EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Stock Price Prediction using Artificial Neural Model: An Application of Big Data

Malav Shastri[1], Sudipta Roy[2] and Mamta Mittal[3, *]

[1] Department of Computer Science and Engineering, Ganpat University, Mehsana-Gozaria Highway, Mehsana-384012, Gujarat, India
[2] Washington University in Saint Louis, MIR department, 510 South Kings highway Blvd., MO 63110, USA
[3] Department of Computer Science & Engineering, G.B. Pant Govt. Engineering College, Okhla, New Delhi, India

Abstract

In recent time, stock price prediction is an area of profound interest in the realm of fiscal market. To predict the stock prices, authors have proposed a technique by first calculating the sentiment scores through Naïve Bayes classifier and after that neural network is applied on both sentiment scores and historical stock dataset. They have also addressed the issue of data cleaning using a Hive ecosystem. This ecosystem is being used for pre-processing part and a neural network model with inputs from sentiment analysis and historic data is used to predict the prices. It has been observed from the experiments that the accuracy level reaches above 90% in maximum cases, as well as it also provides the solid base that model will be more accurate if it trained with recent data. The intended combination of sentiment analysis and Neural networks is used to establish a statistical relationship between historic numerical data records of a particular stock and other sentimental factors which can affects the stock prices.

Keywords: News Headlines, Stock Market, Big Data, Artificial Intelligence, Artificial Neural Networks, Sentimental Analysis.

Received on 11 November 2018, accepted on 17 December 2018, published on 03 January 2019

doi: 10.4108/eai.19-12-2018.156085

*Corresponding author. Email: mittalmamta79@gmail.com

1. Introduction

The importance of data has been immensely increased in last decade. The pace at which earth is producing data is meteoric, no wonder the rapid growth of data has given a birth to many problems like storage, analysis and processing of the data. These problems have termed as Big Data. As time has passed people have not only addressed the issue of its storage but also, they have started using that data for analyzing the trends and patterns exists in them and to use those patterns to predict the future trends. Every day there are millions of people who buy and sell stocks of various companies. Thus, Terabytes or even Petabytes of data are being generated from different exchanges. Financial organizations and retail traders can extract a great amount of information which can help them in their trading decisions. Financial market is largely based on the daily trading of stock so by the use of machine learning techniques one can create prediction models which can predict the stock prices in advance.

From last few years there have been many up's and downs in stock market, as there are n factors which can affects a share market. Thus, due to its dynamic nature, it is very highly difficult to predict a stock price. To address this issue there should be some system which can detect the pattern in stock prices when influenced by political, economic and natural environment as well as which can take what are the people's sentiment about the particular company. In this paper, authors have presented a possible mixture of sentiment analysis and historical stock price. Now one of the major concerns before predicting the stock prices is to use the reliable historic stock data, thus pre-processing of data is also must.

In paper [1] the authors have proposed a system in which the stock market data is extracted by applying keywords on twitter data and store it into Hadoop Distributed File System (HDFS) using flume or Hadoop

commands. After that data is pre-processed by removal of slang words and other unnecessary elements of the particular tweet. In another paper [2], the authors have used Bombay Stock Exchange (BSE) data and used Hadoop MapReduce techniques to preprocess the datasets of BSE stock exchange and predicted the stock values of stocks.

In paper [3], the authors have presented a system that can predicts stock market movement, based on historical stock prices and market sentiment analysis. They used data of Standard and Poor's 500 (S&P 500) from January 2008 to April 2010 from Yahoo! Finance. After that they used Naïve Bayes classification for sentiment analysis, and stock movement were predicted using Support Vector Machines (SVM), Logistic and Neural network techniques. In paper [4] prediction of stock prices of three Indian National Stock Exchange (NSE) listed companies has been done using SVM technique.

In paper [5], authors have considered two strategies, series and parallel in financial time series forecasting and has scrutinized the performance of the same. In this, authors have concluded that using Auto Regressive Integrated Moving Average (ARIMA) along with multilevel perceptron produces much more accurate results. In paper [6] authors have predicted the stock value using sentiment analysis. Authors have considered news headlines for the sentiment analysis purpose. They have used three approaches: KNN, SVM and Naïve Bayes classification. In paper [7] authors have used clustering and multiple regression for forecasting the stock price. In paper [8] authors have developed a Natural Language Processing (NLP) model for stock forecasting. This model basically uses online news to forecast future stock values. In paper [9] to overcome the time series forecasting, authors have used an outlier data mining technique for stock forecasting. They concluded that their approach is better for predicting long term behavior of stock trend. In paper [10] by using decision tree classifier specifically ID3 and C4.5 authors have suggested better times for buying and selling the stock prices. In paper [11] authors compared the performance of four machine learning algorithms which are SVM, Random Forest (RF), Naïve Bayes (NB), and Artificial Neural Network (ANN), in predicting the future value for Reliance and Infosys datasets. In paper [12] authors proposed a polynomial neural network for the task of stock market forecasting. They have also used the concept of partial descriptions and used them with the original features. Researchers have applied machine learning and deep learning technologies for prediction in number of applications like in health sector, crime sector and images analysis [13-17].

