Stock Trend Prediction using Deep Neural Networks in Time Series and Social Sentiment Analysis

Dr. P. Ravichandran[1], Dr.J.Dafni Rose[2], and K.Vijayakumar[3]
[1,2,3]St.Joseph's Institute of Technology, OMR, Chennai-119
ravichan3064@gmail.com[1]
jdafnirose@yahoo.co.in[2]
mkvijay@msn.com[3]

Abstract. In this paper, we propose an approach towards predicting the trend of stock price values by analyzing the relevant words occurring in social media like Twitter and by performing a time series analysis of the performance of the stock over the years. We obtain training data and train them separately against normalized values of stock prices themselves using neural networks and obtain the desired results by using the outputs of these separate approaches as the training data for another separate neural network that predicts the trend in the stock's future pricing along with the values with a certain degree of accuracy.

Keywords: Neural Networks, Word2Vec, LSTM

1 Introduction

Stock prices are changing dramatically everyday in market. There is a trade off between supply and demand. If the demand increases and the price will go automatically high. Alternatively if the people want to sale their goods than buying the demand would be more than the supply and the cost will be down.
Instead of collecting complete news articles predictions are made based on the tweets made by people on a social platform like Twitter to discern the people's opinion about a particular stock.

The reasoning behind this approach is that some of the greatest progress in deep learning has been in the area of text processing. This capability opens the door for the possibility of leveraging the enormous corpus of unstructured, qualitative textual data related to companies that can be found in SEC filings, news reports, blog posts, social media, and earnings transcripts.

The idea behind this paper came from a suggestion in an online article that Berkshire Hathaway's prices changed whenever tweets related to actress Anne Hathaway were made [11]. From this fact it was discerned that automated trading algorithms made trading decisions based on the topics that were trending on Twitter. In this paper we attempt to identify such potential fluctuations caused by the trading algorithms that use twitter to make trading decisions

* No academic titles or descriptions of academic positions should be included in the addresses. The affiliations should consist of the author's institution, town/city, and country.

ICASISET 2020, May 16-17, Chennai, India

DOI 10.4108/eai.16-5-2020.2304210

We try to extrapolate stock prices of large scale publicly traded companies like Alphabet(GOOG/GOOGL) which are frequently talked about in social media. This idea is discussed in six sections. Following the introduction in Sect.1,the research material used for studying keyword based prediction of stock values are discussed in Sect.2. The process of obtaining and cleaning social network data and stock quotes is discussed in Sect.3. In Sect.4 we explain the proposed approach for predicting stock values.

2 Related Work

The approach taken by existing methods tries to make predictions about the direction of movement of the stock that is if the value of a stock is going to increase or decrease. The existing methods of predicting stocks use a sentiment analysis of the text in the traditional sense by trying to figure out if a given body of text conveys a positive or negative tone towards a subject[3]. There has also been an instance where recurrent neural networks such as LSTM networks have been used to model the trend followed by the price values taken by the stocks which accounts for patterns of rallies and corrections.

[1] tries to improve stock prediction using an implementation of neural networks with a back propagation algorithm for the stock market. In[2] this study, the authors were able to get a prediction with accuracy of 64% which is taken as the original method to benchmark against in our studies.

3 Data Pre-Processing

3.1 Cleaning Twitter Data

The Tweets are extracted from Twitter using their APIs' to search for specific keywords related to the company that was studied in the sample. The keywords were chosen empirically and the tweets containing those keywords were extracted and stored.

1) Tokenization: Tweets can be divided into individual words based on the space and irrelevant symbols like emoticons are removed. List of individual words for each tweet can be formed by us.

2) Stopword Removal: Words which don't have emotion are named as Stop words. After splitting a tweet, words like a, is, the, with etc. are removed from the list of words.

3) Regex Matching for special character Removal: Python is used to for Regex matching to match URLs and could be replaced by the term URL. Often tweets consists of hash tags(#)and @ addressing other users. They are also replaced suitably. For example, #Microsoft is replaced with Microsoft and @Billgates is replaced with USER. Prolonged word showing intense emotions like cooooooooool! is replaced with cool! After these stages the tweets are ready for sentiment classificati However, these tweets collected over a period of 3weeks are considered dirty as they contained emojis and they were removed during the cleaning processes that included stemming and lemmatization of text[4]. This was followed byt he creation of a Word2Vec model generated by the corpusof text that was collected using the real-time tweet approachintermittently over a period of 5 months and then applied the same cleaning process as mentioned above. This gave us the Word2Vec values that were used the deep learning model for prediction of changes in values based on not emotional sentiment but the market sentiment that correlates specific keywords with increases or decreases in the values of the stock as was observed in the case of Berkshire Hatha way[5]. Social data for Sentiment analysis is fetched using the Twitter API. It has a python library known as Twython[11].

