Journal of Universal Computer Science, vol. 26, no. 9 (2020), 1128-1147
submitted: 2/12/18, accepted: 27/9/20, appeared: 28/9/20

Comparative Study of Real Time Machine Learning Models for Stock Prediction through Streaming Data

Ranjan Kumar Behera, Sushree Das, Santanu Kumar Rath
(National Institute of Technology Rourkela, India
jranjanb.19@gmail.com, sushreedas008@gmail.com, skrath@nitrkl.ac.in)

Sanjay Misra
(Atilim University, Ankara, Turkey and Covenant University, Ota, Nigeria
ssopam@gmail.com)

Robertas Damasevicius
(Vytautas Magnus University, Kaunas, Lithuania
robertas.damasevicius@vdu.lt)

Abstract: Stock prediction is one of the emerging applications in the field of data science which help the companies to make better decision strategy. Machine learning models play a vital role in the field of prediction. In this paper, we have proposed various machine learning models which predicts the stock price from the real-time streaming data. Streaming data has been a potential source for real-time prediction which deals with continuous flow of data having information from various sources like social networking websites, server logs, mobile phone applications, trading floors etc. We have adopted the distributed platform, Spark to analyze the streaming data collected from two different sources as represented in two case studies in this paper. The first case study is based on stock prediction from the historical data collected from Google finance websites through NodeJs and the second one is based on the sentiment analysis of Twitter collected through Twitter API available in Stanford NLP package. Several researches have been made in developing models for stock prediction based on static data. In this work, an effort has been made to develop scalable, fault tolerant models for stock prediction from the real-time streaming data. The Proposed model is based on a distributed architecture known as Lambda architecture. The extensive comparison is made between actual and predicted output for different machine learning models. Support vector regression is found to have better accuracy as compared to other models. The historical data is considered as a ground truth data for validation.

Key Words: Spark Streaming, NodeJS, Twitter API, Lambda Architecture, MLlib

Categories: H 1.0, H 1.1, H 1.2, H 3.2, H 3.4, H 3.5

1 Introduction

Through the technological advancement, interaction between people is increasing day by day which leads to generation of large volume of data at high velocity. A massive amount of data are constantly generated through multiple sources like sensors, mobile devices, smart phones and other computer gazettes which lead to abundance of information that are valid only for short span of time. Hence the volatility of data becomes one of the important features, which needs attention towards immediate processing for extracting useful information. Unlike traditional processing tools, where data is first stored and processed in batch mode, streaming analytics avoids storing of data and tries to process the data while in motion. A number of applications require real time analysis of stream data which are limited by the capabilities of processing tools used earlier. Stock prediction, weather forecasting, product recommendation, tweets recommendation for user are some of the widely used applications that require real time processing of streaming data. Data stream may be classified into either online or offline stream. Online streaming data can be used in real world applications like network traffic monitoring, fraud detection from database transaction etc. These require hard real time processing, where as offline data stream corresponding to the logs which can be collected from various sources like e-commerce websites, financial transactions data from banking sector, medical reports from health care department etc. They can be analysed through the soft real-time environment.

Software architectures such as Master-Slave Model, Peer to Peer, Publish-Subscribe etc rarely focus on the non-functional properties such as latency, scalability and fault tolerance in a collective manner. However, with the advent of technology and innovation, analysis and prediction of stock market prices which streams over various financial sites seems to be feasible by ingesting data through various applications like Kafka [Thein 2014], Flume [Hoffman 2013] etc. The ingested data is then preprocessed to prepare a regression model that can predict the rise or fall of the stock price in the market. In this study, Twitter data and Google Finance Data are considered as the source of information for knowing the daily prices and the sentiments of the public over a particular brand which shall enable in predicting the price of the stock for a particular period of time. The objective of this study is to implement the regression models of sentiment analysis over streaming data through tools such as Spark MLLIB and Spark Streaming which shall prove to be fault tolerant, faster and scalable [Meng et al. 2016] [Bifet et al. 2015]. Several experiments by researchers motivate us to use machine learning techniques over the history data to prepare model which can train over the existing data and help in predicting the next set of values. One such model which was developed by Yang et al. [Yang et al. 2002] is based on statistical learning theory like support vector regression, is used for predicting the opening price for the next day. Certain other models based on classifier techniques like artificial neural networks, decision tree, K- nearest neighbour proposed by Qian and Rasheed [Qian and Rasheed 2007] were also proficient in predicting the same . The proposed idea in this paper is based on an architecture which continuously evaluate the classifier models to whom data are fetched in streaming mode. This led to the proposal of the first methodology that is an asynchronous event driven tool that can fetch the real-time data and tune the model like NodeJS [Cantelin et al. 2014].

Microblogging sites have also been considered as potential resources for sentiment analysis by number of researchers. N. oliveira et al. [Oliveira et al. 2017] have proposed a method for accessing the microbloggging data for predicting various parameters of stock market like volatility, trading volume, amount of return etc. They have adopted attention indicators and survey indices which are derived from the microbolgging data for analyzing the stock data. The major contribution for their work lies in applying Kalman Filter for merging survey resources and microblogging data in order to predict the stock returns. With the mushrooming growth in accessibility of the mobile devices and Internet connectivity, people have become more expressible regarding their opinion on everything that matters to them. And mining and analysis of this data in the form of sentiment analysis which used the concept of natural language processing proved to be useful in predicting the future stock prices. A few investigations have been done with respect to Twitter sentiment analysis, keeping in mind on the goal to portray its substance and connection to patterns. Si et al. [Si et al. 2013] have proposed a non-parametric topic based sentiment model for analyzing streaming tweeter data in order to predict the stock market. In their work. they have leverage the Dirichlet Mixture model for learning topic based tweets.

