International Journal on Semantic Web and Information Systems
Volume 15 • Issue 2 • April-June 2019

A Real-Time Twitter Trend Analysis and Visualization Framework

Jamuna S Murthy, PES University, Bengaluru, India
Siddesh G.M., Ramaiah Institute of Technology, Bengaluru, India
Srinivasa K.G., National Institute of Technical Teachers' Training & Research, Chandigarh, India

ABSTRACT

Trend analysis over Twitter offers organizations a fast and effective way of predicting the future trends. In the recent years, a wide range of indicators and methods were used for predicting the trend on Twitter with varying results, unfortunately most of the research focused only on the emerging trends which has gained long-term attention on the Twitter platform. This article depicts trend variations, i.e. to predict whether the trend on Twitter will gain attention or not in the next few hours. Hence a novel method called: "Twitter Trend Momentum (TTM)" is introduced for trend prediction which is the enhancement of a well-known stock market indicator called moving average convergence divergence (MACD). Reason analysis for trend variation is also carried out as an extension to the authors' research work. An evaluation of the framework showed the best results which are applied to build a real-time web application called "TwitTrend." The application acts as a real-time update and recommendation system of top trends to users.

KEYWORDS

Apache Storm, Big Data Analytics, Kafka, MongoDB, Scalability, Throughput, Trend Analysis

INTRODUCTION

Twitter is considered as one of the world's largest social networking sites which allow users to customize their public profile, connect with others and interact with connected users. A global survey from one of the well-known companies like "Statista" witness that, as of the fourth quarter of 2017 there are average number of 330 million monthly active twitter users (Statista, 2017). Users post short messages of 140 characters called tweets based on variety of topics ranging from simple ones such as "Hi #xyz☺am@college" to themes such as "#IPL-2017" using different ways such as blogging on Twitter website, using Twitter mobile application and also through other social network applications which allow virtual connections from one application to another such as Instagram. Being a most popular social networking sites twitter uses a network model called "following". Any person can follow any other person who remains to be his/her friend. The person who follows is named as a follower and a follower on twitter can receive all the updates that he/she follows. Tweets are of different types which include normal tweet, reply and retweet. Normal tweets are the one which people post based on their thoughts and opinions which is not a reply to any others tweets or a retweet for concerned tweet. Reply is a tweet which a twitter user post with "@" symbol attached with name of replies' for example "@abc am not interested in you". Retweet is a message which commonly starts with "RT" and it is a post which is shared by the follower to his/her followers. Apart from this the other very important

DOI: 10.4018/IJSWIS.2019040101

features of tweet are HashTags and URLs which make twitter stream more readable and provides an understanding of current topic or an event discussed on twitter (Lau, Collier & Baldwin, 2012).

Twitter has outcome as one of the best platforms of news information propagation in present days due to enormous number of users' active on daily basis and also, it's very spontaneous nature of real-time tweeting that reach out to people within few seconds. It may be any kind of news such as, Prime Minister's decisions of changes in autonomy or company's date of releasing their product, the "following" feature in twitter makes the news to spread around within short interval of time. Nearly there are thousands of topics discussed on twitter daily but most of the users are interested in only the hot topics or top news which is going viral around the world and will remain as trend in next few hours. Due to enormous number of tweets on twitter it is impossible for us to browse the top news topic or trends. Hence there is a need for a real-time trend analysis system which can analyse and predict the top trends which are going viral around the world and will remain to be trend in next few hours or in next few days. The existing systems for trend analysis used traditional techniques such as LDA topic Modelling, TF-IDF algorithm etc for predicting top trends on twitter. But these methods are less accurate, and the results were satisfactory. Thus, for analysing and predicting top trend on twitter a real-time twitter trend analysis and visualization framework is introduced in this research work by implementing a novel method called "Twitter Trend Momentum (TTM)". A deep reason analysis for trend variations is carried out as a major part of this research work and finally the framework is applied to build an interactive web application called "TwitTrend" which acts a real-time update and recommendation system for trend detection and analysis.

Motivation

The main motive of the proposed work is to analyse the Variation of trends on twitter. Based on our analysis of functioning of twitter application we propose three main reasons for changing trends on twitter, they are as give below.

Influential Users

Whenever a topic become hotter on twitter, then it means there is an influential user involved in the discussion i.e. if there are large numbers of tweets on entertainment, then it means some celebrity is involved in the discussion of a topic. Therefore, if we monitor the influential user involved in the topic of discussion then we can easily track the trend of topics over real-time. Chi-Square test is used to identify the influential users. Let T, be the time at which the news topic "TP" becomes a trend. The statistics for the above is given by Table 1.

Case1: P is the twitter user ID where She/he interacts to the topic TP at time T.
Case2: Q is the twitter user ID where She/he interacts the topic TP at time T.
Case3: R is the twitter user ID where She/he doesn't interact the topic TP at time T.
Case4: S is the twitter user ID where She/he doesn't interact the topic TP and also won't tweet at time T.

Then the Correlation of Influential Users "u" with news topic TP at time T is given in the mathematical form as given in Equation (1):

Table 1. Statistical table for identifying influential users

	TP	**TP**
T	P	Q
T	R	S

$$\mathrm{Cu}\left(\mathrm{TP},\ \mathrm{T}\right) = \frac{\left(P+Q+R+S\right)\left(PS-QR\right)^{2}}{\left(P+Q\right)\left(R+S\right)\left(P+R\right)\left(Q+S\right)} \tag{1}$$

Finally the users are ranked using the value obtained from Cu(TP, T) and the top users are selected.

Keywords

Another reason why a topic on twitter becomes a trend is it may be involved in discussion with similar topic or event connected to it. If we monitor the keywords relative to the topics or events then we can find out the trend variations by monitoring the most related keywords to the topic which is becoming hotter on twitter. First, we use Chi-Square test to find the Keywords. Let T, be the time sliced window at which the news topic TP becomes hotter. Then the Correlation of Keywords "w" with news topic TP at time T is give in the mathematical form as given in Equation (2):

$$\mathrm{Cw}\left(\mathrm{TP},\mathrm{T}\right) = \frac{\left(P'+Q'+R'+S'\right)\left(P'S'-Q'R'\right)}{\left(P'+Q'\right)\left(R'+S'\right)\left(P'+S'\right)\left(Q'+S'\right)} \tag{2}$$

Finally the Keywords are ranked using the value obtained from Cu(TP, T) and the top Keywords are selected.

Interaction of Topic

Another reason for trend variation is the rate at which the users discuss about the topic. When there is a falling of news topic it means only few users are discussing it. Whereas when there are more tweets on twitter then it means the topic is gaining more attention. Generally, users concentrate on the rising topic or topic which is gaining more attention and usually neglect the topics which are becoming cold. Thus, it becomes a situation for trend falling on twitter.

Contributions of Proposed Work

- The proposed framework includes four main components in terms of Data Collection, Data Parsing, Data Analytics-Prediction and Data Visualization implemented using Apache storm programming module.
- A novel method is deployed for natural language processing of tweets which is called a "Pre-Processing" algorithm which is explained in detail in "Proposed Work" section.
- TTM method is novel method used for real-time trend predictions of topics on twitter represented as keywords. It is an enhanced version of a well-known stock market indicator called MACD, which was created by Gerald Apple. MACD indicator provides many rules and guidelines which are used for future prediction of trends.
- The results of analysis are visualized on the real-time dashboard with the help of Data Visualization module built as a web framework called "TwitTrend" using java library D3 and python micro server Flask.
- As an extension to the proposed framework we also propose some reasons for trend variations which are mentioned in the "Motivation" subsection.