The cleaned tweets are preprocessed and converted into numeric representation for training the dense network.

3.2 Collection of Stock Quotes

The financial data is collected from Yahoo Finance, which is a data source for stock market prices.

4 Proposed System

In our proposed approach as shown in Fig.1, one Long Short Term Memory(LSTM) Network[7] , one Dense network [6] and one Shallow Network [6] is used. The LSTM [7] is used to perform multivariate time series analysis on historical stock data obtained from Yahoo! Finance. The dense network is used to perform sentiment analysis on social data collected from Twitter.

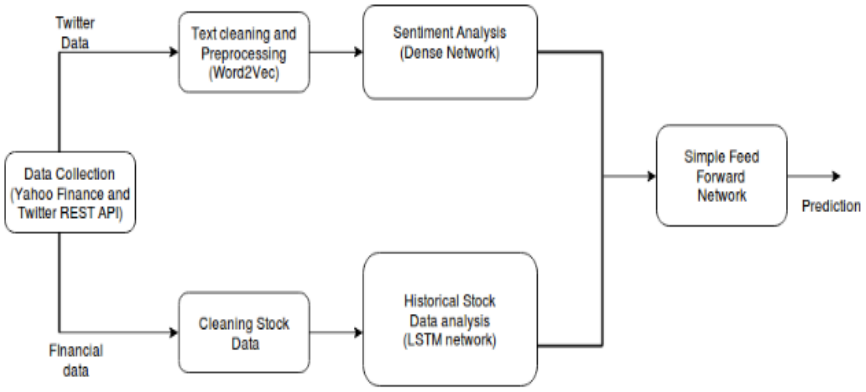

Fig.1: Architecture of the Proposed System

4.1 The LSTM Network Trained on Time Series

The LSTM [7] is used to perform multivariate time series analysis on Historical stock data obtained from Yahoo! Finance.

Processing the Window

The stock data is preprocessed by converting it into windows of a specified length. These windows are given as an input to the LSTM network [7]. The target value to be predicted is the stock price values for the next day after the end of the window.

```
result = []
for index in range(len(df) - sequence_length):
  result.append(data[index: index + sequence_length])
```

Normalizing Stock Data

The historical stock data does not have a nice pattern that could be easily approximated. Using these raw values in a neural network would break the optimization process itself as the network will not converge to any sort of optimum values. Hence we need to normalise these kind of real world data before feeding it into a network. We will take each n-sized window of training/testing data and normalize each one to reflect percentage changes from the start of that window, so the data at point i = 0 will always be 0.

$$ni = \left(\frac{p_i}{p_o}\right) - 1$$

Denormalizing Stock Data:

After the network is trained, we use it to predict a value by giving a window as input. This predicted value is in the normalized format, which we need to denormalize to get the actual output. We can get the denormalised output by using the given formula

$$p_i = p_0(n_i + 1)$$

4.2 The Multilayer Dense Network Trained on Twitter Corpus

This step would be using the word2vec model for the deep learning implementation using a dense network[6]. The algorithm used in our implementation is elucidated below:

Initialization:

embeding_model = doc2vec(tagged_sentences)

embedded_model.wv["crashing"] = [-0.327,0.325,...]

Reduction of Dimensions:

```
for i,document in enumerate(documents):
  for j,word in enumerate(document):
    if j == len(enumerate(document)):
      break
    else:
      if word in d2v_model:
        x[i,j] = get_unique_representation(word,d2v_model)
      else:
        x[i,j] = assign_zero()
```

The variable d2v model represents the Word2Vec Model [5] that was generatedfrom a collected twitter corpus.

Neural Network Design:

X is the training data vector of numerical representation of tweets

W is the weight matrix

Y be the output matrix

Neuron Function:

The neuron equation is defined as :

$$F(x) = (< w, x >) = \sum_{i=0}^{k} w_i x_i$$

The loss function is given by

$$E(x,y) = \frac{1}{n}\sum_{i=0}^{n} y_i \, lna(x_i) + (1 - y_i) ln(1 - a(x_i))$$

where,

x = x_1, x_2,…x_ni.e input set

y = y_1, y_2,y_ni.e output set

The losses are adjusted and the weights are updated using the following formula

$$\boldsymbol{W_{current} = w_{current} - \eta \nabla E(w_{current})}$$

where,
$w_{current}$is the current weight matrix
η is the learning rate
E is the error function

The numerical representation of the tweets are fed into the neural network[6] thus constructed to obtain stock price predictions independent of the previous LSTM method[7].