Plethora of research has already been done to predict the stock prices or market trends, by considering either the numerical historical stock prices or the textual sentiments data and maximum researcher did not consider them together. Moreover, while doing the sentiment analysis data is taken from twitter, which are less reliable compared to the news headlines. So, in this paper, authors have considered both historical stock prices as well as sentiment analysis from the news which is novel in itself. Moreover, Big Data technologies have been presented to handle the large data by using Big Data ecosystem which is a solution for cleaning the data before using it for prediction purpose.

2. Proposed Methodology

The proposed methodology has included sentiment analysis of news dataset as well as historic stock prices. The reason behind considering a news dataset for sentiment analysis is that unlike other resources news headlines are majorly made upon the statistical facts and different events. Big data doesn't always mean HDFS or map reduce. Hive ecosystem have been used to clean dataset as it can directly interact with HDFS (Hadoop Distributed File System). The architecture of proposed methodology is shown below in Figure 1. In this, the stock market data is cleaned by HIVE and passed to ANN whereas on news dataset sentiment analysis has performed and generated sentiment score is passed to Multilevel Perception Network.

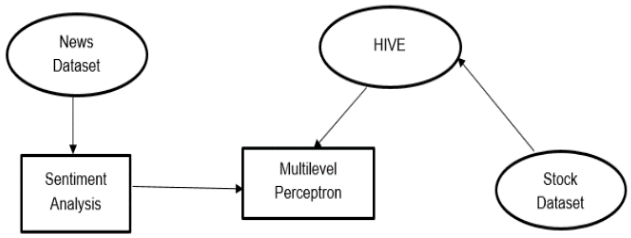


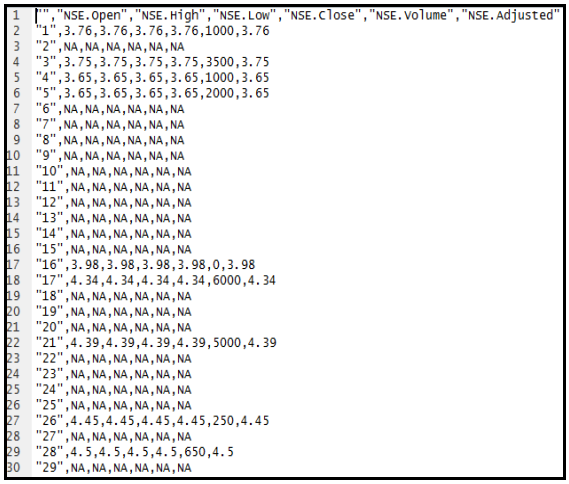
Figure 1. Architecture of the proposed methodology

2.1. Data Cleaning With HIVE

As mostly data available from twitter and stock market is in unorganized form, so to get insight into it, one should opt data cleaning. Hive is better solution for cleaning of data because of three things, apart from the fact that HIVE is built upon Hadoop MapReduce, which is a framework for distributed and parallel processing of large data, its architecture and query language make it a unique [18-19].

Using Hive Query Language (HQL) users can perform multi query on same input data. Most interesting part of HIVE is that compiler of HQL translates the statement into a directed acyclic graph of map reduce jobs. So the query is divides into smaller map reduce jobs. The data with too many null values may drive some

