Stock Market Real Time Recommender Model Using Apache Spark Framework

Mostafa Mohamed Seif(✉), Essam M. Ramzy Hamed(✉), and Abd El Fatah Abdel Ghfar Hegazy(✉)

Arab Academy for Science, Technology and Maritime Transport, Cairo, Egypt
mmseif87@yahoo.com, {essam.hamed,Ahegazy}@aast.edu

Abstract. The stock market is considered a complicated and nonlinear system. Now stock market prediction is recognized as an attracting point for financial investors. The historical price is not considered as the main factor to predict the stock market trend. There are many other factors such as politics and natural events that affect social media environments like Twitter and Facebook which generate huge datasets needed data analysis to extract the polarity of these data and its effectiveness on the stock market. On the other hand, these data may be unstructured and need special handling on storing and processing. This paper proposes a real-time forecasting of stock market trends based on news, tweets, and historical price. A supervised machine learning algorithms used to build this model. Historical price will be combined with sentiment analysis to build the hybrid model based on Apache Spark and Hadoop HDFS to handle big data (structured and unstructured) generated from social media and news websites. The proposed model works in two modes; the offline mode that works on historical data including today's data after ending of a stock market session, and real-time mode that works on real-time data during the stock market session. This model increases the accuracy of prediction due to the additional features added by sentiment analysis on StockTwits and market news data. In addition, this model enhances the performance of handling this data set due to parallel processing occurred on data using Apache Spark.

Keywords: Sentiment analysis · Supervised learning · Apache Spark · Big data Hadoop · HDFS · StockTwits

1 Introduction

The stock market prediction has become a very important topic nowadays as it is used by business people. There are two traditional methods used to predict stock market fundamental analysis and technical analysis. Fundamental is a technique used to evaluate the security by studying everything that can affect the security's value, including economic influences (like the overall economy and industry conditions) and individually influences (like the management of companies and financial condition). A fundamental analysis looks like the balance sheet, loss statement, the profit, financial ratios and other data that could be used to predict the future of a company [1–3].

A. E. Hassanien et al. (Eds.): AMLTA 2018, AISC 723, pp. 671–683, 2018.
https://doi.org/10.1007/978-3-319-74690-6_66

The main disadvantage of fundamental analysis technique is time-consuming, it can also get you on board a good stock but at the wrong time and you may need to hold on to the stock for a long time. Technical analysis has nothing to do with the financial performance of the underlying company. In this method, the analyst simply studies the trend in the share prices. The underlying assumption is that market prices are a function of the supply and demand for the stock, which, in turn, reflects the value of the company. This method also believes that historical price trends are an elevator of the future performance. There are some drawbacks in using technical analysis as well. They are as difficult to master and check their accuracy and validation against a biased view. The main issues on these two methods are not considered as the mirror to events that happen in any country during market session, so this paper introduces another prediction model to enhance the accuracy of prediction and reflect real-time market events and their effect on prices. The model uses sentiment analysis to analyze the data generated from news websites and social media using supervised machine learning algorithms and combine the results with historical price as an additional feature to find the final classification result as binary classification up or down. On the other hand, some well-known companies like Google, Amazon, LinkedIn, and Yahoo have generated a huge amount of structured and unstructured data every day that needs huge storage to store these big data and high-performance processing environments to process it in few minutes.

The problem is how to enhance the accuracy of stock market prediction using real-time data generated from sources like social media and news channels, store it on data storage, and make parallel processing on it to enhance the recommendation delivery time to the stock market trader. The proposed model built on Hadoop Distributed File System (HDFS) as data storage and use Apache Spark framework on data processing using Resilient Distributed Dataset (RDD) to parallelize data processing, make the best utilization of resources, and get quick prediction results.

1.1 Literature Review

In the stock market, you have to take the right decision on the right time to gain profit and maximize your wealth. This right decision will be through buying or selling a stock and the decision taken depends on many factors. Nowadays most of the stock analytics predict stock prices depending on historical data, but with the huge amount of data today and data variety, it must use new data sources to increase the accuracy of prediction and find new ways to take a right decision. However, this takes place through handling all types of data at the same time, with these huge amounts of data need a new approach and brilliant data processing framework.