Some of the authors have also considered the deep learning for stock prediction. Ding et al. [Ding et al. 2015] have proposed a event driven deep learning model for predicting the stock prices. They have first extracted several events from the social media like blogs, tweets and other articles etc and then applied CNN and LSTM model to measure the influences of these events on stock market. Another deep learning model based on LSTM and paragraph vector has been proposed by Akita et al. [Akita et al. 2016]. They have utilized both the textual and numerical information from social media in order to have accurate financial time series forecasting. They have considered paragraph vector for representing the textual information into fixed length vector. Chong et al. [Chong et al. 2017] have presented a deep analysis of different data representation for various deep learning networks as the data representation heavily influences on the performance of the model. They have also presented a comparison of different feature extraction techniques like PCA, autoencoder and Boltzman machine on the network's ability for stock prediction.

Nguyen et al. [Nguten et al. 2015] have proposed another topic based twitter sentiment model for stock movement prediction. They have embedded the mood information which can be derived from sentiment analysis based on specific topic instead of overall sentiments. In their work they have first extract the set of topics from the text representing tweets and leverage them in sentiment analysis for predicting future stock prices. However. they have analyzed it based on the static data rather than stream processing.

Bollen et al. [Bollen et al. 2011] have proposed a model which measured the mood factor of each tweets and classified them into six different dimensions namely calm, kind, alert, happy, vital and sure. Upon several permutation and combinations that were applied over those 6 dimensions, they later confirmed that any change that is done in the feature calmness can predict the direction changes in the closing prices for the subsequent days of the Dow Jones Industrial Average Index (DJIA) [Donaldson and Kim 1993]. Chen and Lazer [Chen and Lazer 2013] affirmed the consequences of Bollen et al. and demonstrated the same with significantly elementary methods of sentiment analysis. Their experiment proved of a relationship that is existant between data that involves sentiments based on twitter data and stock price movement. Morck et al. [Morck et al. 2000] also confirmed that the social media data taken from various networking sites having a stronger impact on the stock market. Mittal and Goel [Mittal and Goel 2012] also established their work for checking if there is a connection between public opinion and trading system. Their outcomes are in concurrence with that of the outcomes of Bollen et al. [Bollen et al. 2011]. However, they demonstrated that the mood like calm and happiness has a decent connection with the Dow Jones Industrial Average values [Cutler et al. 1988]. Daily sentiments were calculated on the aggregated data collected from multiple sources such as Google Finance, Twitter or Yahoo finance. In their analysis, they demonstrated that openly accessible information which are there in various microblogs, news, and popular forums do have the power of predicting the value of the stock on the next day. Sprenger et al. [Sprenger et al. 2014] after examining of around 250,000 tweets that were related to stock and found that the feelings that are hidden in the tweets are more or less related to the returns made by the exchange of that stock and the volume of the carried message also helps in predicting the exchanging volume for the subsequent days. Furthermore, they also demonstrated that clients who give better than expected investment counsel are re-tweeted followed by many users, which demonstrates their influence in microblogging forums. Hence, it can be inferred that Twitter can turn out to be a reliable source for data ingestion as suggested from the literature.

2 Underlying Architecture

Lambda Architecture has been considered as the stream processing framework upon which most of the machine learning models can effectively run. It was designed by Marz and Warren [Marz and Warren 2015] keeping in mind the general model which is followed while doing any online analysis, with a guarantee of fault tolerance and scalability. This architecture consists of three layers namely, batch layer, serving layer and speed layer as shown in Figure 1. The online data is fetched from various streams, are fed into both the layers. In the batch layer, historical data is stored along with the newly fed data that is continuously being

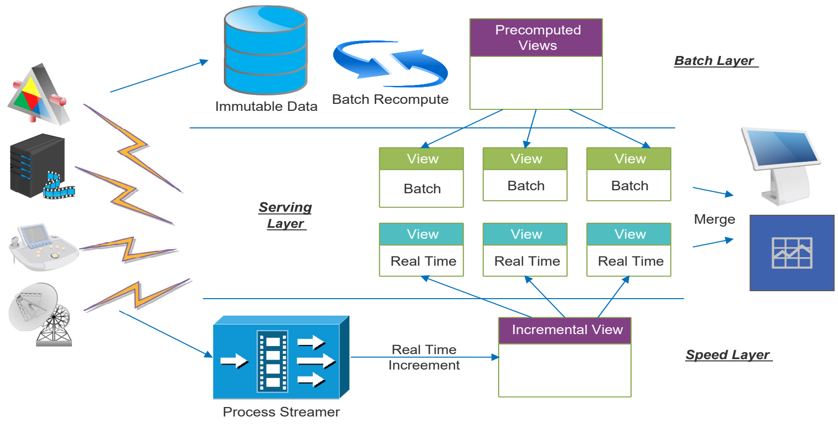


Figure 1: Lambda Architecture

fed from the streams. This data is immutable and is stored in various big data platforms like Hadoop Distributed File System (HDFS) [Behera et al.] wherein any form of structured, unstructured or semi structured data can be dumped and later transformed according to developer's need. A scalable model is prepared based on the historical data which incorporates various machine learning techniques computed through the Map-Reduce framework. These are used to make a decision based on the streaming data that is constantly provided as input to the model. As a part of integrating tools based on different architectures, the Table 1 gives a summary of the pros and cons of each one of them.

Table 1: Comparision Table for Real-Time Processing Tools

Tools	Architecture	Integration/ Query	In-Memory	Recovery
Hadoop Online	Master-Slave	Integration	Yes	No
Storm	Peer to Peer	Integration	Yes	Manual
Flume	Peer to Peer	Integration	Yes/ Backed by Files	Manual
Spark Streaming	Master-Slave	Both	Yes	Parallel
Impala	Peer to Peer	Analytics Query	Yes	Yes

3 Adopted Machine Learning Techniques

This section presents the proposed approach to the existing problem of streaming analysis and prediction of financial data. Two case studies have been considered, one involving datasets with online financial data from various websites namely Google Finance, Yahoo Finance etc for our prediction and the other case study involving Twitter dataset. The second case study is a prediction based on the sentiment attached with the ticker whereas the first case study just involves the implementation of the underlying architecture on a scalable platform.