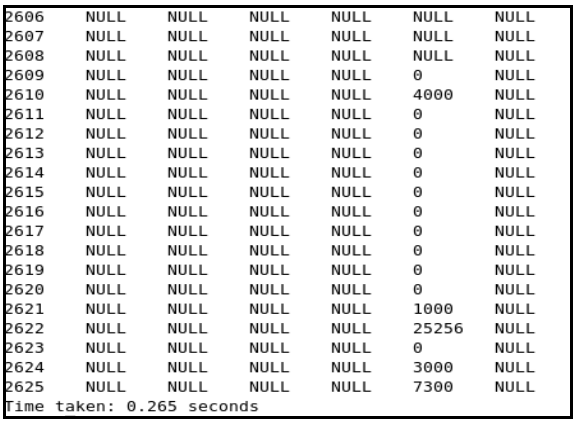
undesirable results, so it is really important to clean the data. Figure 2 is the snapshot of NSE data which containing many null values.



```
1  "","NSE.Open","NSE.High","NSE.Low","NSE.Close","NSE.Volume","NSE.Adjusted"
2  "1",3.76,3.76,3.76,3.76,1000,3.76
3  "2",NA,NA,NA,NA,NA,NA
4  "3",3.75,3.75,3.75,3.75,3500,3.75
5  "4",3.65,3.65,3.65,3.65,1000,3.65
6  "5",3.65,3.65,3.65,3.65,2000,3.65
7  "6",NA,NA,NA,NA,NA,NA
8  "7",NA,NA,NA,NA,NA,NA
9  "8",NA,NA,NA,NA,NA,NA
10 "9",NA,NA,NA,NA,NA,NA
11 "10",NA,NA,NA,NA,NA,NA
12 "11",NA,NA,NA,NA,NA,NA
13 "12",NA,NA,NA,NA,NA,NA
14 "13",NA,NA,NA,NA,NA,NA
15 "14",NA,NA,NA,NA,NA,NA
16 "15",NA,NA,NA,NA,NA,NA
17 "16",3.98,3.98,3.98,3.98,0,3.98
18 "17",4.34,4.34,4.34,4.34,6000,4.34
19 "18",NA,NA,NA,NA,NA,NA
20 "19",NA,NA,NA,NA,NA,NA
21 "20",NA,NA,NA,NA,NA,NA
22 "21",4.39,4.39,4.39,4.39,5000,4.39
23 "22",NA,NA,NA,NA,NA,NA
24 "23",NA,NA,NA,NA,NA,NA
25 "24",NA,NA,NA,NA,NA,NA
26 "25",NA,NA,NA,NA,NA,NA
27 "26",4.45,4.45,4.45,4.45,250,4.45
28 "27",NA,NA,NA,NA,NA,NA
29 "28",4.5,4.5,4.5,4.5,650,4.5
30 "29",NA,NA,NA,NA,NA,NA
```

Figure 2. Data retrieved from the data source containing many null values

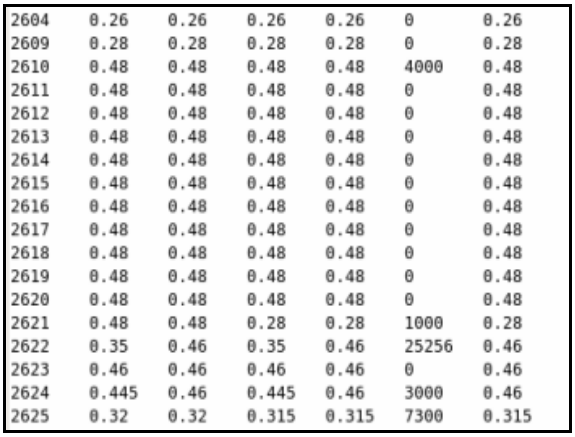
Now, schema of table is to be created in Hive according to data retrieved from data sources. Major columns are date, opening price, closing price, volume, adjusted closing price. Data is collected in csv format which makes it easier to store in Hadoop as well as to load in Hive table. Figure 3 shows the data after loading it in a Hive table and this data contains many null values in it. So, HQL is being used here so that it doesn't consider the records which has null values.



```
2606    NULL    NULL    NULL    NULL    NULL    NULL
2607    NULL    NULL    NULL    NULL    NULL    NULL
2608    NULL    NULL    NULL    NULL    NULL    NULL
2609    NULL    NULL    NULL    NULL    0       NULL
2610    NULL    NULL    NULL    NULL    4000    NULL
2611    NULL    NULL    NULL    NULL    0       NULL
2612    NULL    NULL    NULL    NULL    0       NULL
2613    NULL    NULL    NULL    NULL    0       NULL
2614    NULL    NULL    NULL    NULL    0       NULL
2615    NULL    NULL    NULL    NULL    0       NULL
2616    NULL    NULL    NULL    NULL    0       NULL
2617    NULL    NULL    NULL    NULL    0       NULL
2618    NULL    NULL    NULL    NULL    0       NULL
2619    NULL    NULL    NULL    NULL    0       NULL
2620    NULL    NULL    NULL    NULL    0       NULL
2621    NULL    NULL    NULL    NULL    1000    NULL
2622    NULL    NULL    NULL    NULL    25256   NULL
2623    NULL    NULL    NULL    NULL    0       NULL
2624    NULL    NULL    NULL    NULL    3000    NULL
2625    NULL    NULL    NULL    NULL    7300    NULL
Time taken: 0.265 seconds
```