4.3 The Final Shallow Network Trained on Results from LSTM and Dense Networks

The exact architectures of the LSTM[7] and Dense Neural Networks[8] are designed by empirical methods to get the desired efficiency. The results from the time series model(LSTM)and the twitter sentiment model(Dense Network) [9] are fed into a shallow two layer network which is largely similar to the dense network described in Sect. 4.2 but has only two layers in its neural network which is used to find the weighted averages between the two outcomes. This gives us the final result for the prediction.

5 Result Analysis

The results obtained from proposed prediction models are as follows:

1. The opening values predicted as in Table 1 have an error margin of $\pm$0.008565%
2. The high values predicted as in Table 2 have an error margin of $\pm$0.005578%
3. The low values predicted as in Table 3 have an error margin of $\pm$0.009845%
4. The closing values predicted as int Table 4 have an error margin of $\pm$0.007049%

We get an average error margin of $\pm$0.007759% in all these results shown below in the Tables[1-4] and the graphs in Fig.[2(a)-2(d)] results in which the graphs above in each of the charts represent the real market prices while the graphs below in each of the charts the predicted prices generated by our model.

Table 1: Actual Vs. Predicted Open

Actual Open	Predicted Open
836.00	820.57
843.28	831.13
844.00	838.19
844.00	842.14
844.00	839.59
843.64	843.68
847.59	844.16
849.03	842.94
851.61	846.14
850.01	847.57
850.01	845.56
850.01	843.01
851.40	844.65
831.91	834.74
821.00	830.92
820.08	824.46
806.95	821.60
806.95	823.33
806.95	824.25
820.41	827.34
825.00	828.81
833.50	828.90

Table 2: Actual Vs. Predicted High

Actual High	Predicted High
842.00	829.32
844.91	837.84
848.68	844.51
848.68	847.75
848.68	845.66
847.24	849.27
848.63	849.37
850.85	848.73
853.40	851.77
850.22	853.30
850.22	851.65
850.22	849.56
853.50	850.41
835.55	841.08
822.57	836.96
821.93	830.34
821.63	827.56
821.63	828.97
821.63	829.73
825.99	832.36
832.77	834.27
833.68	835.76

Table 3: Actual Vs. Predicted Low

Actual High	Predicted High
834.21	819.02
839.50	827.89
843.25	834.55
843.25	837.91
843.25	835.75
840.80	839.42
840.77	839.60
846.13	838.84
847.11	841.87
845.15	843.35
845.15	841.64
845.15	839.48
829.02	840.50
827.18	831.17
812.26	827.18
808.89	820.70
803.37	817.94
803.37	819.41
803.37	820.19
814.03	822.89
822.38	824.66
829.00	825.81

Table 4: Actual Vs. Predicted Close

Actual High	Predicted High
838.68	820.03
843.25	831.12
845.54	838.65
845.54	842.82
845.54	840.13
845.62	844.46
847.20	844.93
848.78	843.69
852.12	847.11
848.40	848.65
848.40	846.53
848.40	843.85
830.46	845.53
829.59	834.96
817.58	830.83
814.43	823.88
819.51	820.82
819.51	822.64
819.51	823.61
820.92	826.87
831.41	828.49
831.50	828.75

