first step in the artificial neural network’s process. Input layers, hidden layers, and output layers are common components of artificial neural networks. Because the input layer is the initial layer of the network, artificial neurons in the input layer have a distinctive function to perform. Then the data is sent from the input layer to the following layers.Before sending the data forward, the input layer runs it via the activation function. The input layer receives the data, conducts the computations using its neurons, and then sends the results to the next layers. Firstly tokenization is done to make the machine understand English language where data is converted to numeric values. And then the converted values are inserted as inputs. The encoding of the sentence takes place in two stages. Those encoded sentences later on are received by embedded layer which relates to headlines from day’s news stories organized in chronological order. To generate word embedding, a model named word2vec is used which can be trained with words in order. This embedding is a one of a kind vector with length of continuous values. The sentence vector is created by averaging all of the word vectors in a title to generate a single vector for the whole title. A single hot encoding technique is used to encode each title in the data set. The produced word embedding vectors from the word2vec model may capture language regularities such as semantic and syntactic regularities, which is a desired characteristic in NLP applications [7].

B. Embedding Layer

High-dimensional and sparse one-hot encoded vectors are the two major reasons why Embedding layer is important. When performing Natural Language Processing using a 2000-word vocabulary each word is represented by a vector of 2000 integers. And there are zeros in 1999 of these numbers. This method is not computationally efficient when dealing with large data sets. During the training of the neural network, the vectors of each embedding are updated. Related words can be discovered in a multi-dimensional space which enables us to see connections between words, as well as anything else that can be converted into a vector using an embedding layer. Word embedding are the source of embedding. The first step in utilizing an embedding layer is to use indices to encode this phrase [8]. Each unique word is given an index in this instance. Lastly, embedding matrix is produced. Each index has a certain number of ”latent factors” allocated to it. This is essentially how long we want the vector to be. Lengths of 32 and 50 are common usage cases. Rather than ending up with large one-hot encoded vectors, we may utilize an embedding matrix to make each vector considerably smaller. A vector does not, however, replace every word. It is replaced with an index that is used to find the vector in the embedding matrix.

When working with large data sets, this is computationally efficient.

C. Recurrent Layer

Recurrent networks are built using layers. The distinct recurrent layers used here are LSTM and GRU. The input structure for recurrent layers is intended to be (batch size, sequence length, num inputs), which is similar to feed-forward layers.

The RNN serves the goal of simulating the features of the input data sentences. A unique Recurrent neural network called LSTM is utilized to add memory cell at this step which is a new and different structure. The output gate, determines whether the cell state has an impact on other neurons. This structure has the benefit of allowing for the modeling of dependencies which is available in the data input. And in the mean time it also avoids the issue of 'Vanishing gradient' [7]. We used a second recurrent multi-layer which is GRU. It is a better version of the conventional recurrent neural network, with its update and reset gates (RNN). In actual applications, RNN can only use information due to the issue of "vanishing gradient" in the network topology. Long Short Term Memory (LSTM) and GRU were provided with a specifically built network architecture that can learn information organically in order to address this issue. They are two vectors that may be taught to retain old or prior knowledge while discarding data that isn't relevant to the forecast.

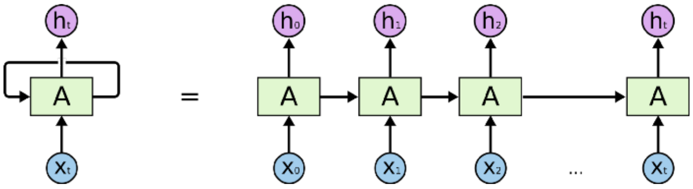


Fig. 1. RNN

V. Experiments

A. Data Description

Two data sets were used for this work : (i) one consisting of various news headlines and (ii) another consisting of stock market indices. These separate data sets were merged into one single data frame to meet our need for workflow.

The dataset compiled for news headlines contains 2,990,931 headlines from various sources such as Reuter, CNN, Fox News, Al Zajeera etc. etc. corresponding to the period from January 26th 2008 to April 28th 2021 was collected from Reddit where the users posted them. As for the stock market indices, Yahoo Finance was put to use for different stock entities for the same time period containing its indicators - High, Low , Close, Adj. Close and Volume. The topics of all news articles are not limited to finance related news, rather they cover many areas.