Figure 3. Data after loading operation in HIVE table, showing NULL values

Figure 4, presents the snapshot of final table which is having cleaned data, without null values.



```
2604    0.26    0.26    0.26    0.26    0       0.26
2609    0.28    0.28    0.28    0.28    0       0.28
2610    0.48    0.48    0.48    0.48    4000    0.48
2611    0.48    0.48    0.48    0.48    0       0.48
2612    0.48    0.48    0.48    0.48    0       0.48
2613    0.48    0.48    0.48    0.48    0       0.48
2614    0.48    0.48    0.48    0.48    0       0.48
2615    0.48    0.48    0.48    0.48    0       0.48
2616    0.48    0.48    0.48    0.48    0       0.48
2617    0.48    0.48    0.48    0.48    0       0.48
2618    0.48    0.48    0.48    0.48    0       0.48
2619    0.48    0.48    0.48    0.48    0       0.48
2620    0.48    0.48    0.48    0.48    0       0.48
2621    0.48    0.48    0.28    0.28    1000    0.28
2622    0.35    0.46    0.35    0.46    25256   0.46
2623    0.46    0.46    0.46    0.46    0       0.46
2624    0.445   0.46    0.445   0.46    3000    0.46
2625    0.32    0.32    0.315   0.315   7300    0.315
```

Figure 4. Data in the output table without null values

2.2. Predicting adjusted closing prices with sentiment analysis and Artificial Neural Network

Multilevel Perception ANN has been used to predict the future trends. In this news headlines, are classified using NB classifier in two classes positive and negative. Thus sentiment score has been created from that classification. Now this sentiment score is used along with other five attributes, which are date, opening price, highest value on that day, lowest value on that day and volume of shares traded on that day as an input to ANN. Date is important attributes as it helps in establishing statistical relationship between dates with other attributes, so that it enables us to extract patterns between dates and closing prices, and stock prices are subject to time series data, there are noticeable effects on stock prices as and when time passes. Moreover, date also help in knowing the stock prices patterns before the weekend and after the weekend same with the public holidays which may affect particular stocks prices. Author's goal is to forecast closing prices of a stock on a particular day by giving previous day's data as an input.

In this paper authors have demonstrated the complete method for Apple stock, the reason behind taking up the Apple stock is that it is more consumer faced company, which has many end users worldwide, so they assume that the news which are being daily created for Apple should have some decent amount of portion of news which are directly related to problems that it's consumers are facing as well as their sentiments about the products of the company. So, historical stock prices as well as news are taken for the period 2013 to 2016 from www.nasdaq.com. and further this dataset has divided into two parts, 3/4th portion of the data is used as training data and 1/4th portion as test data. One more case is considered in which data of year 2016 has been taken into consideration and it is also further divided into training and testing datasets.

Sentiment Analysis

Sentiment Analysis plays very important role in s stock market as its prices are majorly dependent on external factors like political factors or geographical factors. Thus, the given news is divided into two classes one is positive (POS) another is negative (NEG) using Naive Bayes classifier. These classes are assigned a score 1 to positive and 0 to negative. This score is then used with other data attributes like opening price, high, low, volume etc. The model is represented in Figure 5.

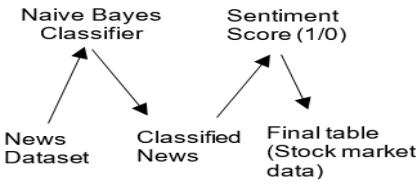


Figure 5. Flow of Data from sentiment analysis to final stock data table

Textual Data Classification using Naïve Bayes Classifier

It operates on word to word and find each unique word probability [20]. In this apple's stock prices is considered and news headlines regarding apple stock is collected, small chunk of it is presented in Table 1. Basically, it is a training data where authors have manually classified news in POS and NEG classes.

Table 1. Training data example for sentiment analysis

Headline	Text	Class
1	People loved the iPhone X	POS
2	People hated the iPhone X	NEG
3	A great iPhone X, good iPhone X	POS
4	Poor design	NEG
5	Great Design a good iPhone X	POS

In first step, unique words are identified like People, loved, the, iphone10, hated, a, great, good, poor, design. Next step is to convert these words into the feature matrix, which represents how many times unique word is occurring. Table 2 represents the feature matrix for POS and NEG classes.

Further, subsets of feature matrix, one with feature matrix of positively classified headlines, and another with feature matrix of negatively classified headline are represented in Table 3 and 4 respectively.