Nayak et al. [4] used the neural network to predict stock market price using Hadoop and MapReduce by building two models one for daily prediction based on the combination between historical price and tweets sentiment analysis and the accuracy was up to 70%. The second model finds the stock market trend correlation between two months and the correlation result was very small. Mukesh and Rohini [5] used a neural network and Hadoop HDFS to compare between two algorithms and proved that Least Square Algorithm better than Sigmoid Algorithm by calculating Root Mean Square "RMS" error. He also proved that using Hadoop MapReduce and Hadoop HDFS in parallel

processing is better than using single node processing. Bachhav et al. [6] presented a method to make sentiment analysis using machine learning techniques and Hadoop HDFS as data storage and analyze online feedback of users from online sites to detect impressions and make sentiment analysis of a specific topic. Khairnar and Kinikar [7] used Support Vector Machine (SVM) - a machine learning technique - and Hadoop HDFS to prove that the accuracy of sentiment analysis using Latent Semantic Analysis "LSA" is more accurate than using SVM only and that it enhances the processing of data by using Hadoop MapReduce. Ghaiehchopogh et al. [8] applied linear regression algorithms using Relational Database Management System (RDBMS) to calculate the relation between two variables "average price and volume" per day to predict the next stock market price after comparison occurred between results observed and stock market values, he obtained a similarity of 61.35%.

The remainder of this paper is structured as follows: Sect. 2, presents the proposed stock market real time recommended model. Section 3, presents experimental results. Section 4, discusses conclusion.

2 The Proposed Stock Market Real Time Recommended Model

The proposed model is designed based on three main phases: data acquisition phase, data storage phase and data analysis phase as shown in Fig. 1. Apache Spark [9, 10] is used to handle these phases. It is the newer framework built on the same concepts and techniques of Hadoop. However, Hadoop is the best solution for large data processing; it drops on some scenarios especially on iterative algorithms. Another problem on Hadoop is that it does not cache intermediate data for faster performance. It releases the data to the disk between each step. In contrast, Spark uses RDD to persist the data on the worker's memory and the concept of caching to avoid reproducing all the pipeline processes when the task is failed. Spark applications run as isolated sets of processes on a cluster coordinated by the SparkContext object in the main program (the driver program). Specifically, to run on a cluster, the SparkContext connects with Cluster Manager that allocates resources across applications. As shown in Fig. 2, when the SparkContext is connected, it acquires executors on nodes in the cluster, which run computations and store data for the application. Then, it sends the application code to the executors. Finally, SparkContext sends tasks for the executors to run.

On data acquisition phase, the used dataset consists of three sources: StockTwits, Market News, and Historical Prices. They have been collected in the interval from the period 1/2/2013 to 30/6/2016. StockTwits is used as the source of social media data. Its content is focused on the discussion about stock markets. It is believed that the user on StockTwits has good experience to write tweets related to stock markets and financial topics. StockTwits creates $Ticker tag to enable and organize "Streams" of information around stocks and markets across the web and social media. Every tweet includes information about creation date, message content and message source.

StockTwits is collected for three companies Apple ($AAPL), International Business Machines ($IBM) and Google ($GOOG). Market news is used to reflect the pulse of the market and mirror the events occurring on the market during a stock market session.