Out of the numerous machine learning algorithms being used for data analysis, four machine learning algorithms have been chosen. The reason for choosing the four of the regression models under supervised learning is because of its technical feasibility and availability through Spark's MLlib [Meng et al. 2016]. Since the libraries for those models have been proved to be effective and scalable, they can be used to process real time data in distributed manner.

3.1 Support Vector Machine: Regression (SVR)

SVM is mainly used for classification. However, it can be used for prediction also where real number is being predicted based on the features available. The basic elements of the SVM is the set of support vectors which are to be separate as maximum as possible to set a marks for separating objects with different classes. The mean value of support vectors is considered as the margin of separation. The training process is based on symmetrical loss function which equally penalize to both high and low estimation. SVR has been found to be effective tools in stock prediction in number of research papers [Yang et al. 2002] [Henrique et al. 2018]. One of the major advantage of SVR is that it can seamlessly process dataset with large number of features.

3.2 Decision Tree

Decision tree is one of most effective decision based classifier which consist of a tree like structure, where each node represents a constraint on an attribute and the branch represents the decision taken out from the condition. They are also effective for supervised regression problem where real value is being estimated based on set of conditions. The order of feature selection from top to bottom in the hierarchical structure is based on few evaluation parameters like information gain, gini index, chi-square etc. They are simple to implement and easy to understand through visualizing tree like structure.

3.3 Random Forest Regression

Random forests algorithm is based on the concept similar to decision trees. It generate multiple decision trees in order to get more accurate results. Hence it is also called as ensemble decision tree method. Each split that is done for building a tree resides on many factors such as the value of information gain, Gini index, etc. In this paper, we have considered a random forest model which inherently generate 20 number of different decision tree which takes the features randomly but in independent manner. The final output is predicted based on the output of all such decision trees. The predicted outcomes is voted by each individual tree. Based on the majority of votes the final decision is delivered.

3.4 Polynomial Regression

Polynomial regression is a statistical learning technique whose objective is to establish a relationship between dependent variable with the independent variable with quadratic curve. It is suitable to handle more complex problem than the linear regression. It usually fits the curve using least square method to minimize the variance of unbiased estimator of the coefficients. The degree of polynomial equation should be chosen carefully as if it is less, it may not fit all the data and, if the degree is high, over-fitted condition may arise.

4 Proposed Work and Analysis

Prediction of stock market prices with high accuracy has been a challenging task to achieve. Hence, it has gathered the attention of good number of researchers. In this paper, an attempt has been made into analysing and predicting values of stock market prices for the subsequent days. Partial availability or unavailability of data for that matter poses a major drawback in ingesting data into our domain for analysis. In order to implement the proposed work, two case studies have been considered to predict the future stock price. The first case study is based on data collected from online financial websites like Google finance and Yahoo finance. Features such as opening price, closing price, volume of stock involved etc. are collected for the particular day for a particular ticker symbol that is entitled for a particular company. The second case study is based on the sentiment analysis of online review data like Tweeter data in order to predict the stock prices. The two case studies have been explained in the subsequent two sections.

4.1 Case study I: Prediction of opening price based on data collected through Financial Websites

4.1.1 Data Ingestion through NodeJS

NodeJS is a free open source software which does asynchronous event based handling by using JavaScript on the server [Cantelin et al. 2014]. For collecting, the required information from the google finance website, Few built in utilities like express, https, cors and socket.io has been considered. The basic function in NodeJS takes two parameters, namely the socket and the ticker. We have considered 15 seconds as one ticker. The cycle length of fixed to be 15 second. The cycle is repeated for ingestion of data from financial websites for testing the machine learning models. All the responses are collected by issuing a GET request from the google server with port 443 for the required ticker that needs to be listened in the localhost. Later, this data can be accessed through various data storage platforms and processed by machine learning techniques to build models and predict future values through streaming data.

4.1.2 Implementation

In this work, data from online financial websites have been collected. The historical data for opening prices for each ticker on each day from Jan,2000 till Aug,2017 was procured. In the batch processing section, 85% of the historical data has been considered for train the model and the rest of the data are used as testing the model. Moreover, in the streaming layer, data is procured for every 15 second from the Google Finance's website through NodeJS. Parameters such as current price, date, change percent, dividend are fetched which is then rendered to the model formed over the batch layer. Data streamed in this layer has been considered in order to predict the price for the next instance.

4.1.2.1 Phase-1: Batch Processing

There are a considerable measure of financial pointers which are complicated and furthermore, the prices being highly volatile. But, with the advancement in technology, there arises a chance of gaining a fortune through stock markets. This encourages researchers to look for those indicators that can very well predict the price of the subsequent days. This gives rise to the fact that with such prediction comes with a lot of risk too. Hence, history data needs to be properly scrutinized and modelled for properly predicting the values and also keeping the risk low.The following steps enumerate the process followed for data validation and analysis:

1. **Collection of Data**: In this step, the historical stock data containing various features like opening price, date, closing price, volume etc. are collected from *https: //finance.google.com/finance/historical?q=ticker*. Then, this historical data which is collected as a part of analysis is utilised for the predicting the subsequent prices of the ticker involved.

2. **Preprocessing of Data**: The pre-processing stage includes the following steps:
   - Discretization of data: This generally relates to data reduction yet has a particular importance. This method is hence very important for numerical data.

– Transformation of Data: Normalization to the existing data is carried out in order to make the ranges of a particular feature restricted to the specific limits.

– Cleansing of Data: In order to deal with the missing data, this method is adopted for filling up the null or missing values.

– Integration of Data: Integration of multiple data files into one file might be the needed in processing for certain algorithms. Thus, this method ensures the same.

Once all the cleansing and formatting of data is completed, the entire dataset is then divided into training and testing datasets for evaluation. For that matter, more recent values are generally considered for training purposes and a few percentage of it is considered for testing purposes which tends to 10-15% of the entire dataset.