Table 2. Feature Matrix

Headline	People	Loved	The	iPhoneX	Hated	A	Great	Poor	Design	Good	Class
1	1	1	1	1							POS
2	1		1	1	1						NEG
3				2		1	1			1	POS
4								1	1		NEG
5				1		1	1		1		POS

Table 3. Feature matrix of only positive classes

Headline	People	Loved	The	iPhoneX	Hated	A	Great	Poor	Design	Good	Class
1	1	1	1	1							POS
2											
3				2		1	1			1	POS
4											
5				1		1	1		1		POS

Table 4. Feature matrix of only negative classes

Headline	People	Loved	The	iPhoneX	Hated	A	Great	Poor	Design	Good	Class
1											
2	1		1	1	1						NEG
3											
4								1	1		NEG
5											

Next step is to find the probability of each words which falls either in POS or NEG class. Authors have provided textual data classification using Naïve Bayes classifier in their research work [20]. Here 'wk' represents any word. The conditional probability for a given word 'wk' is what the probability that the headline is positive or what the probability that the headline is negative. Probability of word given that it is in positive class is given by equation 1:

$$p(wk|POS) = nk + 1/n + |vocabulary| \quad (1)$$

Similarly, Probability of word given that it is in negative class is given by equation 2:

$$p(wk|NEG) = nk + 1/n + |vocabulary| \quad (2)$$

Where 'nk' is the number of times word k occurs in these cases. Table 5 and 6 represents the probability calculation of each word of POS class and NEG class respectively.

Table 5. Probability calculation for positive class

p(People|POS) = 1+1/14+10 = 0.0833	p(Loved|POS) = 1+1/14+10 = 0.0833
p(The|POS) = 1+1/14+10 = 0.0833	p(iPhoneX|POS) = 1+1/14+10 = 0.0833
p(A|POS) = 2+1/14+10 = 0.125	p(Great|POS) = 2+1/14+10 = 0.125
p(Design|POS) = 1+1/14+10 = 0.0833	p(Good|POS) = 2+1/14+10 = 0.125
p(Hated|POS) = 0+1/14+10 = 0.0417	p(People|POS) = 0+1/14+10 = 0.0417

Table 6. Probability calculation for negative class

p(People|NEG) = 1+1/6+10 = 0.125	p(Loved|NEG) = 0+1/6+10 = 0.0625
p(The|NEG) = 1+1/6+10 = 0.125	p(iPhoneX|NEG) = 1+1/6+10 = 0.125
p(A|NEG) = 0+1/6+10 = 0.0625	p(Great|NEG) = 0+1/6+10 = 0.0625
p(Design|NEG) = 1+1/6+10 = 0.125	p(Good|NEG) = 0+1/6+10 = 0.0625
p(Hated|NEG) = 1+1/6+10 = 0.125	p(People|NEG) = 1+1/6+10 = 0.125

By this way, model has trained. Now, a random news is considered and on the basis equation 3 values of POS and NEG has calculated.

$$V = argmax\ P(v_j) \prod_{v_j \in V} P(W|V_j) \quad (3)$$

In equation 3. V stands for "value" or "class" and W is representing Words. Similarly, v_j the probability of a particular class, $P(W|V_j)$ is the class conditional probability.

Let's say new Headline is: "People hated the new design" so for positive class v_j = POS:
p(POS)p(People|POS)p(hated|POS)p(the|POS)p(new|POS)p(design|POS) = 6.03×10^{-7}

and for negative class v_j = NEG:
p(NEG)p(People|NEG)p(hated|NEG)p(the|NEG)p(new|NEG)p(design|NEG) = 1.22×10^{-5}

Now from these two answers it is clearly seen that NEG has higher probability than POS, so headline will fall under NEG class. Further, table 7 represents the sample of table created by classifying headlines into two classes, where first column is representing the date which is converted in integer format and other column is the class to which it belongs.

Table 7. Date wise classification of news headlines in two classes

Date	Class
20080722	Pos
20080723	Pos
20080724	Pos
20080725	Neg

It also contains the accuracy at the end which is 0.90378 and represented by Figure 6. This figure, consists of most informative features result their positive negative ratio and the accuracy of the complete classification process which is 90.378%.

```
Most Informative Features
       contains(gadgets) = True              pos : neg    =      6.4 : 1.0
    contains(conference) = True              pos : neg    =      5.7 : 1.0
       contains(twitter) = True              pos : neg    =      5.7 : 1.0
       contains(across) = True               pos : neg    =      5.7 : 1.0
   contains(subscribers) = True              pos : neg    =      5.0 : 1.0
       contains(buoyed) = True               pos : neg    =      5.0 : 1.0
       contains(claims) = True               pos : neg    =      5.0 : 1.0
    contains(published) = True               neg : pos    =      5.0 : 1.0
       contains(banks) = True                neg : pos    =      5.0 : 1.0
    contains(ventures) = True                neg : pos    =      5.0 : 1.0
0.9037800687285223
```

Figure 6. Snapshot of results of most informative features and accuracy (At the end)

In stock data, authors have added one more column sentiment score after adjusted closing price. This final table has been given in table 8.