In the merged data frame, each entry contains a date, news article headlines concatenate together that is published on that date and a decision value (0/1) based on whether it correlates to any price-rate change of the particular stock in the next day. The actual content of the news article has not been taken into account . The reason for using this filter step is that general news can contain a lot of irrelevant information. The authors of [8] confirm this claim. This merged data set would be used for modeling purposes. The training was run on the labelled data before January 1st 2021. And for testing, everything afterwards was used.

B. Data Labeling

To label the data for the merged data set, we chose the Close indicator of Apple Inc and Microsoft arbitrarily. Yahoo Finance provides various indicators but the information collectively forms a candle chart, choosing any indicator over the other does not provide any significant advantage at all. 25 headlines per day were taken into account to find correlation between the event and stock price change in the future. Roughly 5 Thousand data points were generated through this approach for each of the companies. The training and testing were done on these labelled data.

C. Modelling

We have implemented multiple deep learning models/architectures to train and test which are LSTM [9], biLSTM, GRU [10], Random Forest [11] and XGBoost [12]. We have used Long Short-Term Memory (LSTM) modelling. Long Short-Term Memory is capable of learning order dependence in sequence prediction problems. As stock markets are most likely to fluctuate every now and then, so short-term persistence would make the model prone to error for steep changes in rates and anything that does not have the ability to keep persistence for a long time may come up with less accurate predictions. LSTMs can address the issue of frequent fluctuations in stock market prices by keeping the context into account and preserving the information necessary while discarding the unnecessary. Bi-Directional Long Short-Term Memory (biLSTM) has also been used as within the fixed timeframe, the model should have the capability to predict both forward and backward price-rate changes. Using the biLSTM ensures that the model understand that there can be steep changes in price-rates within a time-frame. Similar to LSTMs, another RNN based model, Gated Recurrent Units (GRU) has been put to use. GRU has been used to see if the hidden states alone are sufficient enough for going forward with new inputs to learn from and then predict outcomes. Equation 1 and equation 2 below shows the functionalities of LSTM and GRU respectively:

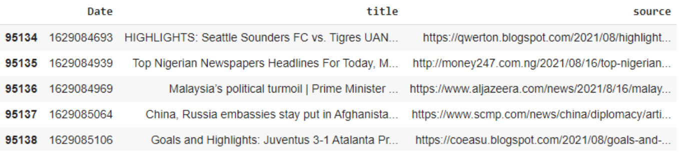
$$\begin{aligned}
\mathbf{i}_t &= \sigma\left(\mathbf{W}_i \cdot \mathbf{h}_{t-1} + \mathbf{W}_i \cdot \mathbf{x}_t + \mathbf{b}_i\right), \\
\mathbf{f}_t &= \sigma\left(\mathbf{W}_f \cdot \mathbf{h}_{t-1} + \mathbf{W}_f \cdot \mathbf{x}_t + \mathbf{b}_f\right), \\
\tilde{\mathbf{c}}_t &= tanh\left(\mathbf{W}_c \cdot \mathbf{h}_{t-1} + \mathbf{W}_c \cdot \mathbf{x}_t + \mathbf{b}_c\right), \\
\mathbf{c}_t &= \mathbf{f}_t \mathbf{c}_{t-1} + \mathbf{i}_t \tilde{\mathbf{c}}_t, \\
\mathbf{o}_t &= \sigma\left(\mathbf{W}_o \cdot \mathbf{h}_{t-1} + \mathbf{W}_o \cdot \mathbf{x}_t + \mathbf{b}_o\right), \\
\mathbf{h}_t &= \mathbf{o}_t tanh\left(\mathbf{c}_t\right).
\end{aligned} \tag{1}$$

$$\begin{aligned}\mathbf{z}_t &= \sigma\left(\mathbf{W}_z \cdot \mathbf{h}_{t-1} + \mathbf{W}_z \cdot \mathbf{x}_t\right),\\ \mathbf{r}_t &= \sigma\left(\mathbf{W}_r \cdot \mathbf{h}_{t-1} + \mathbf{W}_r \cdot \mathbf{x}_t\right),\\ \tilde{\mathbf{h}}_t &= tanh\left(\mathbf{W}_o \cdot \mathbf{r}_t + \mathbf{h}_{t-1} + \mathbf{W}_o \cdot \mathbf{x}_t\right),\\ \mathbf{h}_t &= (1-\mathbf{z}_t)\mathbf{h}_{t-1} + (\mathbf{z}_t)\tilde{\mathbf{h}}_t.\end{aligned} \tag{2}$$

Random Forest and XGBoost are also implemented. Random Forest chooses random time-frame samples from the dataset without any sequential order at all rather trains on the generated datasets in parallel, the model still can make correlations and decision trees are generated using the average. XGBoost does the exact opposite, instead of choosing a random time-frame it sequentially takes data from the stock market and makes a correlation and then predicts the outcome.