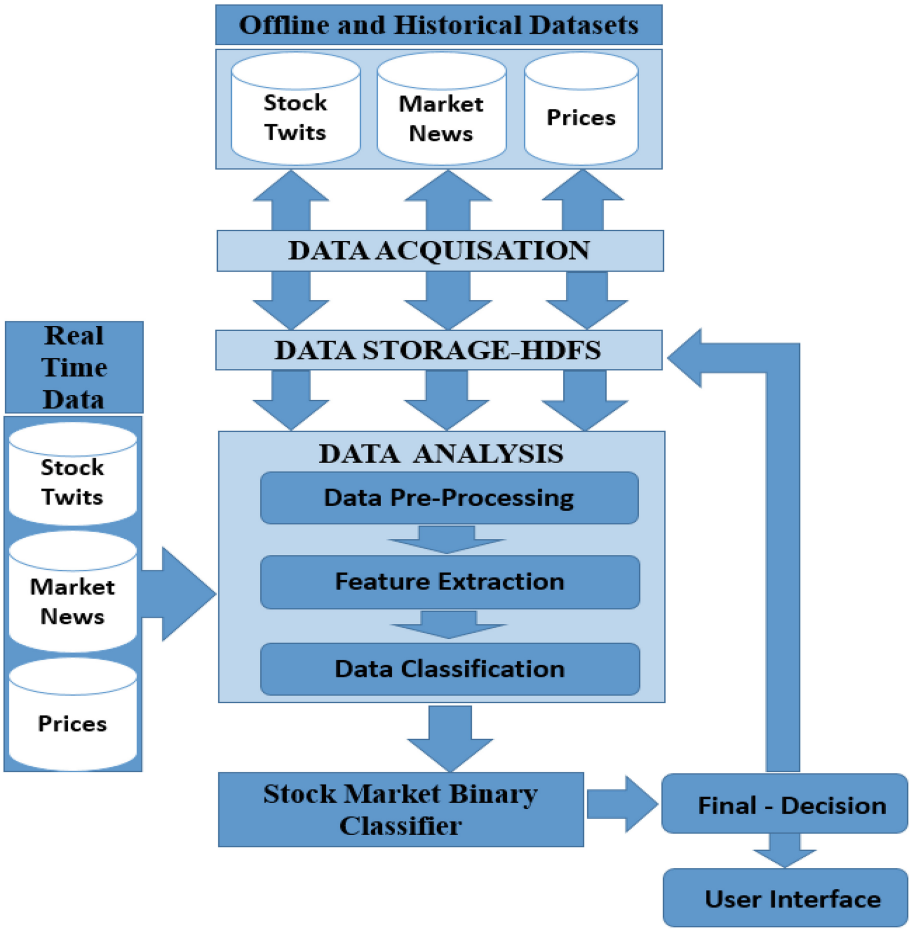

Fig. 1. The proposed framework

Examples of market news are politics news and public events. Historical prices are used as the main data source of stock market prediction algorithms. Yahoo finance is used to get data related to $AAPL, $IBM and $GOOG stocks and retrieve data related to stock prices details.

The proposed model works with two modes: real-time mode and offline mode. In real time mode, the model is running only on real-time data. After creation of both StockTwits or market news, an event fired and the model triggered to work and classify tweet or news body to get its polarity positive or negative then combines the result of classification with price of this stock at this moment to give final recommendation to trader, so on this model the main features needed are like open price, high price and low price from historical stock prices data, and from tweets and market news data the message body and date are only taken. Offline mode works after a stock market session is ended because new features are already generated like close price, adjusted close

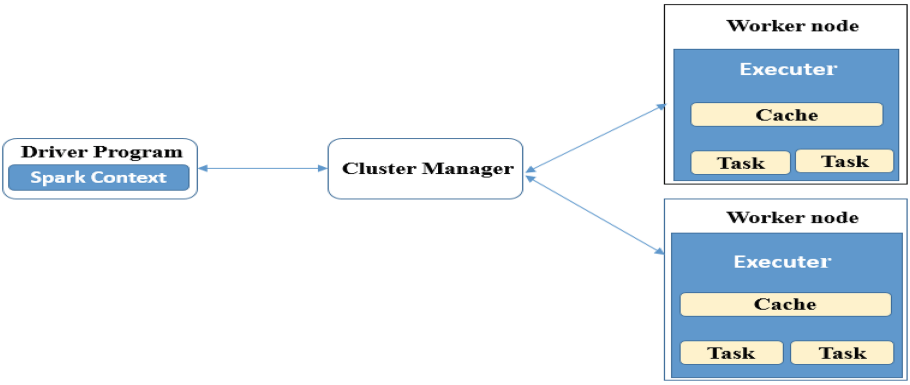

Fig. 2. Spark architecture

price- AdjClose- (which contains close price with considerations to any dividends occur on this stock) and total volumes plus extra features will be generated from the main features that already exist like the change that occurs on close price between current close price and number of days. On the other hand, the accumulative sentiment analysis of tweets and news generated during this day will be considered as features in offline mode. All these features will be explained on feature extraction section.

On data storage phase, HDFS is used to store data collected from multiple data sources. HDFS is the file system component of the Hadoop framework. HDFS is designed to play down the storage overhead and mine a large amount of data on distributed fashion hardware. Every file stored on HDFS is divided into 128 MB with three copies stored on three different nodes on Hadoop cluster. As shown in Fig. 3, the cluster has two main components: Name Node and Slave Nodes. Name Node contains the