Multilevel Perceptron: Multilevel perceptron uses different loss functions for classification and regression. Authors trained it using back propagation and identity function is used as activation function in the output layer.

The output is a set of continuous values, and like every prediction based application of neural network it is also having one output node. One more important thing is error signal which is a difference between the desired output and the actual output, the weights get updated in training period, so that generated error after some iteration should be minimum. The maximum number of iterations can be fixed, so that the training period will stop after that much iteration. In this model: Date, opening price, high of the day, low of the day, volume and sentiment score are the input variables along with a bias value. The hidden layer and an output layer which gives a single value as an output makes this system a multilevel perceptron regression. So internally the network regresses different independent input variable onto the single dependent variable by using equation 4 to compute the loss function [21]:

$$Loss(\hat{y}, y, W) = \frac{1}{2}\left\|\hat{y} - y\right\|_2^2 + \alpha/2\|W\|_2^2 \quad (4)$$

It is also clear by the formula that it's a squared error function, and $\alpha>0$ is a non-negative hyper parameter that controls the magnitude of the penalty. Now coming to optimization according to gradient decent which is an algorithm for finding a minimum of the function or let's say optimization algorithm the gradient ∇LossW of the loss with respect to the weights is computed by equation 5:

$$W^{i+1} = W^i - \varepsilon \nabla Loss^i w \quad (5)$$

Where, i is the iteration step, and ε is the learning rate with a value larger than 0. At last the learning process stops when it reaches the maximum number of iteration or when the error loss is below the predefined threshold value. After creating an instance of the perceptron model with such kind of perceptron model is being trained. In input, two parameters one the input data and other is the targeted value has been provided where, input data is the training dataset and target values are the closing prices of the next day. So it is fitting a model for training data of a particular day with the target value of next day's adjusted closing price.

Table 8. Final table to be used as input in prediction algorithm

Date	Open	High	Low	Close	Volume	Adjusted Closing	Sentiment
22-07-2008	149	162.76	146.5	162.02	4.7E+08	20.991	1
23-07-2008	164.99	168.37	161.5	166.26	2.65E+08	21.540	1
24-07-2008	164.32	165.26	158.4	159.03	2.1E+08	20.603	1
25-07-2008	160.4	163	158.6	162.12	1.58E+08	21.004	0

3. Results

In this study, the stock prices data from 2013 to 2016 has been considered and this dataset is divided into two parts: 3/4th portion of the dataset as training data and other 1/4th part is considered as test data. In the first case, test dataset is of the period from 04-01-2016 to 30-12-2016 and two hidden layers are taken. 1st hidden layer consists of 7 neurons and second layer has 9 neurons. Figure 7 is the scatter plot of predicted stock prices generated by the Perceptron network fitted on the actual values of the test data from 04-01-2016 to 30-12-2016.

In this figure, black data points are actual data points where red are the predicted values. Table 9 represents the last 15 predicted values and actual values of the period 04-01-2016 to 30-12-2016 out of total 252 predicted values.

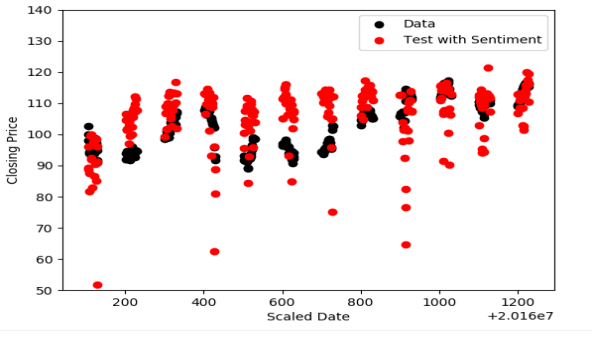


Figure 7. Values AAPL stock generated by the Perceptron network fitted on the actual values from 04-01-2016 to 30-12-2016

Table 9. Last 15 Actual and predicted values of the period 04-01-2016 to 30-12-2016

Date	Actual	Predicted
09-12-2016	113.4580	112.5176
12-12-2016	112.8109	108.3283
13-12-2016	114.6927	112.9139
14-12-2016	114.6927	102.9985
15-12-2016	115.3200	108.5401
16-12-2016	115.4693	101.4045
19-12-2016	116.1364	102.6459
20-12-2016	116.4451	112.1114
21-12-2016	116.5546	115.7410
22-12-2016	115.7879	114.3940
23-12-2016	116.0169	113.0787
27-12-2016	116.7538	119.8395
28-12-2016	116.2559	117.5277
29-12-2016	116.2260	116.0375
30-12-2016	115.3200	119.3883

Error in this model is measured as Mean Absolute Percentage Error (MAPE), which is the measurement of accuracy of prediction model, this error is 8.2148 in this case, thus this model is 91.8% accurate.