3. **Feature Selection**: The size of feature set in online financial data is often large in nature which is quite difficult to process through models. Proper feature selection is necessary in order to reduce the dimension of the feature set. This is the major phase in preprocessing, where only important features are selected from multi-dimensional data. In this paper, we have used Hashtag Term Frequency and Inverse Document Frequency (TF-IDF) [Aizawa 2003] for feature selection in pre-processing steps of Spark MLib. Each tweet that to be pass is split into words using the Tokenizer. The hashtag-TF is then used for indexing each tweet in a blogs. the feature vector is then rescaled using IDF. The feature vector obtained is then passed to the machine learning components of the Mlib packages. The final size of the feature vector varies from different web logs of tweets. However, the number of features to be considered is power of 2. The minimum feature vector size is found to be 256 and the maximum feature vector size is observed to be 1024. On an average the size of the feature vector is found to be 500.

4. **Training Models**: In this phase, data procured is provided as input to the respective algorithms and trained for prediction by assigning random weights and biases.

4.1.2.2 Phase-2: Stream Processing

Stream processing is carried out in the second phase of implementation wherein NodeJS has been used for ingesting the data from online financial websites [Cantelin et al. 2014]. The injested data is then stored in HDFS for further analysis. Spark DStream (Discretized Stream) is the basic abstraction of Spark Streaming [Bifet et al. 2015]. DStream is a continuous stream of data. It receives input from various sources like Kinesis, TCP sockets, Kafka, or Flume. This can also involve a stream of data that is recreated through transforming the original stream itself. At its core, DStream is a continuous stream of RDDs (Spark abstraction), which contains data from a particular interval. Thus, an operation that is performed on a DStream becomes applicable to all the underlying RDDs which are a part of that DStream.

The implementation was carried out using Scala IDE along with the aid of NodeJS for data collection. This data is then input to a scala program that implements certain algorithms which are a part of the Spark component called the MLlib. Packages that have been developed as a part of this component have been imported in order to predict the opening price of the stock for the subsequent day.

MLlib is one of Spark‘s many components that is dedicated to machine learning (ML) [Cutler et al. 1988]. The **spark.mllib** package supports various methods for binary classification, multiclass classification, and regression analysis. The goal of this component as a part of Spark is to make practical machine learning easy and scalable. It provides tools such as:

- ML Algorithms: common learning algorithms for classification, collaborative filtering, clustering and regression.
- Utilities: linear algebra, statistics, data handling, etc.
- Pipelines: tools for constructing, tuning ML Pipelines and evaluating.
- Featurization: feature extraction, selection, dimensionality reduction, and transformation .
- Persistence: load and saving algorithms, models, and Pipelines.

4.2 Case study II: Prediction of Stock price based on Twitter data

4.2.1 Data Ingestion Through Twitter API

```
TwitterAgent.sources = Twitter
TwitterAgent.channels = MemChannel
TwitterAgent.sinks = HDFS
```

Figure 2: Components used for Data Ingestion

Social media reviews play a major source of data analytics in the field of stock prediction. Proper data collection and feature engineering is necessary to obtain fruitful information as it accounts for huge amount of streaming data in the form of event data (tweets in this case). In this research work, Apache Flume has been considered as the tool for ingestion mechanism [Hoffman 2013]. Since the rate of incoming data varies with every tick of the clock, Flume mediates between data producer and the centralized stores through the memory channel making it a reliable source for data transferring. The pivotal daemon process called the Flume Agent receives events from sources such as Twitter API and forwards it to the sink (HDFS in this case). Interceptors, a component of the Flume Agent, inspects the incoming events which are then transferred through the channel. It ensures fault handling by having two transactions per event. Preliminary naming of the components for the realization of the ingestion phase is presented as a code snippet in Figure 2. After listing out the components for the ingestion process, individual property description is made for each of the components. Enumeration of one snippet is shown in Figure 3.

```
TwitterAgent.sources.Twitter.type = Twitter (type name)
TwitterAgent.sources.Twitter.consumerKey =  GeneratedFromApp
TwitterAgent.sources.Twitter.consumerSecret = GeneratedFromApp
TwitterAgent.sources.Twitter.accessToken =   GeneratedFromApp
TwitterAgent.sources.Twitter.accessTokenSecret = GeneratedFromApp
```

Figure 3: Description of Component Feature

4.2.2 Implementation

4.2.2.1 Batch Processing Layer

The sentiment analysis of history data has been processed through one of the Spark's components called the MLlib (Machine Learning Library). It exploits the iterative computation performed through Spark in distributed manner. It is an RDD based API which can be used by importing the packages. These classes ensure certain functionalities that involve loading the training set and mapping them into classes after the model is trained. Several machine learning techniques like linear SVM, logistic regression, decision tree, random forest, gradient-boost tree etc. are available in MLlib, which can be helpful for sentiment analysis for stock prediction.

Table 2: Categorical Classification for the Sentiments

Sl.no	Condition	Prediction
1	$S \leq 0.0$	NOT UNDERSTOOD
2	$0.0 < S \leq 1.0$	VERY NEGATIVE
3	$1.0 < S \leq 2.0$	NEGATIVE
4	$1.0 < S \leq 3.0$	NEUTRAL
5	$3.0 < S \leq 4.0$	POSITIVE
6	$4.0 < S \leq 5.0$	VERY POSITIVE
7	$S \geq 5.0$	NOT UNDERSTOOD

4.2.2.2 Stream Processing and Serving Layer

Stanfords CoreNLP API has been considered in order to obtain the sentiments on real-time streaming data. This involves the usage of Recurrent Neural Network (RNN) which allow the model to keep the dependencies among the words in a sentence. It is fundamentally based on Java Virtual Machine (JVM) annotation pipeline framework that provides most of the popular natural language processing steps. Five datasets have been considered, each of which contains data filtered based on the tags of each company. In this experiment, prediction of the movement of stock prices for three major companies have been considered, namely, Google, Microsoft and Apple Inc. For this, at each ticker, tweets that had keywords like MSFT, GOOG, AAPL, MICROSOFT, GOOGLE, or APPLE have been filtered. Then they were segregated and collected in three separate datasets. Later, batch data from online sources like Yahoo Finance were collected for that particular time frame and company. The main class that is to be defined for such a classification is SentimentAnalysisUtils wherein the definitions of the properties such as annotators and tokenize, split, pos, lemma, parse, sentiment have been set. After importing all the necessary packages from edu.stanford.nlp, sentiment can be predicted for the tweets at any time stamp. The code snippet for the same is shown in Figure 4. This enables to calculate the weighted sentiment for each tweet. Final classification of the sentiment is processed based on the constraints shown in Table. 2.