BACKGROUND

Twitter is a real-time microblog which contains up to 140 characters. The tweets can contain images or videos as well as the most significant feature are its instantaneity. Additionally, Twitter contents are limited by the number of characters restriction; most tweets contain around 14 words and uses URL or hashtags to express the main idea. The hashtags indicate the keywords of tweets, however most hashtags are abbreviations (Weller et al., 2016). The data from the twitter can be fetched using public Twitter APIs available. The two most prominent Twitter APIs are Twitter4j a java library (Yamamoto et al., 2010) and Tweepy a python library (Roesslein, 2015). Both are considerably high at throughput rate but in our framework we deploy Tweepy which is most efficient in collecting geographical information and also helps in conversion of JSON tweets to normal text directly (Hawker, 2010).

Text mining is "the process of extracting interesting and non-trivial patterns or knowledge from unstructured text documents". The basic concepts of text mining involve tokenization, n-gram, part-of-speech (POS), stop word, and Term Frequency Inverse Document Frequency (TF-IDF) (Extracting Information from Text, 2017).

- **Tokenization:** Tokenization transmits sentence to words.
- **N-gram:** "an n-gram is a contiguous sequence of n items from a given sequence of text or speech"
- **POS:** Part-of-speech tags the speech of word in the sentence. In most cases, noun is more important than other speeches.
- **Stop words:** Stop words are a list of meaningless words. Normally, NLP will remove stop words from the text to increase the accuracy of classification or clustering.
- **TF-IDF:** TF-IDF (Tf–idf, 2017) is "a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus". There are several weighing schemes to calculate the IDF score. The document frequency smooth scheme is given by: $(t,)*\log D/1+df(t)$. Applied in the calculation, tf(w,d) means the times word w occurs in document d, idf = log(n / df(w,D)), and df means the number of document which contains word w (Zhang et al., 2011) .

The combination of all these techniques forms an efficient natural language processing (NLP) pipeline for extracting the information from tweets. Hence in our framework we propose a new pre-processing algorithm for text mining of twitter data deployed using Stanford NLP techniques (Manning et al., 2014).

In distributed computing Hadoop is the most popular framework which addresses batch data. However, the proposed framework aims at processing the real-time stream data which means the data flows continuously and unboundedly. Additionally, the latency requirements of the stream data are stricter than batch data. Apache Storm is considered as number one real-time distributed stream processing engine for its best scalability, flexibility and multi-programming capability (Ankit Jain, 2014). Thus, proposed system implements Data Parsing Module with Apache Storm. The programming model of Apache Storm is called spout and bolts in which a spout is any data source of topology used to collect data (i.e. a messaging queue) and bolt is a processing unit which is irreversible. Spout and bolts are together combined to form a topology using stream grouping mechanism. Kafka is a distributed message queuing system with highest throughput rate, low runtime overhead and persist messages with complexity of O (1). According to the benchmark, the throughput of a single Kafka producer or consumer can reach around 1 million messages per second. Hence proposed system implements Kafka spout to fetch the tweets from twitter over real-time (Garg, 2013).

Trend Analysis or Prediction of future trends is emerging area of research today. It mainly deals with aspects of technical analysis that tries to predict future movement of that particular trend identified. Since there are varieties of topics discussed on twitter, it has outcome as one of the best platforms for trend analysis. Commonly most of the trend detection system deals with emerging trends. Emerging trends on twitter is nothing but the topic which is going viral and has grabbed most of attention over

the times at past. For example, android phones such as "Samsung Galaxy" was trend in 2010's and remained for quite long time but due to the emergence of "Xiomi Redmi" phones in 2014 the market of Samsung products has gradually reduced.

In the recent times LDA-Style Topic Modelling has become the wide area of research in the field of Machine Learning and Information Retrieval (Zhao et al., 2012). "Twitter Trend detection (TTD)" is a system" developed (James et al., 2013) for detecting emerging trends which included LDA topic modelling. It mainly focused on event datasets collected from twitter. "Trend Analysis Model (TAM)" mainly focused on temporal words and other words from the datasets for (Salas-Zarate et al., 2016) analysing the emerging trends over time. But all these models discussed only about evolution of trends but proposed system implements a novel method for trend variations. That is proposed system predicts topics over real-time which will become a hot topic in next few hours or loses attention.

Rashid Kamal et al., in 2017 designed a "Real-time Opinion Mining of Twitter Data using Spring XD and Hadoop". The framework mainly focused on deciding the trends based on HashTag analysis and visualization. But the main drawback of the framework is it did not include the URLs for trend analysis which is also a major part of deciding trends on twitter. Also, the framework included Hadoop which is least suitable for real-time processing in terms of speed. But the proposed system proves to be most efficient at present times since it uses Apache storm programming module which is well suited for streamed data analysis and also the proposed TTM method is most accurate form of option mining of twitter data since it also includes URLs analysis.

There are many statistical methods for trend predictions and the one used by most of the researchers is TD-IDF (Term Frequency Inverse Document Frequency), which involves document frequency smooth scheme. But from the studies it is evidenced that stock market indicators are more mature and are accurate. The most prominent ones include Exponential Moving Average (EMA) (Lu et al., 2012), MACD (Durantin et al., 2014), Rate of Change (ROC) etc. The proposed system considers MACD and EMA to define a new concept called TTM by analysing the keywords from news topics. First, some keywords of topic are monitored and pre-processed to compute two moving average of time slices for them. Later TTM of the news topics is defined by subtracting Longest Moving Average (LMA) with Shortest Moving Average (SMA) by giving an x-factor for LMA called "α". Later we use Moving average to smooth it to find final value. Thus, final TTM equation obtained is used for predicting trends of news topics for future which is most accurate among all existing techniques.

Twitter is a real-time application and hence twitter data flows in the forms of streams every second. Usually the data streams are unstructured and arrive so swiftly that they are infeasible to store and operate. MongoDB (Kanade et al., 2014), a NoSQL database is used for dealing with stream data efficiently because it stores the data in JOSN file with no loss. Hence proposed System selects MongoDB database to complement for real-time visualization of the trend analysis results which is an essential part of proposed analytical framework.

D3.js (Data Driven Documents) (Bostock et al., 2011) is a java script library which can be used for building dynamic web pages. Also, micro-server flask which is a python web framework library helps in programming of web pages. Thus, proposed system selects D3.js and Flask server (Grinberg, 2014) for building dynamic web application of top 10 trends which is visualized as real-time dashboard.

PROPOSED WORK

The proposed framework for real-time twitter trend analysis and visualization is shown in Figure 1. It includes four major modules in terms of Data Collection, Data Parsing, Data Analytics and Data Visualization. Data Collection module fetches real-time tweets using the twitter data source with the help of twitter streaming API. The fetched tweets are passed through the message queue to reach the Data Parsing module. The Data Parsing module implements pre-processing algorithm extract the keyword from tweets for trend analysis. The keywords are detected and analysed using Data Analytics and Prediction module which predicts top 10 trends using novel TTM algorithm. The

Table 2. Message string format

Tweet Content	Screen Name	Created Time	Geographical Info (long, lati)	Country Code (ISO-alpha)	TTM Value

analysed results are stored in NoSQL database and are displayed on Visualization Dashboard using Data Visualization module.