Authors have also considered one another case in which model is trained on one-year dataset which is of 2016. For this one hidden layer which contains 9 neurons is taken. Figure 8 is the scatter plot of predicted stock prices generated by the Perceptron network fitted on the actual values on the test data from 03-10-2016 to 30-12-2016.

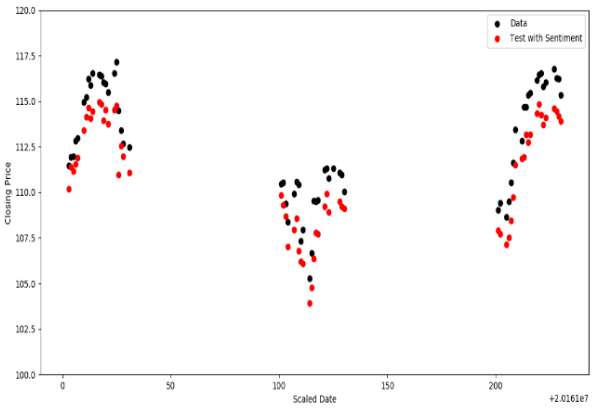


Figure 8. Values AAPL stock generated by the Perceptron network fitted on the actual values from 03-10-2016 to 30-12-2016

Table 10 is the tabular representation of the comparison of actual vs predicted values for the period 03-10-2016 to 30-12-2016. In this 1/4th dataset of 04-01-2016 to 30-12-2016 is taken as test data. MAPE value in this case is 1.5830. So, this model is 98.42% accurate, which is considerably higher than the previous model.

Table 10. Last 15 Actual and predicted values of the period 03-10-2016 to 30-12-2016

Date	Actual	Predicted
09-12-2016	113.4580	109.7141
12-12-2016	112.8109	111.4948
13-12-2016	114.6927	111.8714
14-12-2016	114.6927	111.9265
15-12-2016	115.3200	113.1732
16-12-2016	115.4693	112.7562
19-12-2016	116.1364	113.1682
20-12-2016	116.4451	114.3335
21-12-2016	116.5546	114.8210
22-12-2016	115.7879	114.2386
23-12-2016	116.0169	113.6912
27-12-2016	116.7538	114.0956
28-12-2016	116.2559	114.5863
29-12-2016	116.2260	114.4405
30-12-2016	115.3200	114.1875

The reason behind considering two different model with different size of datasets are to compare the results which are quite noticeable. The results suggest that stock prices prediction is more likely effective for the short term. The reason behind this is the time series nature of the data, in first case, model is trained on perceptron network, with the data of three years 2013, 2014, 2015 and predicted the value for the fourth year i.e. 2016. In this case, perceptron network has no knowledge about the price patterns of the year 2016 as it is only familiar with the price patterns of 2013, 2014 and 2015. Thus, it has a high error value as compared to second case, where model is trained with specifically 2016th dataset, and after that predicting the value for any given instance in 2016 works better than the previous case. These results clearly suggesting that stock prices predictions are more effective for shorter period of time.

4. Conclusion

In this paper, the authors have focused on stock market data's preprocessing with the help of HIVE, Hadoop ecosystem. HIVE can do work extremely fast as it divides the query in several map-reduce job which can be executed in parallel as well as it can store large amount of data easily with the help of HDFS. Thus, HIVE is much more effective than any other relational database system. Further, the stock price prediction is based on the sentiment analysis of news headlines and historical stock data. In this model, two different scenarios have been taken, one with training on longer period of data (three years data) and another on shorter period of data (one-year data). From the experiments it has been observed that accuracy reaches 91% in first case whereas 98% in other

case, which clearly indicates that Stock price prediction model is more effective for shorter period of data.

Other than this in future, the words which are actually affecting the stock price can also be extracted so that those can be used to find the news data. In that way the sentiment analysis can be improved more. These kinds of solution will definitely make a better prediction system for stock market.

References

[1] Parmar, R.R., Roy, S., Bhattacharyya, D., Bandyopadhyay, S.K. and Kim, T.H., 2017. Large-scale encryption in the Hadoop environment: Challenges and solutions. IEEE Access, 5, pp.7156-7163.