VI. Methodologies

The stock market is so volatile that to predict a particular day's price is almost impossible because it is entirely random. There are many supervised learning techniques to forecast the stock market price, but those are not that efficient. Because of its volatility, people cannot predict it with some specific algorithm. There are many investors out there who are making money by buying and selling the stock. As we already mentioned, predicting the stock market is almost impossible, but it is possible to predict the trend. Many factors can change the stock market. One of the main factors is news. So if we use the news to predict the direction, that will be even more accurate. In this paper, we have introduced a model that can efficiently predict the stock market trend.



	Date	title	source
95134	1629084693	HIGHLIGHTS: Seattle Sounders FC vs. Tigres UAN...	https://qwerton.blogspot.com/2021/08/highlight...
95135	1629084939	Top Nigerian Newspapers Headlines For Today, M...	http://money247.com.ng/2021/08/16/top-nigerian...
95136	1629084969	Malaysia's political turmoil | Prime Minister ...	https://www.aljazeera.com/news/2021/8/16/malay...
95137	1629085064	China, Russia embassies stay put in Afghanista...	https://www.scmp.com/news/china/diplomacy/arti...
95138	1629085106	Goals and Highlights: Juventus 3-1 Atalanta Pr...	https://coeasu.blogspot.com/2021/08/goals-and-...

Fig. 2. Data snapshot

At first, we started with noisy news data from the world-news subreddit. After collecting over 14 years of news, we collected over 14 years of stock market data from different companies. Then we need to label the news data with the stock market data. However, as We already mentioned, the data was noisy, so we needed to go through a rough preprocessing. At first, as we can see from the dataset snapshot in "Fig. 2" , we have a column named timestamp that represents a date and time in a singular value, so we first convert the timestamp into date values, then we have multiple news for a single date. We find those dates and combine numerous news headlines into a single header. After combining that news, we have a date and a news column containing multiple news for a single date. This preprocessing was essential because this is the baseline of our dataset. After that, we take our second dataset, and as we can see from the snapshot, we have multiple columns named Open, High, Low, Close, Volume. Then, to label our dataset, we take a single date from our news dataset, find the particular date in our stock dataset, and then find the stock price a day before and a day after that date and calculate the difference. If the difference is positive, we label it with 1. If it is negative, we mark it with 0.

$$Prediction = Previousdaystockprice - Nextdaystockprice \tag{3}$$

Moreover, thus how we label our dataset. Above equation describes how we calculate the trend for current date. We also need to drop those rows whose dates are not present in the stock market dataset. Because news is published every day, but the stock market is not open. And there are specific trading hours in each stock exchange that can affect our data. After preparing our dataset, we now need to convert our news into numerical numbers because machine learning algorithms can only understand numerical values. However, as we already mentioned, our dataset is noisy, so we need to remove the noisy data by further preprocessing the data. First, we take our news data and convert it into lower case so that our model can see the difference between similar words. Moreover, this is more helpful for text featurization techniques like frequency. After converting it into lowercase, we then remove the punctuation from the data. By removing the punctuation, our model can better understand the data. Then there are some stopwords in every language which does not add any significant value to our model. So we remove the stopwords from our dataset, and for the stopwords data, we use nltk stopwords. Then in our dataset, we find some most and least frequent words, which make our dataset noisier. So we remove those frequent words from the dataset. After that, we remove some rare words from the data which are used in the English language. Then we used nltk PorterStemmer to stemming our data, which reduces inflected words in our word stem.

After stemming as we are working with the English language, we used snowball stemmer because porter stemmer changed the meaning of some of the words. In this process, we also try to use Lemmatization, but the results are not great. Even if the results are not great, the observation was well worth the effort. After that, we can see that some authors are using emojis that are not suitable for our model in some news, so we remove those emojis from the data. After all, emojis might have no contribution to the experiment we are conducting. After that, we found that in some news, there are some news links available that are not important for our model, so we removed URLs from the dataset. Not at last to tokenize our final data, we used the tf-idf algorithm we have also used word vectorizer but after training our model with word vectorizer we find out that our model cannot converge properly and the accuracy is not very high then we tried tf-idf by using this tokenization algorithm we get better results from our model. That is how we preprocess our English news dataset. Now In "Fig. 3' we have described our data preprocessing pipeline.