Data Collection

The main aim of this module is to fetch the user timeline tweets randomly based on trending with the help of twitter streaming API and produce messages to Kafka broker instantly so that they can be injected into the Data Parsing module for obtaining correct form of data. With twitter API developers can access 1% of public tweets. Twitter also formulates rate limits to restrict handling of APIs. Twitter APIs utilize 15minute sliding window whether an application exceeds rate limits. The system involves Twitter API implementations of python version which is called Tweepy. Twitter status is the basic entity of Twitter message object, the system extracts five fields from it, namely tweet content which may contain URL, hashtags, mentioned user, whether it is retweeted, text, and emoji, screen name which is the account ID of user and is unique, created time which indicates when this message is made, geographic information including latitude and longitude, country code which is formatted in ISO alpha-2. The reason why we do not use country name is that sometimes country name may change by location, for example, the country name of Japan may become Japanese characters. The general message string format for our system is shown in Table 2. The message string consists of five fields extracted as mentioned before which are mandatory. Later the values are added sequentially from fifth field with a delimiter " |" based on our needs and here we add TTM value as sixth field. If any value is missing in the message string it is represented as "n/a."

Data Parsing

Twitter data is up-to-date and informative; however, it includes lots of fragments and noise. Before processing Twitter data has to be pre-processed. The proposed "pre-processing algorithm" is defined as shown below:

Step1: Analyzing the trends of Twitter account by fetching its timeline through the Twitter API.

Step2: Fetching the content of URLs which are shared by users: Twitter's contents are limited to 140 characters, most tweets can hardly describe entire viewpoints. Instead of describing ideas, users post tweets with URLs and HashTags. URLs content describe what users want to share in detail and HashTags are keywords of the tweet which are recognized by users.

Step3: Tokenizing the contents of URLs. The URLs are in the form of regular expressions which are tokenized using the Stanford Natural language processing library.

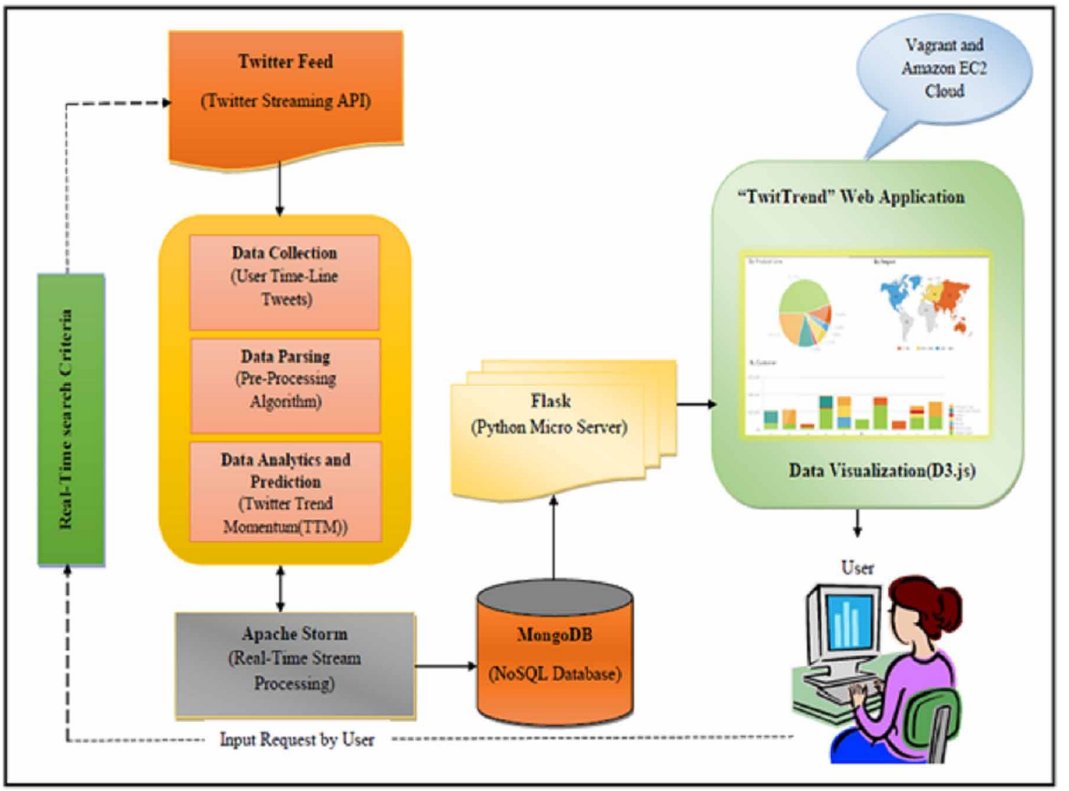
Step4: Filtering out stop-words. Besides the normal stop words, Twitter text may contain some special stop words, such as "http", "RT", usernames.

Step5: Tagging the words left with their part of speech. In trend analysis, the noun is more important than the adjective and verb. In order to improve accuracy, the algorithm also involves the concepts of unigrams, bigrams and trigrams. The combinations of speech tag are listed below.

1. Unigrams: noun whose length is more than three.
2. Bigrams: noun + noun.
3. Trigrams: noun + noun + noun, noun + conjunction + noun.

Step6: Stemming, lemmatization and normalization is carried out to remove words affix and transform all different variations of words to the canonical form.

Figure 1. Real-time twitter trend analysis and visualization framework



Data Analytics-Prediction

As discussed above Proposed TTM algorithm is combination of two different moving averages defined using well known stock market indicator MACD. Therefore, before giving the definition of TTM Moving Average is defined as a baseline for the algorithm. Over a real-time "T" of continuous time interval, partition the total time into equal parts T_i. For each partition T_i, the occurrence of the keywords collected using Data Parsing Module is given by $G(T_i)$ and the size of partition time is given by p. The m^{th} time sliced Moving Average is given by MA (m, p) as shown in Equation (3).

$$\text{MA (m, p)} = \frac{\sum_{i=m-p+1}^{n} G\left(Ti\right)}{p} \tag{3}$$

If m<p, then the MA (m, p) is redefined as shown in Equation (4):

$$\text{MA (m, p)} = \frac{\sum_{i=1}^{n} G\left(Ti\right)}{m} \tag{4}$$

Moving Averages are the best indicators for tracking trending topics on twitter. Also, the different time partitioned moving averages can track the trends of different time periods. As defined by the original MACD indicator, the TTM is defined as shown in Equation (5):

$$TTM(n) = MA(m, p_s) - MA(m, p_l) \quad (5)$$

"p_s" is Shortest Moving Average (SMA), while "p_l" is Longest Moving Average (LMA).

The TTM simply defined as the value obtained by subtracting SMA with LMA. But when it is concerned to trending topics on twitter it varies to stock market indicator MACD with few characteristics. When the LMA of a particular topic on twitter given by a keyword is high for at current time, then that topic remains for longer period. Whereas when it is concerned to stock market, if the stock price is high at current time, then it is initial stage of falling. So TTM as enhanced version of MACD in proposed system is defined as shown in Equation (6):

$$TTM(m) = MA(m, p_s) - (MA(, p_l))^{\alpha} \quad (6)$$

Where $0 < \alpha < 1$.