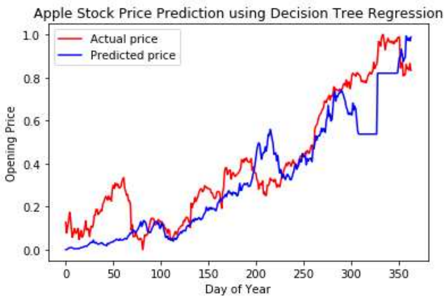
```
val sentiment = RNNCoreAnnotations.getPredictedClass(tree)
```

Figure 4: Twitter Sentiment Prediction

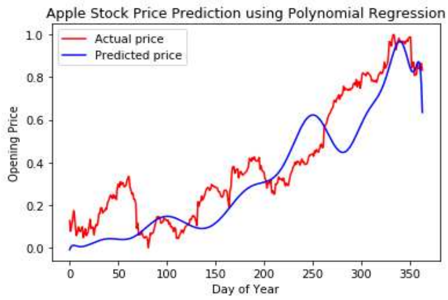
5 Performance Measures

Stock prices for different companies have been predicted through real time analysis of financial data and twitter data. Machine learning models have been designed to predict the stock prices. The efficiency and accuracy of the system have been analyzed through the performance parameters such as Mean Square Error (MSE) [Willmott 1982] and Mean Absolute Error (MAE) [Willmott 1982]. MSE is the mean of the square of all of the error. MAE is a measure of difference between two continuous variables. Accuracy is the agreement between an experimental value, or the average of several determinations of the value, with an accepted or theoretical (true) value for a quantity. Less is the value of MSE or MAE, more is the accuracy of the model.

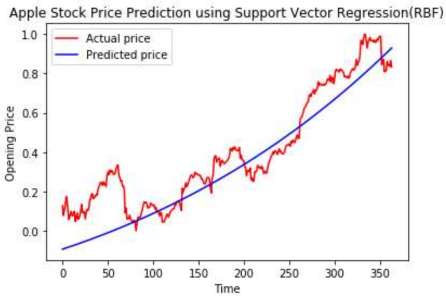
6 Results and Analysis



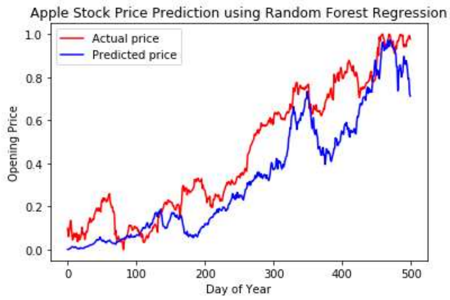
(a) Decision Tree Regression



(b) Polynomial Regression

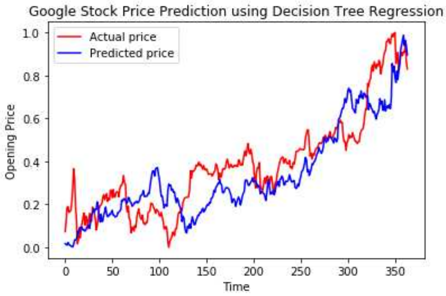


(c) Support Vector Regression

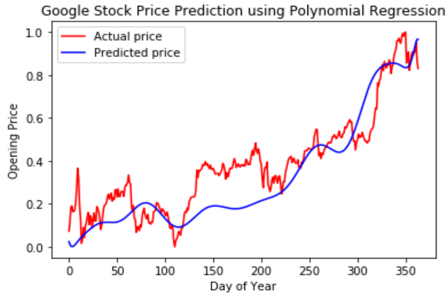


(d) Random Forest Regression

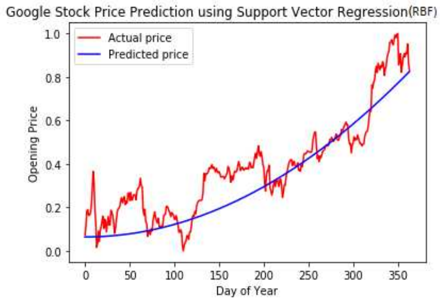
Figure 5: Prediction of Apple's opening price through Finance data



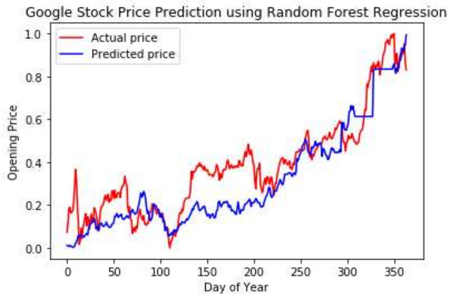
(a) Decision Tree Regression



(b) Polynomial Regression



(c) Support Vector Regression

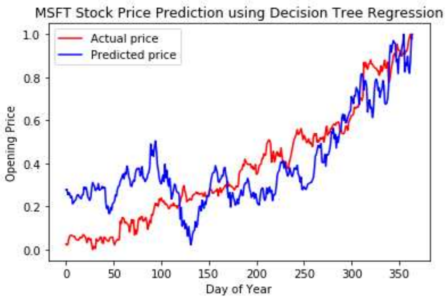


(d) Random Forest Regression

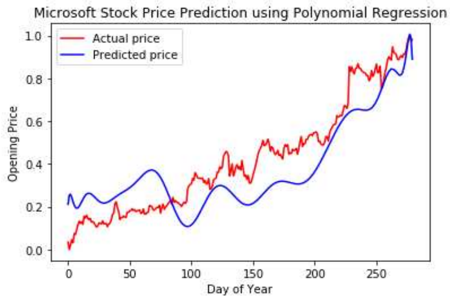
Figure 6: Prediction of Google's opening price through Finance data