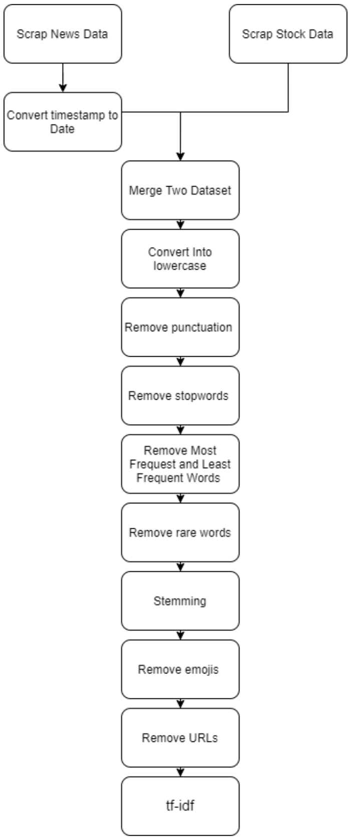


Fig. 3. Preprocessing pipeline.

Now, After preprocessing the data, we no longer have text data. Now we have numerical data. Using this data now, we need to select a model by which we can predict the market trend. At the early stage of model selection, we have used Random Forest to predict the data. After tuning the hyperparameter of the Random Forest algorithm, we find out that the output scores are not good. So we moved on to the Gradient Boosting algorithm and fine-tuned the algorithm's parameters, and the result was the same and not satisfactory. Then we used neural networks, especially recurrent neural networks (RNN), which are good for sequential data. Moreover, we can see that we have sequences of new English words that make excellent sense. So initially, we tried with simple RNN with LSTM models and also experimented with RNN and GRU. Then we also introduced regularization and dropout into our model. After fine-tuning the model for several episodes, we finally found a model that efficiently predicts the market trend. Furthermore, we used binary cross-entropy as a loss function and adam as an optimizer for our neural network. We also used sigmoid as our output function as our output dimension is binary. After experimenting with RNN, we also experimented with BidirectionalRNN, which produces a better result in NLP tasks. In BidirectionalRNN we experimented with the LSTM and GRU layer in BidirectionalRNN and used dropout to avoid overfitting. In "Fig. 4' we plot our final model.

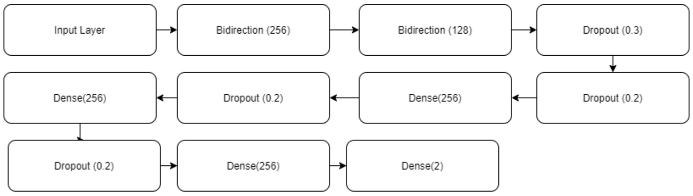


Fig. 4. Final Model

TABLE I
RESULTS OF TREND PREDICTION

Model	**Model**	***Training***	***Test***
RNN LSTM	No	52%	58%
RNN LSTM	Yes	61%	59%
RNN GRU	No	59%	54%
RNN GRU	Yes	63%	61%
Bidirection GRU	No	65%	72%
Bidirection GRU	Yes	62%	71%
Bidirection LSTM	No	65%	75%
Bidirection LSTM	Yes	**69%**	78%
RandomForest	*	50%	56%
XGBClassifier	*	59%	65%

VII. RESULT AND DISSCUSSION

After experimenting with dozens of algorithms and tuning their hyperparameter for a longer time, we can see from Table 1 that using Bidirectional RNN with the LSTM layer and using dropout gave us the most accurate result. In Figure 4, we can see the model architecture, which uses 300 pad sequences as input followed by an embedding layer then a 256 Bidirectional hidden LSTM layer. After that, a dropout of 0.3, then another Bidirectional hidden LSTM layer of 128 then Followed by three 256 dense layers with RELU activation and each layer a dropout of 0.2 was added finally a dense layer of 2 was added because we need to determine that the next day will the trend go up or not.

VIII. CONCLUSIONS AND FUTURE WORK

This work consists of experimenting with various machine learning/deep learning models/architectures. As per our experiments, it is evident that biLSTM works better than any other with the accuracy rate of 69%. The better prediction of biLSTM is an indicator that the fluctuations of stock indexes based on the news headlines is seemingly temporary but still there is correlation of distant events. Also, not only news but also other events do project an impact on the stock market.

The motivation for the work has been the successes of Deep Learning methods in Natural Language Processing (NLP). The future work will include the use of test methods such as reported in [13] and [14] for making better embedding vectors for the news headlines. Instead of using Deep Learning Algorithms, the use of Reinforcement Learning Algorithms can be another track to train the proposed model on market simulation.