As defined above TTM is enhanced version of well-known stock market indicator MACD which differs in two different characterises. Firstly, we are using Moving Average (MA) in proposed system instead of EMA. This is because the topic frequency (i.e. Keywords) is very volatile in nature at varied time partitions. Since EMA is concerned to frequency of current time partition, it makes the keyword even more volatile than before. Thus, using MA makes keywords to remain non-volatile. Secondly, an x-factor is added to the LMA called "α". This is because, when it is concerned to the topics on twitter we prefer hot topics should be given more attention even though it has similar TTM as falling topic. The hot topics are usually LMA's in nature. Therefore, giving x-factor "α" to LMA highlights the hot topic on twitter which becomes a trending topic. However, the TTM value calculated using Equation (6) is more volatile. Therefore, we use MA for smoothing as shown in Equation (7):

$$Mom(m) = MA(TTM(m), p) \quad (7)$$

Different guidelines and rules are defined by MACD indicator for predicting the future stock market price. But in the proposed system the simplest one is used i.e. whenever the value of TTM changes from Negative to Positive, the then it is a raising edge of the current topic on twitter. Contrary, whenever the value of TTM changes from Positive from Negative then it means the topic is falling down or dying. Trend analysis is implemented using the Apache Storm programming module. The following are details of its bolts, input and output:

Trend Analysis Topology

Input: The inputs of this topology are basic Twitter messages and Location based information file.
Output: The outputs are key-value pairs.
Document Fetch Bolt: The document fetch bolt extracts web page contents through URL within tweets.
Tokenize Bolt: The tokenize bolt converts the document contents to words.
SWR Bolt: This bolt removes the skip words. Removing skip words indirectly increases the accuracy.
POS Bolt: This bold does tagging of words left with their part of speech. The nouns are converted to unigrams, bigrams and trigrams.
SLN Bolt: This bolt does Stemming, lemmatization and normalization to remove words affix.

TTM Bolt: This bolt implements the proposed TTM algorithm and calculate the Momentum values for each word.

Intermediate Bolt: This bolt ranks the values with from highest to lowest.

Total Ranking Bolt: The total ranking bolt converges all results of Intermediate bolt and sends results to a Kafka producer.

The data from this module is stored in MongoDB database. MongoDB is a NoSQL database and stores the results in JOSN file format with historical data of time slices which is further utilized for visualizing the data on TwitTrend web application.

Data Visualization

Data visualization components are implemented by D3.js and a web framework Flask. Proposed framework implements an interactive web application for the users called "TwitTrend". D3.js allows users to build visualization by themselves. The data of the web framework is sent from the server continuously. To address real-time data, there are four options: polling, long polling, WebSocket and Server-Sent Event. Polling and long polling are techniques that the client side sends requests to the server side periodically. They are the easiest to implement, even though they are costly. WebSocket and Server-Sent Event (SSE) are more popular, in which WebSocket allows both clients and servers to send messages to each other, while SSE, as its name shows, is only responsible for sending messages from the server side to the client side. For only the server side to send data to the front-end periodically, the proposed system exploits SSE to update stream data, which only need to set Content-Type to text/event-stream in HTTP header. In order to update data on webpage, the website checks whether data updated every second with window.setInterval function. The user interface for "TwitTrend" web application is shown in Figure 2. It displays the top ten trends predicted for the user based on location search. This is very useful for the users who wish to know the top trending news with positive or negative highlights from any region. Red coloured keyword indicates the negative word with SMA and green coloured keyword indicated the word with LMA. After some time the keyword shown in red colour in the figure disappears with a falling edge.

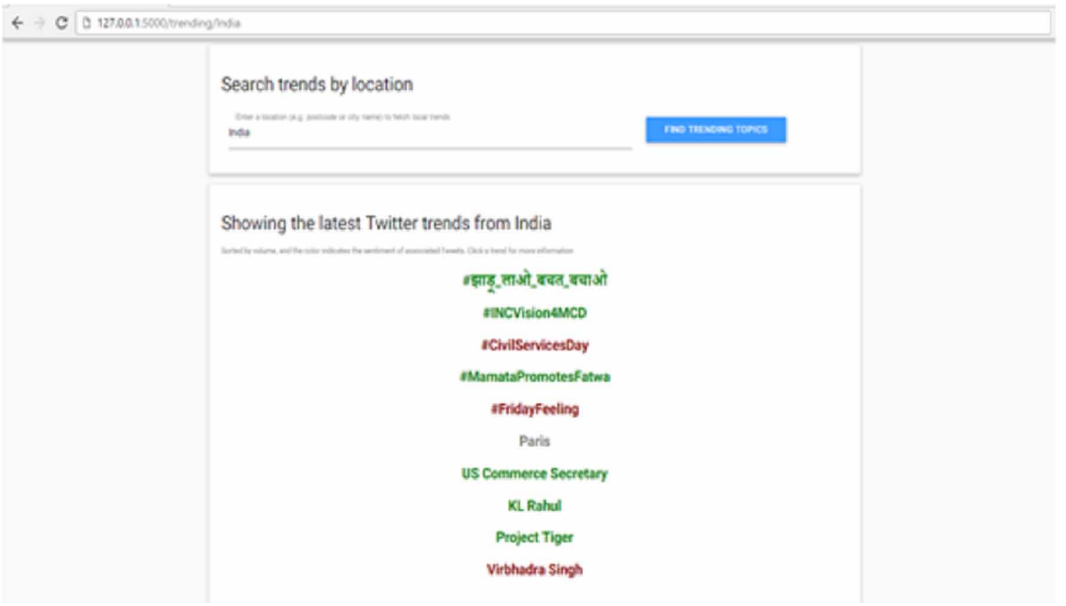
EVALUATION RESULTS

The system is evaluated in three cases first by validating novel TTM prediction method, second by evaluating the proposed framework for throughput and scalability results and third by validating the proposed reasons for trend variations. All experiments are deployed with local virtual machines or cloud computing platforms. To run the system locally, we apply Vagrant (Staubitz et al., 2016) and Virtual Box to host virtual machines of Ubuntu 14.04. In the cloud, we chose AWS EC2 (Portella et al., 2017)) as the cloud computing platform to build the cluster of Ubuntu machines.

Validation of TTM algorithm

TTM method is validated by considering two datasets over real-time. First is "News-Event" dataset, which contains news topics. Here we consider each news event as particular topic. The second dataset is the "Twitter Trends" datasets which can be crawled obtained using the JOSN file format from MongoDB database of our framework. First, the top headlines were crawled from any press website. Then the related tweets to the headlines were fetched using Data Collection module of our proposed framework using "query string" filter. This dataset was named as "News-Event" dataset and it contained 2000 headlines and 350 thousand tweets. Secondly the "Twitter Trends" dataset was collected using the Data Collection module of the framework with the help of Twitter Streaming API. The dataset contained 1million tweets with 1154 trending topics. By analysing the two datasets we incurred there are two main factors which should be considered. One is, not many users are active

Figure 2. "TwitTrend" web application



on twitter. The Figure 3 shows the graph of distribution of tweets for different number of key users. The other factor is when a new event contains larger number of tweets it has longer life span. The Figure 4 shows the graph with life span of different tweets.