Actual stock price of Google, Microsoft and Apple have been collected using Google finance website, which is treated as ground truth for performance measure. The predicted stock price for the testing data is compared with the actual price. Table 3 represent the performance measure values for the different algorithms that have been considered for the first case study, i.e. analysis through finance data for three datasets namely Apple Data, Google data and Microsoft data. It can be observed that support vector regression poses the most useful regression technique that precisely helps in predicting the opening price for the subsequent days. The graphs that have been shown in Figure 5, 6 and 7 as a part of the result also indicate the prediction accuracy of the proposed models. It can be observed that there is a pattern for increase and decrease of the stock value for an entire year. For example, for the current scenario where we have considered the stock prices of the US, the values of the stock prices tend to sore up during the Christmas and thanksgiving time, else maintain the average price.

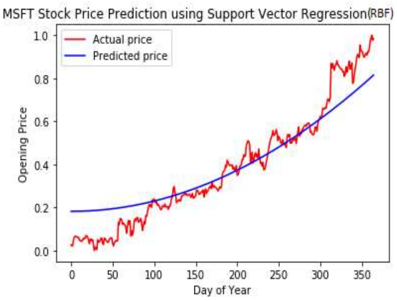
Tables 4 represents the performance measure values for the different algorithms that have been considered for the second case study, i.e., prediction



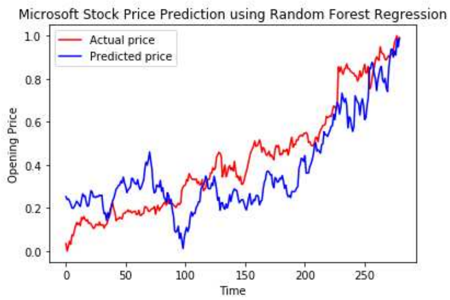
(a) Decision Tree Regression



(b) Polynomial Regression



(c) Support Vector Regression

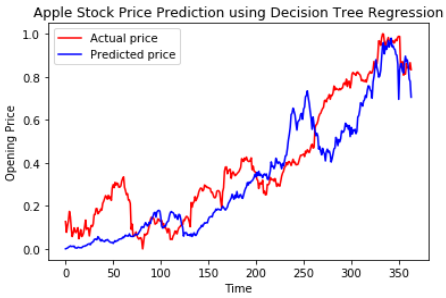


(d) Random Forest Regression

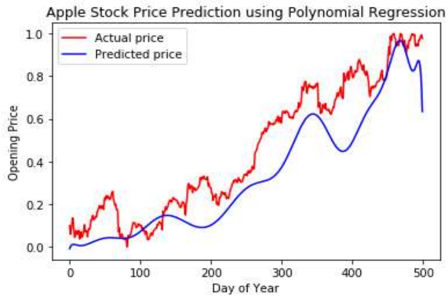
Figure 7: Prediction of Microsoft's opening price through Finance data

Table 3: Performance Analysis of models developed from data collected though financial Websites

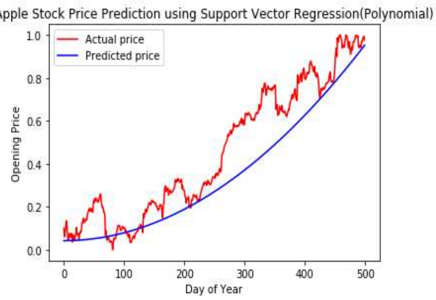
	Apple (Stock Symbol: AAPL)		**Google(Stock Symbol: GOOG)**		**Microsoft (Stock Symbol: MSFT)**	
Classifier Model	MAE	MSE	MAE	MSE	MAE	MSE
Decision Tree	0.104944	0.016948	0.096190	0.014100	0.132543	0.025465
Polynomial Linear Regression	0.112830	0.018638	0.098133	0.01476	0.121925	0.018240
Support Vector Regression	**0.100732**	**0.015288**	**0.08977**	**0.012372**	**0.069381**	**0.008119**
Random Forest Regression	0.117092	0.020056	0.096152	0.014026	0.125516	0.020409



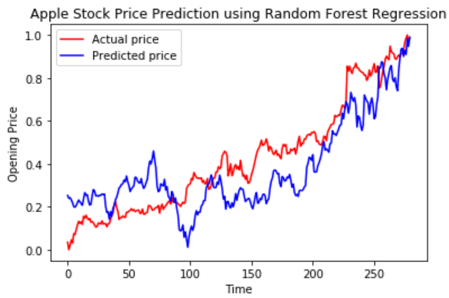
(a) Decision Tree Regression



(b) Polynomial Regression



(c) Support Vector Regression

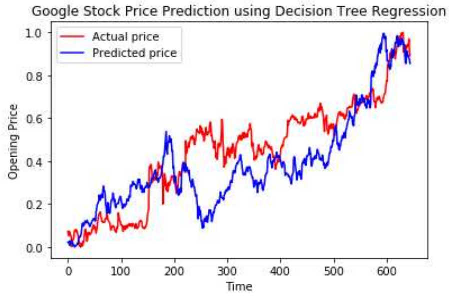


(d) Random Forest Regression

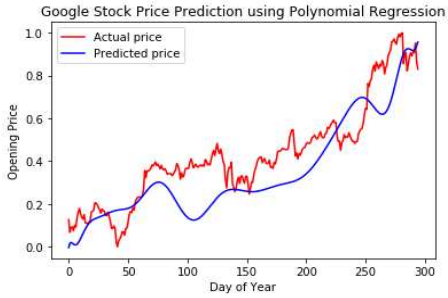
Figure 8: Prediction of Apple's opening price through Twitter Sentiment Analysis