Acknowledgment

We are grateful to PushShift for providing a free API that allows to pull data from Reddit. Thanks to the users of Reddit community who have posted authentic news from various sources in the WorldNews SubReddit.

Thanks to Annajiat Alim Rasel, Senior Lecturer, Department of Computer Science and Engineering and Deputy Director, Institutional Quality Assurance Cell, BRAC University for the guidelines.

Dan Jurafsky, Jackson Eli Reynolds Professor in Humanities, Professor of Linguistics, Professor of Computer Science, Stanford University and his co-author, James H. Martin, Professor of Computer Science and a fellow in the Institute of Cognitive Science, University of Colorado Boulder for their open source book, Speech and Language Processing, 3rd Ed. which help for NLP terminologies and technology.

References

[1] Yadav, R., Kumar, A. V., Kumar, A. (2019). News-based supervised sentiment analysis for prediction of futures buying behaviour. IIMB Management Review, 31(2), 157–166. https://doi.org/10.1016/j.iimb.2019.03.006

[2] Thanh, H. T. P., Meesad, P. (2014). Stock Market Trend Prediction Based on Text Mining of Corporate Web and Time Series Data. Journal of Advanced Computational Intelligence and Intelligent Informatics, 18(1), 22–31. https://doi.org/10.20965/jaciii.2014.p0022

[3] Seng, J.-L., Yang, H.-F. (2017). The association between stock price volatility and financial news – a sentiment analysis approach. Kybernetes, 46(8), 1341–1365. https://doi.org/10.1108/k-11-2016-0307

[4] Wen, M., Li, P., Zhang, L., Chen, Y. (2019). Stock Market Trend Prediction Using High-Order Information of Time Series. IEEE Access, 7, 28299–28308. https://doi.org/10.1109/access.2019.2901842

[5] Khedr, A. E., S.E.Salama, Yaseen, N. (2017). Predicting Stock Market Behavior using Data Mining Technique and News Sentiment Analysis. International Journal of Intelligent Systems and Applications, 9(7), 22–30. https://doi.org/10.5815/ijisa.2017.07.03

[6] Huang, B., Huan, Y., Xu, L. D., Zheng, L., Zou, Z. (2019). Automated trading systems statistical and machine learning methods and hardware implementation: a survey. Enterprise Information Systems, 13(1), 132-144.

[7] Hrinchuk, O., Khrulkov, V., Mirvakhabova, L., Orlova, E., Oseledets, I. (2019). Tensorized embedding layers for efficient model compression. In arXiv [cs.CL]. http://arxiv.org/abs/1901.10787

[8] X. Ding, Y. Zhang, T. Liu and J. Duan, "Using Structured Events to Predict Stock Price Movement: An Empirical Investigation", EMNLP, pp. 1415-1425, 2014

[9] Ralf C. Staudemeyer, Eric Rothstein Morris. (2019). Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks.

[10] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.

[11] Gilles Louppe. (2015). Understanding Random Forests: From Theory to Practice.

[12] Chen, T., Guestrin, C. (2016). XGBoost. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[13] Q. V. Le and T. Mikolov, "Distributed Representations of Sentences and Documents", ICML, vol. 14, pp. 1188-1196, 2014.

[14] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen and R. Ward, "Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval", IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), vol. 24, no. 4, pp. 694-707, 2016.

[15] Hushani, P. (2018). Using Autoregressive Modelling and Machine Learning for Stock Market Prediction and Trading. In Advances in Intelligent Systems and Computing (pp. 767–774). Springer Singapore.

[16] Porshnev, A., Redkin, I., Shevchenko, A. (2013, December). Machine learning in prediction of stock market indicators based on historical data and data from twitter sentiment analysis. In 2013 IEEE 13th International Conference on Data Mining Workshops (pp. 440-444). IEEE.

[17] Nti, I. K., Adekoya, A. F., Weyori, B. A. (2020). A systematic review of fundamental and technical analysis of stock market predictions. Artificial Intelligence Review, 53(4), 3007-3057.

[18] Menon, A., Singh, S., Parekh, H. (2019, March). A review of stock market prediction using neural networks. In 2019 IEEE International Conference on System, Computation, Automation and Networking (IC-SCAN) (pp. 1-6). IEEE.

[19] Nikou, M., Mansourfar, G., Bagherzadeh, J. (2019). Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms. Intelligent Systems in Accounting, Finance and Management, 26(4), 164-174.