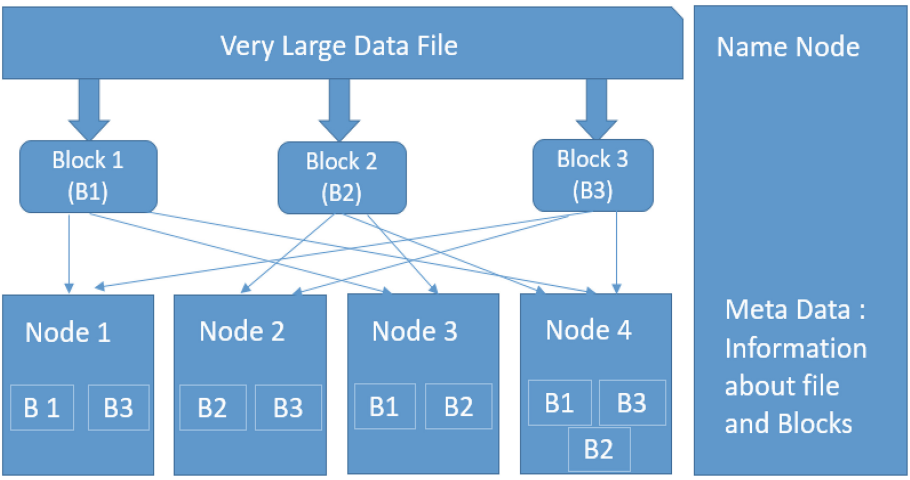

Fig. 3. HDFS architecture

Metadata file which contains the location of each block. Slave Nodes contains the data themselves [11, 12].

In data analysis phase, the meaningful knowledge is extracted from data stored. A set of RDD transformations and actions are implemented on a dataset to make data pre-processing. Spark offers many machine-learning algorithms already implemented on MLib which is Apache Spark scalable machine learning library and it is developed as part of Apache Spark Framework. It contains many implemented machine learning techniques such as classification, clustering, and regression. Spark offers many Application Programming Interface (APIs) in Scala, R and Python languages which are run on HDFS. The python's API is used to build this model. The analysis phase consists of three steps: data pre-processing, feature extraction and data classification.

Data Pre-processing: Before carrying out any "mining" activities, text in StockTwits and news needs to be prepared or pre-processed in a way that can enable mining algorithms to be applied to it. There are many techniques used to pre-process the data before using it, like Tokenization, Case Folding, Lemmatization, and Stemming.

Feature Extraction: It involves reducing the number of resources required to describe a large set of data. On StockTwits the features used are the message content, message source, message time and cash tag of a tweet. For market news, the features used are news content and date. On stock prices, the features used in offline mode are the close price, AdjClose, volume but low price, open price, and high price are used on both offline and online modes. Additional features have been extracted from data already existing only on offline mode. The list of this additional features added is as follows:

Days Return: Percentage difference of adjusted close price of i-th day and (i − 1)th day.

$$\text{Return(i)} = \text{AdjClose(i)} - \text{AdjClose(i} - 1)/\text{AdjClose(i} - 1) \tag{1}$$

Where,

- *Return(i)* is the change occurred from one day ago.
- *AdjClose(i)* is the adjusted close price today.
- *AdjClose(i − 1)* is the adjusted close price yesterday.

Multiple Day Returns: Percentage difference of AdjClose Price of i-th day compared to (i-delta)th day. *Example*: 2-days Return is the percentage difference of AdjClose price of today compared to the one of two days ago.

$$\text{Return(n)} = \text{AdjClose(i)} - \text{AdjClose(i} - \text{n})/\text{AdjClose(i} - \text{n}) \tag{2}$$

Where,

- (*n*) is the number of days.
- *Return(n)* is the change occurred from n days ago.
- *AdjClose(i)* is the adjusted close price today.
- *AdjClose(i−n)* is the adjusted close price on days within n days.

Returns Moving Average: Average returns on last delta days. Example: 2-days Return is the percentage difference of Adjusted Close Price of today compared to the one of 2 days ago.

$$\text{MovAvg(n)} = \text{Return1} + \text{Return2}/\text{Return(n)} \tag{3}$$

Where,

- (*n*) is the number of days.
- *MovAvg(n)* is the Returns average of n days.

Data Classification: For StockTwits and Market News, Sentiment analysis is used to analyze data and it is the main method to get the polarity of human opinion from comments they write. Machine learning algorithms are used to implement sentiment analysis on social media data, Naïve Bayes classifier used to implement classification of StockTwits and market news datasets. Naïve Bayes is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naïve Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Naïve Bayes model is easy to build and particularly is useful for very large datasets. Along with its simplicity, Naive Bayes is known for outperforming even with highly sophisticated classification methods. Bayes theorem provides a way of calculating posterior probability P(cla) from P(c), P(a) and P(alc). For historical stock prices combined with the results extracted from sentiment analysis classifiers, machine learning supervised classification techniques are used like Random Forest, Support Vector Machine, and Logistic Regression.

$$\text{P(c|a)} = \text{P(a|c)}\,\text{P(c)}/\text{P(a)} \tag{4}$$

Where,

- *P(c|a)* is the posterior probability of class (c, target) given predictor (a, attributes).
- *P(c)* is the prior probability of class.
- *P(a|c)* is the probability of predictor given class.
- *P(a)* is the prior probability of predictor.