Table 4: Performance Analysis of models developed from data collected though Twitter Streaming

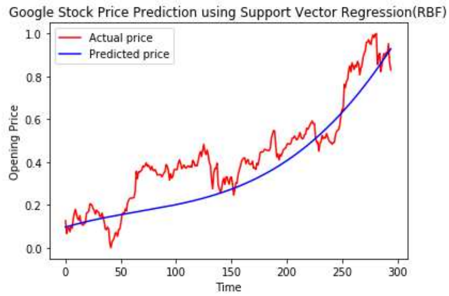
	Apple (Stock Symbol: AAPL)		**Google (Stock Symbol: GOOG)**		**Microsoft (Stock Symbol: MSFT)**	
Classifier Model	MAE	MSE	MAE	MSE	MAE	MSE
Decision Tree	0.117367	0.020158	0.114140	0.018302	0.136649	0.026319
Polynomial Linear Regression	0.123656	0.022604	0.109507	0.018797	0.125346	0.021324
Support Vector Regression	**0.111471**	**0.019033**	**0.096162**	**0.013126**	**0.11140**	**0.016909**
Random Forest Regression	0.124961	0.023175	0.113842	0.018031	0.132289	0.023499



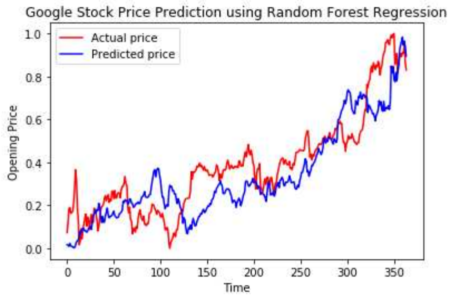
(a) Decision Tree Regression



(b) Polynomial Regression



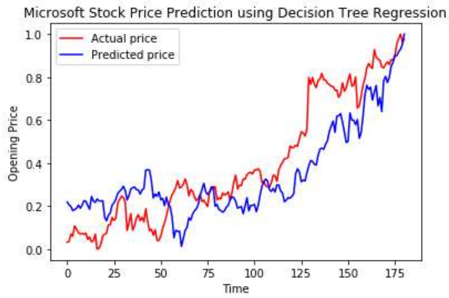
(c) Support Vector Regression



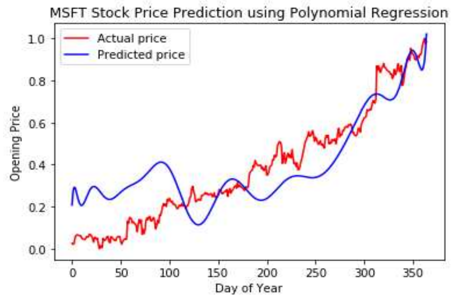
(d) Random Forest Regression

Figure 9: Prediction of Google's opening price through Twitter Sentiment Analysis

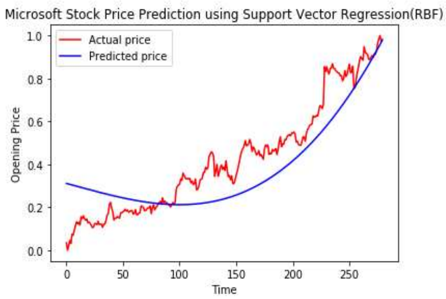
through sentiment analysis from Twitter feeds. The dataset procured contains 560,000 tweets which range over the span of ten years of twitter data starting from May, 2007 till Jun, 2017 and was stored in HDFS file system. The training set contains an equal set of correctly classified as positive and negative reviews about the firm under consideration. Figures 8, 9 and 10 signify that sentiments do have role in predicting the movement of the stock prices. A positive sentiment regarding a particular ticker can sore up the stock price for the subsequent days, similarly, a negative sentiment can bring the prices down. With this experiment, it can be inferred that, there may not be a sudden rise or fall in the prices as in the moment the sentiments are analyzed, but its lagged effect can be seen through the graphs itself.



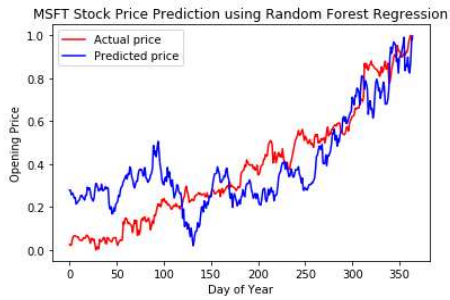
(a) Decision Tree Regression



(b) Polynomial Regression



(c) Support Vector Regression



(d) Random Forest Regression

Figure 10: Prediction of Microsoft's opening price through Twitter Sentiment Analysis

7 Conclusion and Future Work

Predicting future values of stock prices is an challenging task, commonly connected to the analysis of public mood. This study indicates that sentiment analysis of public mood derived from Twitter feeds are helpful to eventually forecast movements of individual stock prices. Moreover, the methodology was adapted to a stream-based setting using the incremental active learning approach, which provides the algorithm with the ability to choose new training data from a data stream for hand-labelling. With this study, stream-based active learning for sentiment analysis of microblogging messages has been introduced in the financial domain. This contributes both to sentiment analysis and the active learning research area, since this issue is still insufficiently explored.

The resultant model discussed as a part of the implementation process is proved to be fault-tolerant since, the RDDs are replicated and check points are introduced through Apache Spark. Scalability and lesser latency is also ensured by designing the model over a distributed architecture. Moreover, feasibility study through batch processing has also been performed with this experiment by taking the aid of various machine learning algorithms such as Random Forest Regression, Support Vector Regression, Decision Tree and Linear Regression, This study further motivates us to dig upon the streaming of online stock data available in various financial websites which may provide us a better model to analyse and predict the future stock prices for a particular company.