Case 1: "News-Event" Dataset

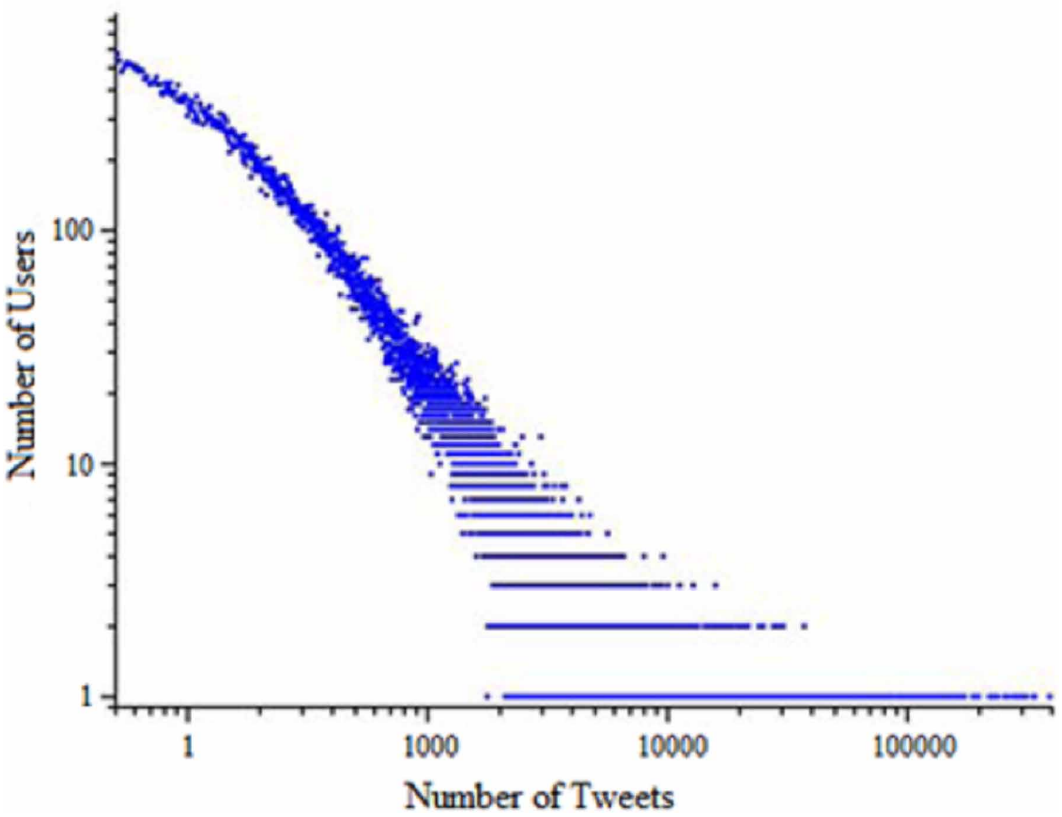
A news headline called "Indian companies and government institutions hit by massive global cyber-attack", which was released by Associated Press on May 13, 2017, 08.38 PM IST is considered for validating the TTM method as shown in Figure 5. After 80 minutes of the appearance of the news on twitter, the TTM changes from negative to positive. Later there is a significant increase in the time line. After about 225 minutes, the value of TTM changes from pos to neg. Later on after that point, not many discussed about the topic.

Case 2: "Twitter Trends" Dataset

A keyword called "Redmi" is considered from the second dataset for validation of TTM method. Redmi becomes a twitter trend on 12th May 2017 at 12:00. The results of TTM are shown in Figure 6. At the 35th hour, the value of TTM changes from negative to positive. Based on the values calculated from proposed TTM method we predict rising point of the twitter trend. It showed LMA as expected. Later after 12 hours, the twitter chose "Redmi" as a hot topic but this topic had already showed falling edge based on our analysis. Thus, we can observe that there is a large difference between the Trend Analysis method of twitter and proposed TTM method. It clearly indicated twitter method showed missing prediction.

The above two datasets were considered to show how our proposed TTM method can be used to predict the trends on twitter. Next, we present quantitative method of evaluating proposed Trend Analysis method. Whenever a news topic is getting a lot of attention on twitter then it will have a concise discussion point of time. This is a point where the density of tweets which talk about this topic is higher. From the "News-Events" dataset as mentioned previously we considered the life span of the tweets and decided the highest point as the main peak using TTM method. This is called as

Figure 3. Number of users vs number of tweets



the Prediction point. Proposed TTM method is compared against the Fixed Threshold (FT) method to prove that our method provided best results. In FT method there is s threshold value which is already fixed and defined. The tweets were considered to calculate the FT at each point of time interval. Wherever the tweets density exceeds the threshold, it means the main peak is arriving. The comparison results are shown in Table 3.

Whenever the prediction point remains far from main peak then it lead to missing predictions. The overall rate of missing prediction is called Error rate. From Table 3, it is clear that proposed TTM method has lowest error rate. From the "Twitter Trends" dataset we have calculated the rate of trending topics which has feature of changing from positive to negative. This is tabulated in Table 4. In the last 16hours, before a topic becomes a trend on twitter it has experience 75.23% of changing rate from positive to negative.

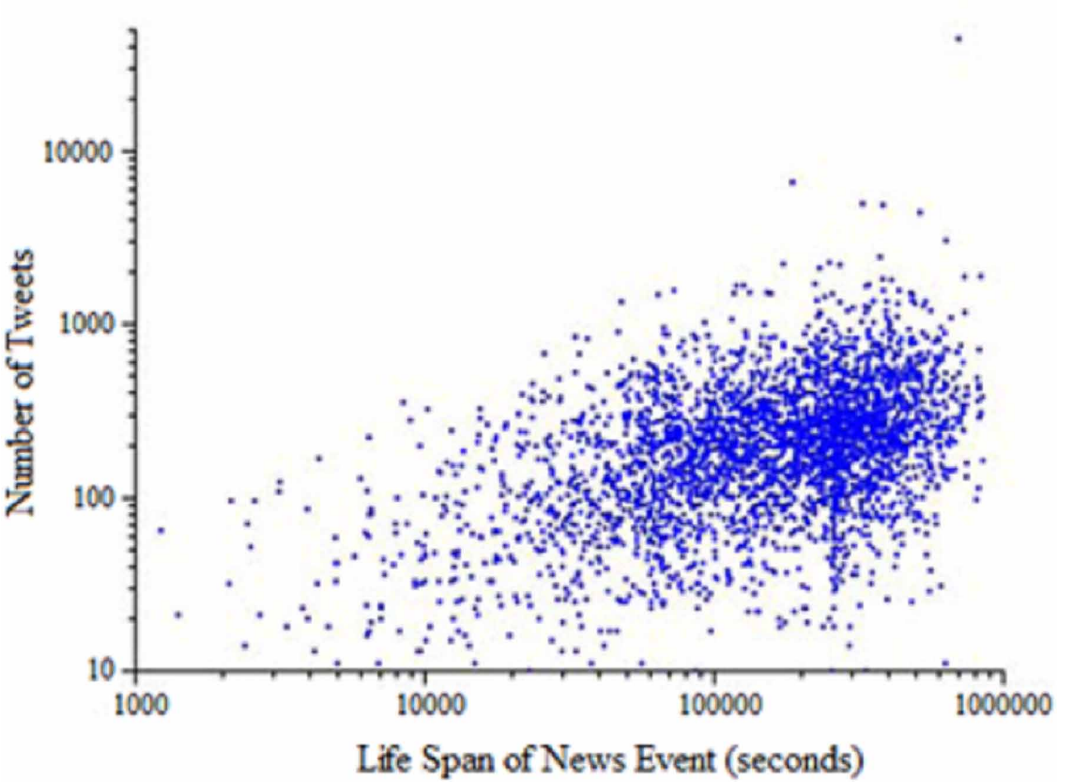
Evaluation of Proposed Framework

The proposed framework is evaluated using two metrics Throughput and Scalability.