3 Experimental Results

For testing and training data, two datasets are used, one for the real-time mode that contains real-time features and the other contains features of offline mode. There are three channels of data provided on this dataset. The main channel is the historical prices. Yahoo finance website is used as the provider of stock prices of $AAPL, $IBM, and $GOOG [13]. StockTwits website is used as the channel to collect tweets for the stock market [14]. The last channel used to fetch market news is Reddit world channel which contains historical news headlines [15].

All data crawled on date interval from 1-2-2013 to 30-6-2016. The dataset is divided into 80% for training and 20% for testing. Weka tool is used to test proposed model.

Weka is an open source tool used in data mining that contains many already-implemented machine learning algorithms. The dataset of $AAPL is classified on Weka using multiple classifier algorithms like Naïve Bayes (NB), Logistic Regression (Log-Reg), Decision Tree (DT), K-Nearest Neighbor (KNN), Support Vector Machine (SVM) and Random Forest (RF) to choose the best three of them and after run, RF, Log-Reg, and SVM have been chosen as the best algorithms as shown in Fig. 4. Requests python package is used to stream data from StockTwits and Reddit world channel. On the other hand, Yahoo Finance Python package is used to fetch data of current prices for a specific stock. Two sentiment analysis classifiers have been built, one of them to classify any new StockTwits and the other for market news. The results extracted from two classifiers combined with stock price data that entered to another classifier to get final recommendation.

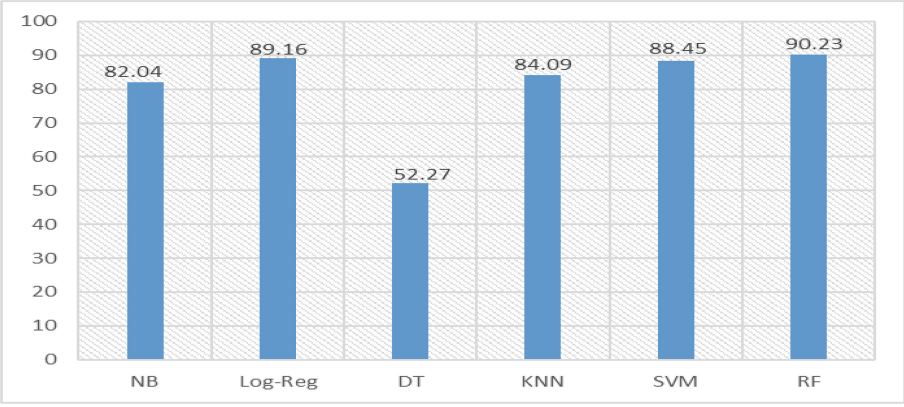

Fig. 4. Accuracy of data mining algorithms

The proposed model trained on an offline model using offline mode dataset and trained in online mode using real-time mode dataset. The model works on offline mode after the stock market session ended and fetch the complete stock prices data that contains close prices and the total volume of the day. The model works on real-time mode during market session time. The model trained by fetching data from sources and storing it on HDFS then transferring it to analyze through three predefined phases to be ready as training data for sentiment analysis classifier and the results combined with stock prices to enter as the data source into final classification phase to give next day recommendation. The same steps will be implemented on real-time model using real-time data sets but do not contain any close price or volumes data because during stock market session this data are still unknown.

After these phases are finished, two Stock market binary classifiers are generated, one for offline mode and the other for a real-time mode. The results of the proposed model are compared with weka tool results. Figure 5 constructs the comparison between Data mining techniques implemented on weka tool and proposed model using offline dataset without adding sentiment analysis features. Figure 6 shows the same comparison

results on the offline dataset with sentiment analysis results. Figure 7 uses real-time dataset on comparison without sentiment analysis features, and Fig. 8 makes the last comparison using sentiment analysis features.