[2] Moghaddam, A.H., Moghaddam, M.H. and Esfandyari, M., 2016. Stock market index prediction using artificial neural network. Journal of Economics, Finance and Administrative Science, 21(41), pp.89-93.

[3] Canelas, A., Neves, R. and Horta, N., 2012, July. A new SAX-GA methodology applied to investment strategies optimization. In Proceedings of the 14th annual conference on Genetic and evolutionary computation (pp. 1055-1062). ACM.

[4] Deshmukh, B.G., Jain, P.S., Patwardhan, M.S. and Kulkarni, V., 2016, August. Spin-offs in Indian stock market owing to twitter sentiments, commodity prices and analyst recommendations. In Proceedings of the International Conference on Advances in Information Communication Technology & Computing (p. 77). ACM.

[5] Khashei, M.; Hajirahimi, Z. Performance evaluation of series and parallel strategies for financial time series forecasting. Financ. Innov. 2017, 3, 1–24.

[6] Khedr, A.E.; Salama, S.E.; Yaseen, N. Predicting Stock Market Behavior using Data Mining Technique and News Sentiment Analysis. Int. J. Intell. Syst. Appl. 2017, 7, 22–30.

[7] Bini, B.S.; Mathew, T. Clustering and Regression Techniques for Stock Prediction. Procedia Technol. 2016, 24, 1248–1255.

[8] Desai, R.; Gandhi, S. Stock Market Prediction Using Data Mining. Int. J. Eng. Dev. Res. 2014, 2, 2780–2784

[9] Zhao, L.; Wang, L. Price Trend Prediction of Stock Market Using Outlier Data Mining Algorithm. In Proceedings of the IEEE Fifth International Conference on Big Data and Cloud Computing, Dalian, China, 26–28 August 2015.

[10] Al-Radaideh, Q.I.; Assaf, A.A.; Alnagi, E. Predicting Stock Price Using Data Mining Technique. In Proceedings of the International Arab Conference on Information Technology (ACIT'2013), Katumu, Sudan, 17–19 December 2013; pp. 1–8.

[11] Patel, J.; Shah, S.; Thakkar, P.; Kotecha, K. Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. Expert Syst. Appl. 2015, 42, 259–268.

[12] Nayak, S.C. and Misra, B.B., 2018. Estimating stock closing indices using a GA-weighted condensed polynomial neural network. Financial Innovation, 4(1), p.21.

[13] Sharma, M., G. Singh, and R. Singh. (2017). Stark Assessment of Lifestyle Based Human Disorders Using Data Mining Based Learning Techniques. IRBM 38(6):305-324.

[14] Mitra, A., Banerjee, P.S., Roy, S., Roy, S. and Setua, S.K., 2018. The region of interest localization for glaucoma analysis from retinal fundus image using deep learning. Computer methods and programs in biomedicine, 165, pp.25-35.

[15] Mittal, M., Goyal, L.M., Sethi, J.K. and Hemanth, D.J., 2018. Monitoring the Impact of Economic Crisis on Crime in India Using Machine Learning. Computational Economics, pp.1-19.

[16] Kaur B., Sharma M., Mittal M., Verma A., Goyal L. M., Hemanth D. J., 2018. An improved salient object detection algorithm combining background and foreground connectivity for brain image analysis. Computers and Electrical Engineering, 71, pp. 692-703

[17] Sharma, M., Sharma, S. and Singh, G., 2018. Performance Analysis of Statistical and Supervised Learning Techniques in Stock Data Mining. Data, 3(4), p.54.

[18] Mittal, M., Singh, H., Paliwal, K.K. and Goyal, L.M., 2017, December. Efficient random data accessing in MapReduce. In Infocom Technologies and Unmanned Systems Trends and Future Directions, 2017 International Conference on (pp. 552-556). IEEE.

[19] Mittal, M., Balas, V.E., Goyal, L.M. and Kumar, R., (2018), *Big Data Processing using Spark in Cloud*, 43, (Singapore, Springer Nature Pte Ltd.).

[20] Slamet, C., Andrian, R., Maylawati, D.S.A., Darmalaksana, W. and Ramdhani, M.A., 2018, January. Web Scraping and Naïve Bayes Classification for Job Search Engine. In IOP Conference Series: Materials Science and Engineering (Vol. 288, No. 1, p. 012038). IOP Publishing.

[21] Sezer, O.B., Ozbayoglu, A.M. and Dogdu, E., 2017, April. An artificial neural network-based stock trading system using technical analysis and big data framework. In Proceedings of the South East Conference (pp. 223-226). ACM.