Evaluation of Throughput

The throughput for the application is evaluated as the rate of twitter stream i.e. the number of messages received from twitter based on query string filter over multi-node Storm cluster with 2vCPUs. As mentioned before, all data collection components integrate java library twitter4j with Kafka, thus the proposed system handles a simple Kafka consumer to count the number of messages using the Kafka Manager tool which displays the message count on the UI. From Figure 7 we observe that the

Figure 4. Number of tweets vs life span of news event (seconds)

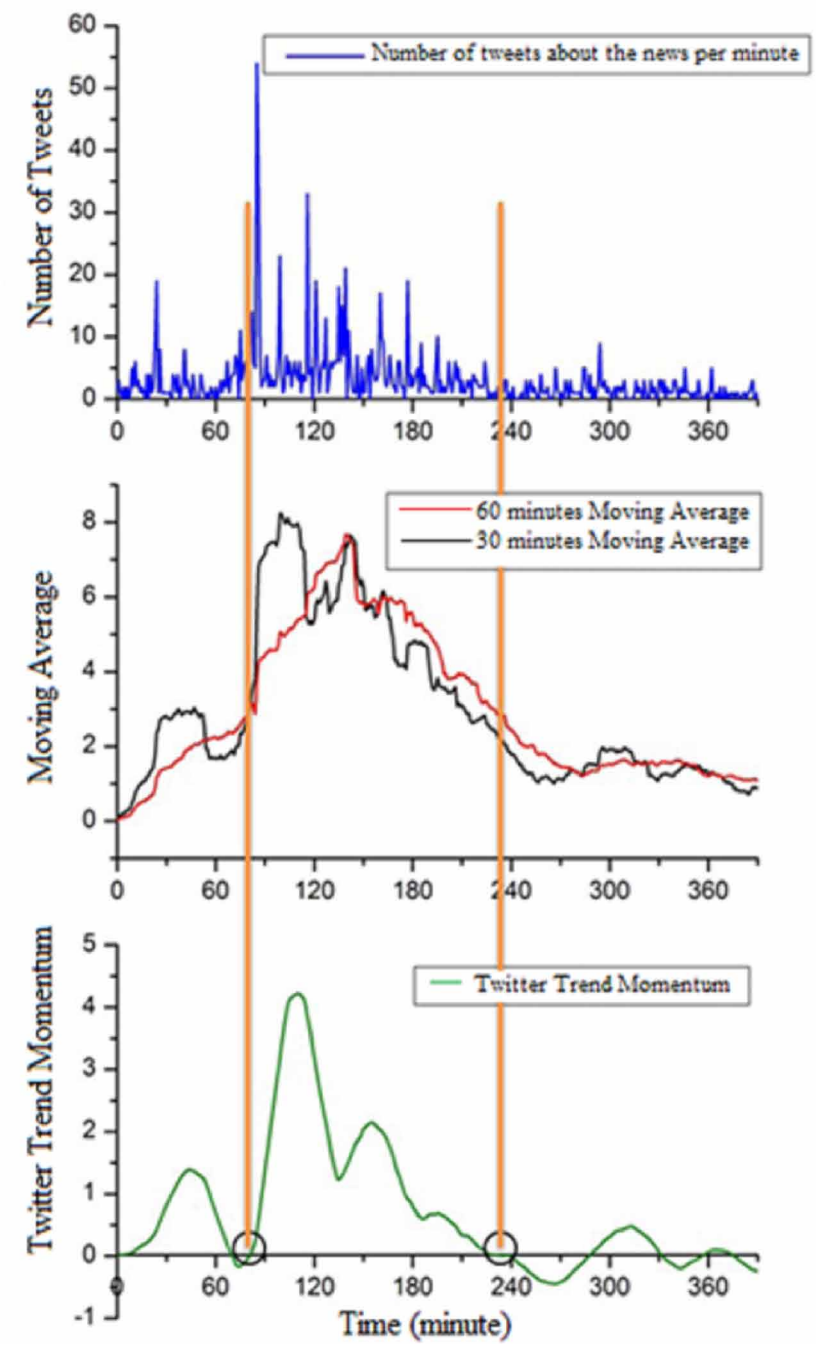


average rate of input messages increases when the number of worker nodes increase. It is observed that when a node is added to a single node cluster the average input rate increases from 380 to 564. By adding another node in the same way to a 2node cluster the input rate increases from 564 to 720, finally with one node added to 3node cluster receives nearly 920 tweets. From this we can incur that the input rate increases with the increase in number of nodes and the time taken to receive the tweets proportionately reduces i.e. we can fetch 380 tweets in 20seconds using a 4node cluster for which a single node cluster took 1minute. Thus, data collection component of our system can collect millions of tweets per day and hence the framework scale well with real-time Bigdata analytics.

Evaluation of Scalability

Here the scalability of the framework is evaluated by altering the number of nodes and vCPUs. Series of five tests were conducted as Test1, Test2, Test3, Test4 and Test5 every minute to obtain the average which is discussed here. From Figure 8 for first two results it is observed that when the number of nodes increases from 1 to 4 with 1vCPU then the output rate decreases massively. The reason behind this is single CPU cannot handle multiple thread since each worker thread uses 120% of the CPU for processing. Thus, to obtain more accurate results vCPU should be monitored 1.2 times beyond the number of nodes. The results in Figure 8 shows that raising the number of nodes by keeping the computing components such as vCPU constant the performance (output rate) of the system cannot be enhanced. For an instance if we observe the first and third results by keeping the number of nodes constant and increasing the vCPUs from 1 to 4 increases the output rate from 562 to 789.Thus we can conclude that one vCPU cannot cover computing resources of Country sentiment analysis application. From third and fourth results of graph, by increasing the number of nodes to 2 the average data fetch rate increase from 752 to 1201 per minute. In the same way in fifth and sixth

Figure 5. TTM values of news topic



row by raising the number of vCPUs and node increases the output rate from 1453 to 1656.Thus if the computing resources are adequate then increasing the number of nodes lead to high output rate. Since proposed framework system uses Twitter4j library to fetch the tweets, whose average fetch rate is 1071 messages per minute, proposed system can completely process all tweets collected, with 4 virtual CPUs and 2 nodes.