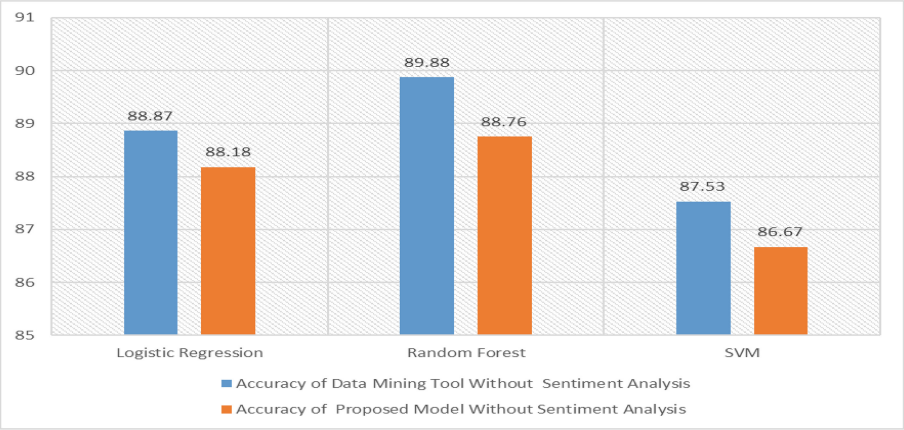

Fig. 5. Accuracy results of data mining techniques vs proposed model without sentiment analysis features (offline dataset)

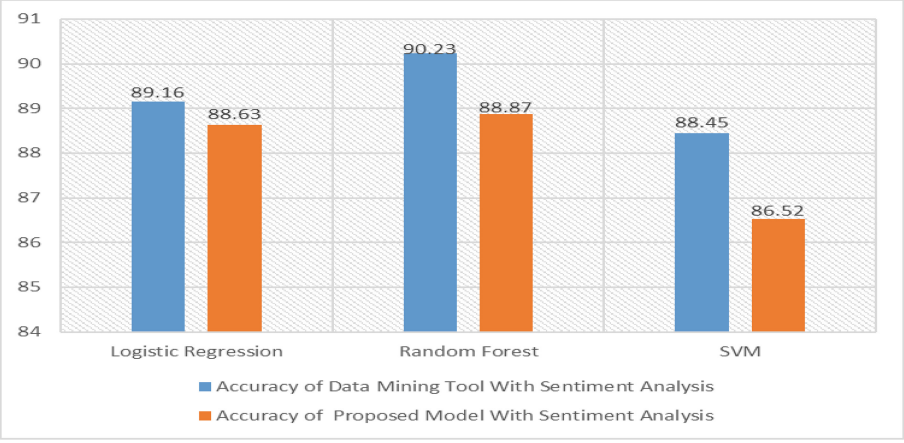

Fig. 6. Accuracy results of data mining techniques vs proposed model with sentiment analysis features (offline dataset)

The real-time proposed model was run on 22/9/2017 from 12:08 PM to 12:21 PM and for every new tweet created on StockTwits or market news published on Reddit World Channel, the proposed model triggered to implement algorithm by preprocessing tweet or news and makes sentiment analysis to calculate the polarity of tweet or market news. The model then fetches the current open price, low price, and high price and combines it with sentiment analysis result to compose the feature vector. The proposed

model takes this feature vector and implements the prediction and the final result is binary result "1" or "0". Value "1" represents the recommendation as buying and "0" represent recommendation as they sell. Figure 9 shows the recommendations given to trader on three companies Apple, IBM and Google.

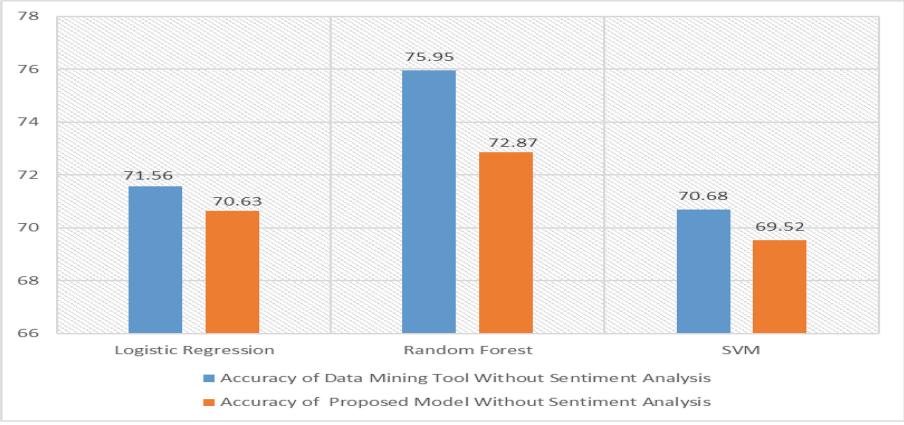

Fig. 7. Accuracy results of data mining techniques vs proposed model without sentiment analysis features (real-time dataset)