Acknowledgement

This research work was supported by Fund for Improvement of S&T Infrastructure in Universities and Higher Educational Institutions (FIST) Scheme under Department of Science and Technology (DST), Govt. of India The authors wish to express their gratitude and heartiest thanks to the department of computer science & engineering, National Institute of Technology, Rourkela, India for providing their research support.

References

[Aizawa 2003] Aizawa, A. (2003). An information-theoretic perspective of tfidf measures. Information Processing & Management, 39(1), 45-65.

[Akita et al. 2016] Akita, R., Yoshihara, A., Matsubara, T., & Uehara, K. (2016, June). Deep learning for stock prediction using numerical and textual information. In 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS) (pp. 1-6). IEEE.

[Altay and Satman 2005] Altay, E., & Satman, M. H. (2005). Stock market forecasting: artificial neural network and linear regression comparison in an emerging market. Journal of Financial Management & Analysis, 18(2), 18.

[Behera et al.] Behera, R. K., Das, S., Jena, M., Rath, S. K., & Sahoo, B. (2017, December). A comparative study of distributed tools for analyzing streaming data. In 2017 International Conference on Information Technology (ICIT) (pp. 79-84). IEEE.

[Bifet et al. 2015] Bifet, A., Maniu, S., Qian, J., Tian, G., He, C., & Fan, W. (2015, November). Streamdm: Advanced data mining in spark streaming. In 2015 IEEE International Conference on Data Mining Workshop (ICDMW) (pp. 1608-1611). IEEE.

[Bollen et al. 2011] Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. Journal of computational science, 2(1), 1-8.

[Buchmann and Koldehofe 2009] Buchmann, A., & Koldehofe, B. (2009). Complex event processing. it-Information Technology Methoden und innovative Anwendungen der Informatik und Informationstechnik.

[Cantelin et al. 2014] Cantelon, M., Harter, M., Holowaychuk, T. J., & Rajlich, N. (2014). Node. js in Action (pp. 17-20). Greenwich: Manning.

[Chen and Lazer 2013] Chen, R., & Lazer, M. (2013). Sentiment analysis of twitter feeds for the prediction of stock market movement. stanford edu Retrieved January, 25, 2013.

[Chong et al. 2017] Chong, E., Han, C., & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. Expert Systems with Applications, 83, 187-205.

[Cutler et al. 1988] Cutler, D. M., Poterba, J. M., & Summers, L. H. (1988). What moves stock prices? (No. w2538). National Bureau of Economic Research.

[Ding et al. 2015] Ding, X., Zhang, Y., Liu, T., & Duan, J. (2015, June). Deep learning for event-driven stock prediction. In Twenty-fourth international joint conference on artificial intelligence.

[Donaldson and Kim 1993] Donaldson, R. G., & Kim, H. Y. (1993). Price barriers in the Dow Jones industrial average. Journal of financial and Quantitative Analysis, 313-330.

[Henrique et al. 2018] Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2018). Stock price prediction using support vector regression on daily and up to the minute prices. The Journal of finance and data science, 4(3), 183-201.

[Hoffman 2013] Hoffman, S. (2013). Apache Flume: distributed log collection for Hadoop. Packt Publishing Ltd.

[Liaw and Wiener 2002] Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R news, 2(3), 18-22.

[Marz and Warren 2015] Marz, N., & Warren, J. (2015). Big Data: Principles and best practices of scalable real-time data systems. New York; Manning Publications Co.

[Meng et al. 2016] Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., ... & Xin, D. (2016). Mllib: Machine learning in apache spark. The Journal of Machine Learning Research, 17(1), 1235-1241.

[Mittal and Goel 2012] Mittal, A., & Goel, A. (2012). Stock prediction using twitter sentiment analysis. Standford University, CS229 (2011 http://cs229. stanford. edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis. pdf), 15.

[Morck et al. 2000] Morck, R., Yeung, B., & Yu, W. (2000). The information content of stock markets: why do emerging markets have synchronous stock price movements?. Journal of financial economics, 58(1-2), 215-260.

[Nguten et al. 2015] Nguyen, T. H., Shirai, K., & Velcin, J. (2015). Sentiment analysis on social media for stock movement prediction. Expert Systems with Applications, 42(24), 9603-9611.

[Oliveira et al. 2017] Oliveira, N., Cortez, P., & Areal, N. (2017). The impact of microblogging data for stock market prediction: Using Twitter to predict returns, volatility, trading volume and survey sentiment indices. Expert Systems with Applications, 73, 125-144.

[Qian and Rasheed 2007] Qian, B., & Rasheed, K. (2007). Stock market prediction with multiple classifiers. Applied Intelligence, 26(1), 25-33.

[Si et al. 2013] Si, J., Mukherjee, A., Liu, B., Li, Q., Li, H., & Deng, X. (2013, August). Exploiting topic based twitter sentiment for stock prediction. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 24-29).

[Sprenger et al. 2014] Sprenger, T. O., Tumasjan, A., Sandner, P. G., & Welpe, I. M. (2014). Tweets and trades: The information content of stock microblogs. European Financial Management, 20(5), 926-957.

[Thein 2014] Thein, K. M. M. (2014). Apache kafka: Next generation distributed messaging system. International Journal of Scientific Engineering and Technology Research, 3(47), 9478-9483.

[Willmott 1982] Willmott, C. J. (1982). Some comments on the evaluation of model performance. Bulletin of the American Meteorological Society, 63(11), 1309-1313.

[Yang et al. 2002] Yang, H., Chan, L., & King, I. (2002, August). Support vector machine regression for volatile stock market prediction. In International Conference on Intelligent Data Engineering and Automated Learning (pp. 391-396). Springer, Berlin, Heidelberg.