Figure 6. Twitter trend momentum of keyword "Redmi"

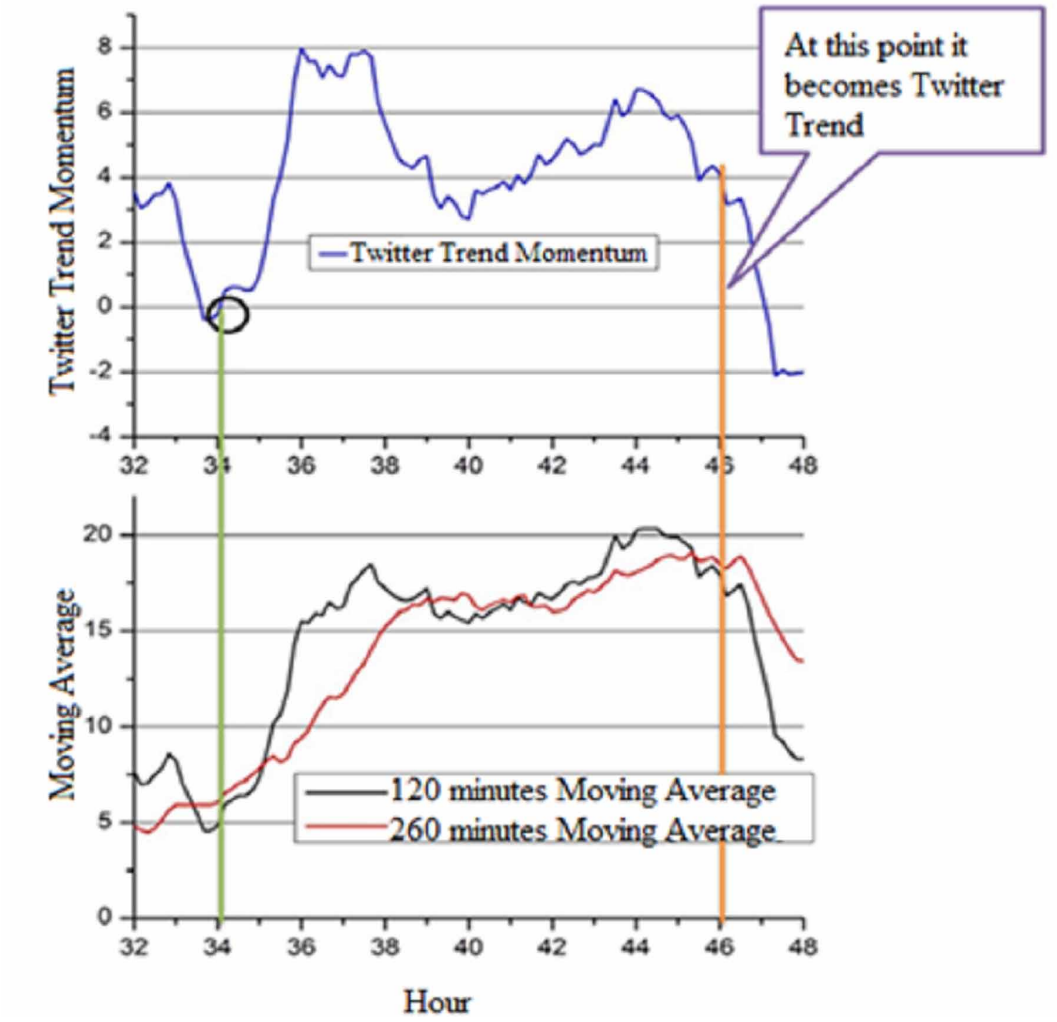


Table 3. TTM method vs FT method

Method	Prediction Point	Error Rate
TMM	25.45	9.23%
FTP μ =10	66.70	16.12%
FTP μ =15	54.27	26.13%
FTP μ =20	46.11	42.45%
FTP μ =25	34.45	43.11%
FTP μ =30	17.12	69.23%
FTP μ =35	12.13	56.12%
FTP μ =40	16.12	76.23%

Table 4. Time interval vs change rate from positive to negative

Time Interval	Change Rate
Zero-Four Hours	11.33%
Zero-Four Hours	25.45%
Zero-Four Hours	30.70%
Zero-Four Hours	11.12%

Validation of Proposed Reasons for Trend Variation

As discussed above, we proposed three main reasons for trend variation out of which first and second were for trend rising and the third one was for trend falling. Figure 9 and Figure 10 depicts the reason for trend rising on twitter. We use the keyword "RohitSharma" from our "Twitter Trends" Dataset and apply our TTM method. It gave us the prediction that the trend will rise at 12th, 35th, 92th and 112th hour. The key words and influential users monitored at the rising hours 12th, 35th, 92th and 113th are given in Table 5. From Figure 9 and the Table 5 at 12th hour the keyword "RohitSharma" is more related to the words "rotation", "opening", "tomorrow" and so on because the opening of the Indian Premier League (IPL) is 12th May 2017. Later at the 35th hour at midnight of May 12th the keyword "tomorrow" disappears. The words such as "stadium", "opening day" and names of cricketers are more related to the keyword "RohitSharma". Figure 10 and Table 5, when the league starts the trend of RohitSharma remains same connected to some player names. Every time when keywords "RohitSharma" has rising edge our method depicts and predicts the phenomenon very clearly.

Next we consider the thirds reason for trend variation i.e. trend falling due to lack of topic interaction. Figure 11 depicts the variation of eight different trends on 12th May 2017.Almost all the topics start rising one by one. This means these topics are gaining lot of attention on twitter and users are posting more tweets on these topics.