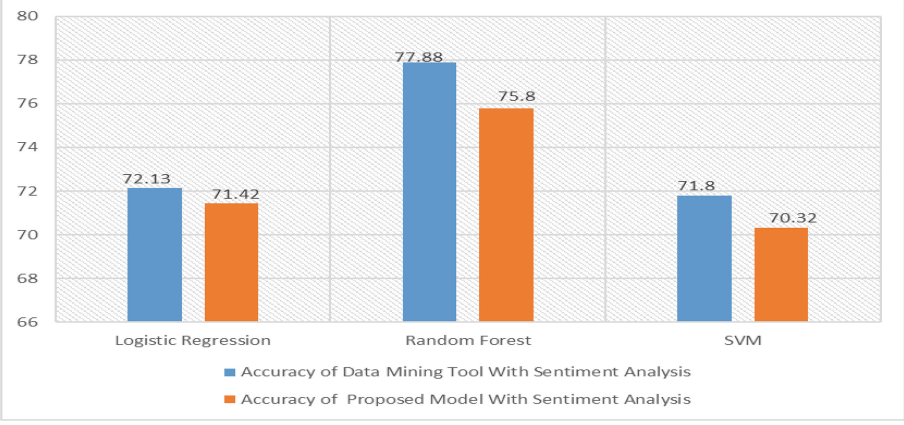

Fig. 8. Accuracy results of data mining techniques vs proposed model with sentiment analysis features (real-time dataset)

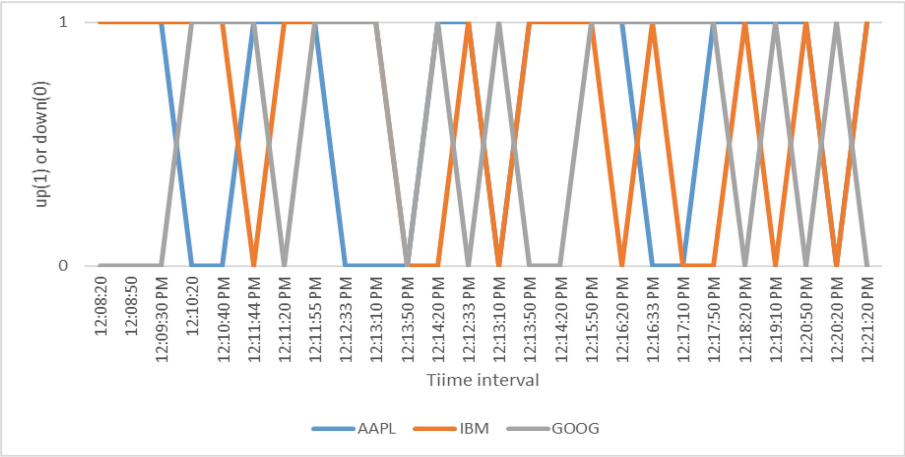

Fig. 9. Model results on three Stock AAPLE, IBM and Google

This case study ran on spark cluster installed on Amazon EC2. According to the dataset used in this case. The time consumed to train model to generate market news sentiment analysis classifier, StockTwits classifier, and the final stock market binary classifier as shown in Fig. 10.