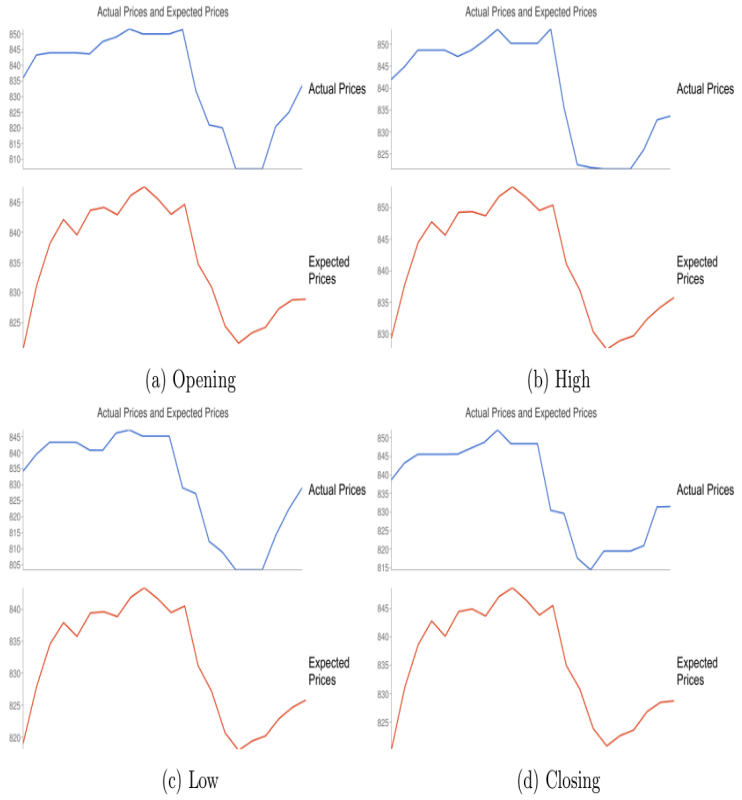

Fig. 2: Predicted Stock Values vs. Actual Values Seen on the Market

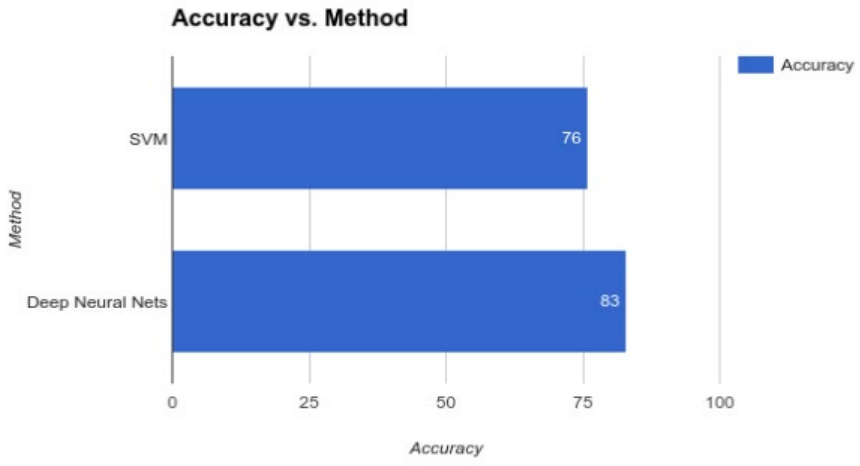

Fig 3.Comparison of Deep Neural Netsvs SVM

6 Conclusion

The graph in Fig.3 gives us an overview of the performance of our proposed deep learning method vs the original method of prediction taken for consideration [10]. When it comes to predicting the direction a stock takes as observed from the above results vs the results from the previously most effective method. Hence we find that the proposed method of using both sentiment data from twitter obtained by training the processed tweets along with the time series data against the stock values

help us gain an advantage over the existing methods of predicting the stock market as we get an approximately average error margin of only ±0.007759% in all the 4 parameters of the stock prices. We could not only find the trend about to be followed by the stock prices in the near future but can also be helpful in reasonably predicting the approximate price of the stock on that day based on public sentiment from twitter and time series analysis of the price trend. This may help in making useful investments decisions to enrich one's investment portfolio.

References

[1] BhagwatChauhan ,UmeshBidave, AjitGangathade, Sachin Kale. Stock Market Prediction Using Artificial Neural Networks International Journal of Computer Science and Information Technologies, Vol. 5 (1) , 2014, 904-907

[2] PierpaoloDondio. Stock Market Prediction without Sentiment Analysis: Using a Web-traffic based Classifier and User-level Analysis. DOI: 10.1109/HICSS.2013.498

[3] HadiPouransari, HamidrezaChalabi.Event-based Stock Market Prediction. http://cs229.stanford.edu/proj2014/Hadi Pouransari,HamidChalabi,Event-based stock market prediction.pdf

[4] Xiao Ding, Yue Zhang, Ting Liu, JunwenDuan. Deep Learning for Event-Driven Stock Prediction. IJCAI 2015 Proceedings of the 24th International Conference on Artificial Intelligence Pages 2327-2333,Buenos Aires, Argentina — July 25 - 31, 2015, AAAI Press.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean.Efficient Estimation of Word Representations in Vector Space.arXiv:1301.3781v3 [cs.CL]

[6] YannLecun, Leon Bottou, YoshuaBengio, and Patrick Haffner. Gradient-based Learning Applied to Document Recognition. DOI: 10.1109/5.726791

[7] SeppHochreiter, JurgenSchmidhuber.Long Short-Term Memory. DOI:10.1162/neco.1997.9.8.1735

[8] ChristopherOlah. Understanding LSTM Networks. http://colah.github.io

[9] Andrej Karpathy. Unreasonable Effectiveness of Recurrent Neural Networks. http://karpathy.github.io/

[10]The Atlantic (March 18,2011) Does Anne Hathaway News Drive Berkshire Hathaway's Stock?. https://www.theatlantic.com/technology/archive/2011/03/does-anne-hathaway-news-drive-berkshire-hathaways-stock/72661/

[11] Stock market analysis using candlestick regression and market trend prediction (CKRM) M. Ananthi & K. Vijayakumar, Journal of Ambient intelligence and humanized computing,April 2020.