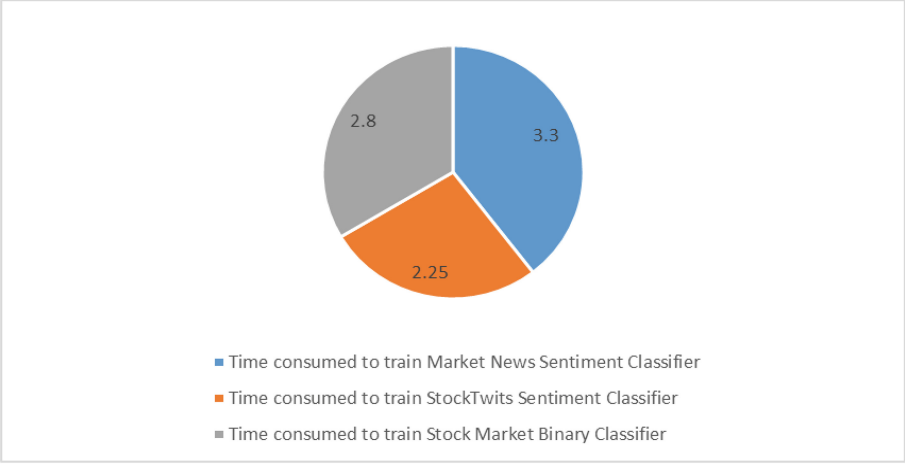

Fig. 10. Spark processing time

4 Conclusion

This work presents a model to predict stock market trend and gives the recommendation to the trader using the combination between stock price and sentiment analysis on social media data and market news under big data environment. Three types of the dataset used

historical stock prices, StockTwits and market news are fetched from multiple data sources and stored in HDFS. Sentiment analyzer has been built to analyze StockTwits and market news data, then the features extracted from them are combined with stock price features and compose another dataset used to build our new classifier. Our model works on two modes, offline mode, and real-time mode. The offline mode works on end of day data like close price, AdjClose price, volume, the accumulative sentiment analysis of all twits and news during the day in addition to normal stock price features. On the other hand, the real-time mode works on live features like open price, high price, low price plus fresh tweets and news generated during the stock market session. All the classified algorithms are implemented with Apache Spark using Apache Spark machine learning libraries, entitled MLib to enhance the performance of processing. The result extracted from Weka tool is compared with the result observed from proposed model and it seems to be more relevant to it.

References

1. Moosa, I., Li, L.: Technical and fundamental trading in the Chinese stock market: evidence-based on time-series and panel data. Emerg. Mark. Financ. Trade **47**(1), 23–31 (2011)
2. Venkatesh, C.K., Tyagi, M.: Fundamental analysis as a method of share valuation comparison with technical analysis. Bangladesh Res. Publ. J. **5**(3), 167–174 (2011)
3. Drakopoulou, V.: A review of fundamental and technical stock analysis techniques. J. Stock Forex Trad. **5**, 163 (2015). https://doi.org/10.4172/2168-9458.1000163
4. Nayak, A., Pai, M.M.M., Pai, R.M.: Prediction models for indian stock market. In: Twelfth International Multi-Conference on Information Processing-2016 (IMCIP-2016), Procedia Computer Science, vol. 89, pp. 441–449 (2016)
5. Mukesh, Rohini, T.V.: Market price prediction based on neural network using hadoop mapreduce technique. In: Computational Systems for Health & Sustainability
6. Bachhav, C., Gite, M., Jadav, K., Malode, K.: Sentimental analysis on big data. Int. J. Res. Eng. Appl. Manag. (IJREAM) **1**(1) (2015)
7. Khairnar, J., Kinikar, M.: Sentiment analysis based mining and summarizing using SVM-MapReduce. Int. J. Comput. Sci. Inf. Technol. (IJCSIT) **5**(3), 4081–4085 (2014)
8. Ghaiehchopogh, F.S., Bonaband, T.H., Rezakhaze, S.: Linear regression approach to prediction of stock market trading volume: a case study. Int. J. Manag. Value Supply Chains (IJARCE) **4**(3), 25–31 (2013)
9. Sasmita, P., Lenka, R.K., Stitipragyan, A.: A hybrid distributed collaborative filtering recommender engine using apache spark. Procedia Comput. Sci. **83**, 1000–1006 (2016)
10. Etaiwi, W., Biltawi, M., Naymat, G.: Evaluation of classification algorithms for banking customer's behavior under apache spark data processing system. Procedia Comput. Sci. **113**, 559–564 (2017)
11. Gerard: Hadoop Essentials – The Eight Things You Need To Know. Working Analytics (2015). http://workinganalytics.com/hadoop-essentials-by-cloudera-eight-lessons-learned/. Accessed 30 Sep 2017
12. White, T.: The Hadoop distributed file system. In: Hadoop: The Definitive Guide, pp. 45–80. O'Reilly&Associates, Sebastopol (2012)

13. Yahoo Finance: Historical Price. https://finance.yahoo.com/lookup. Accessed 30 Aug 2017
14. StockTwits: https://stocktwits.com/home#people-and-stocks. Accessed 30 Aug 2017
15. RichardChen: Sentiment effect on stock price. https://www.kaggle.com/otwordsne1/sentiment-effect-on-stock-price-draft/data. Accessed 30 Aug 2017