Figure 7. Evaluation of throughput

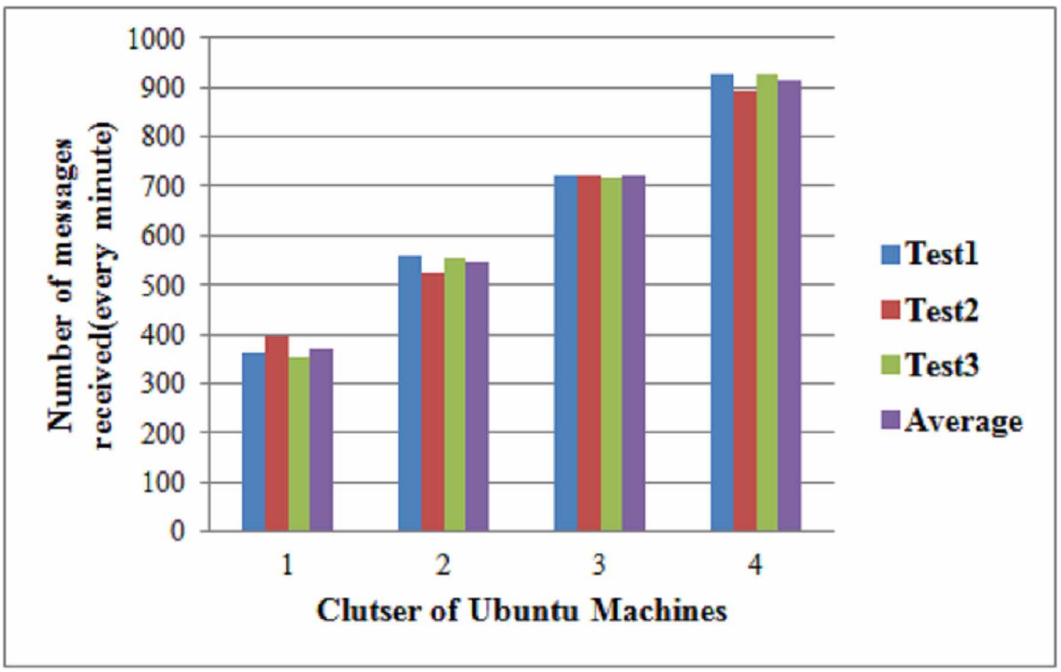


Figure 8. Evaluation of scalability

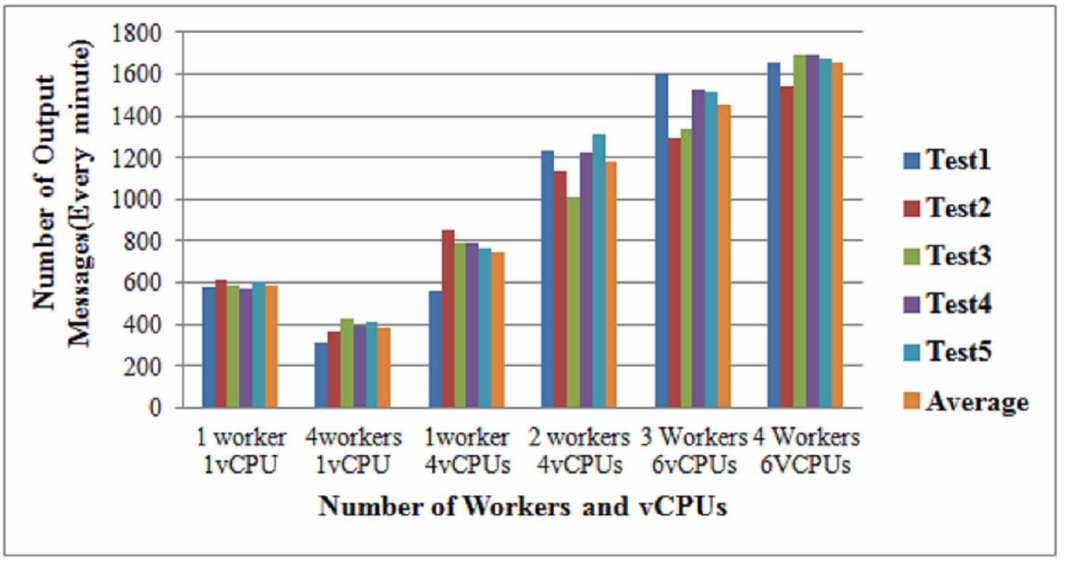


Figure 9. Rising trend results- Part1

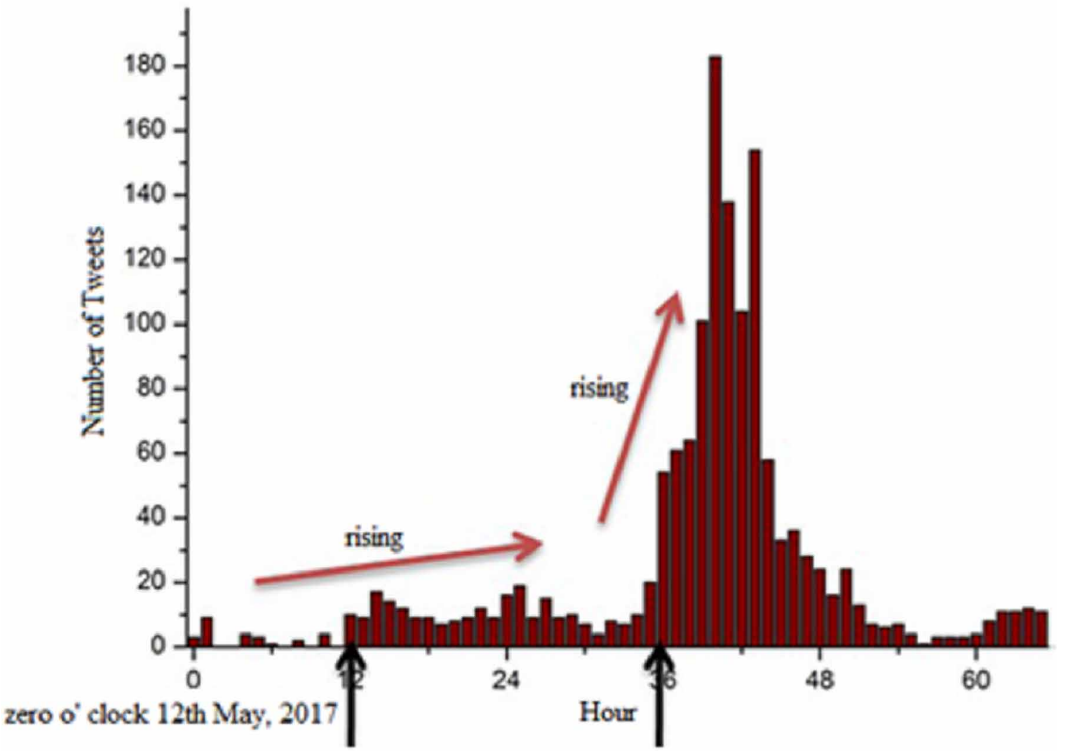
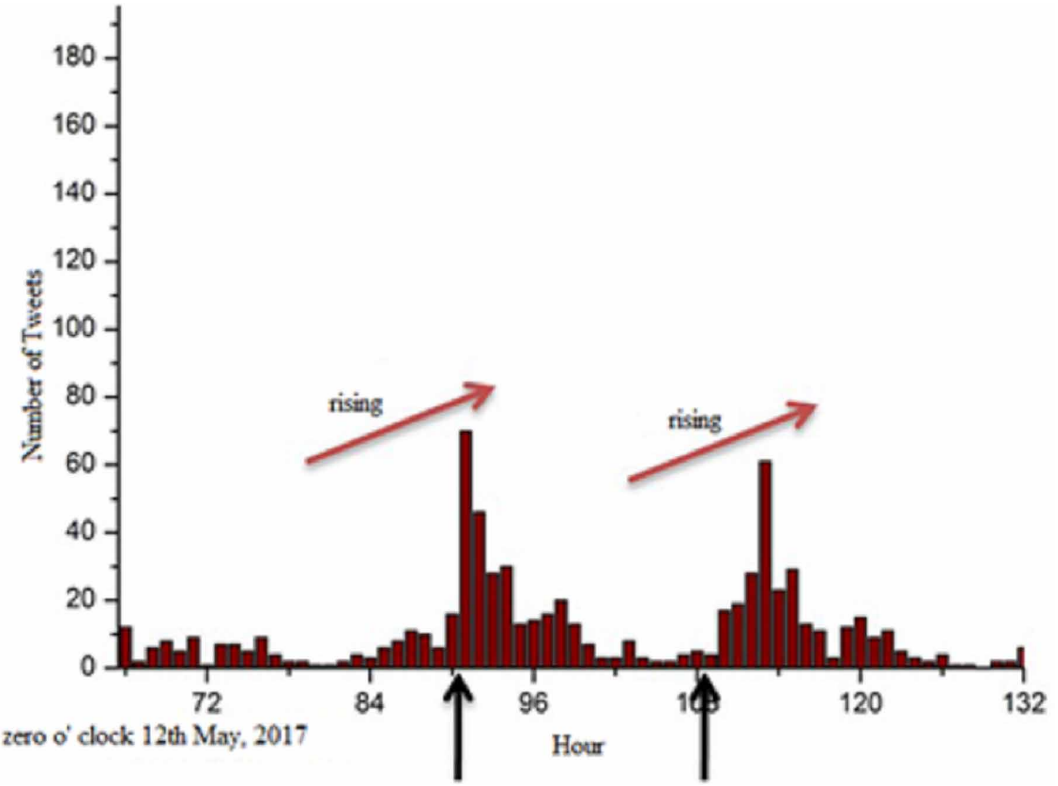


Figure 10. Rising trend results- Part2



From the above keywords it is observed that the twitter trends such as "IPL2017" and "RohitSahrma" are gaining greater importance at real-time. Here "IPL2017" is a celebration of IPL match and RohitSahrma is captain of leading cricket team MumbaiIndians in IPL. Later when the cricket match starts, users stopped talking about topic "IPL2017" and turned their attention towards particular player "RohitSahrma". Hence this led to the fall of "IPL2017" topic and hence proposed "Topic of Interaction" is one of the reasons for trend falling on twitter.

CONCLUSION

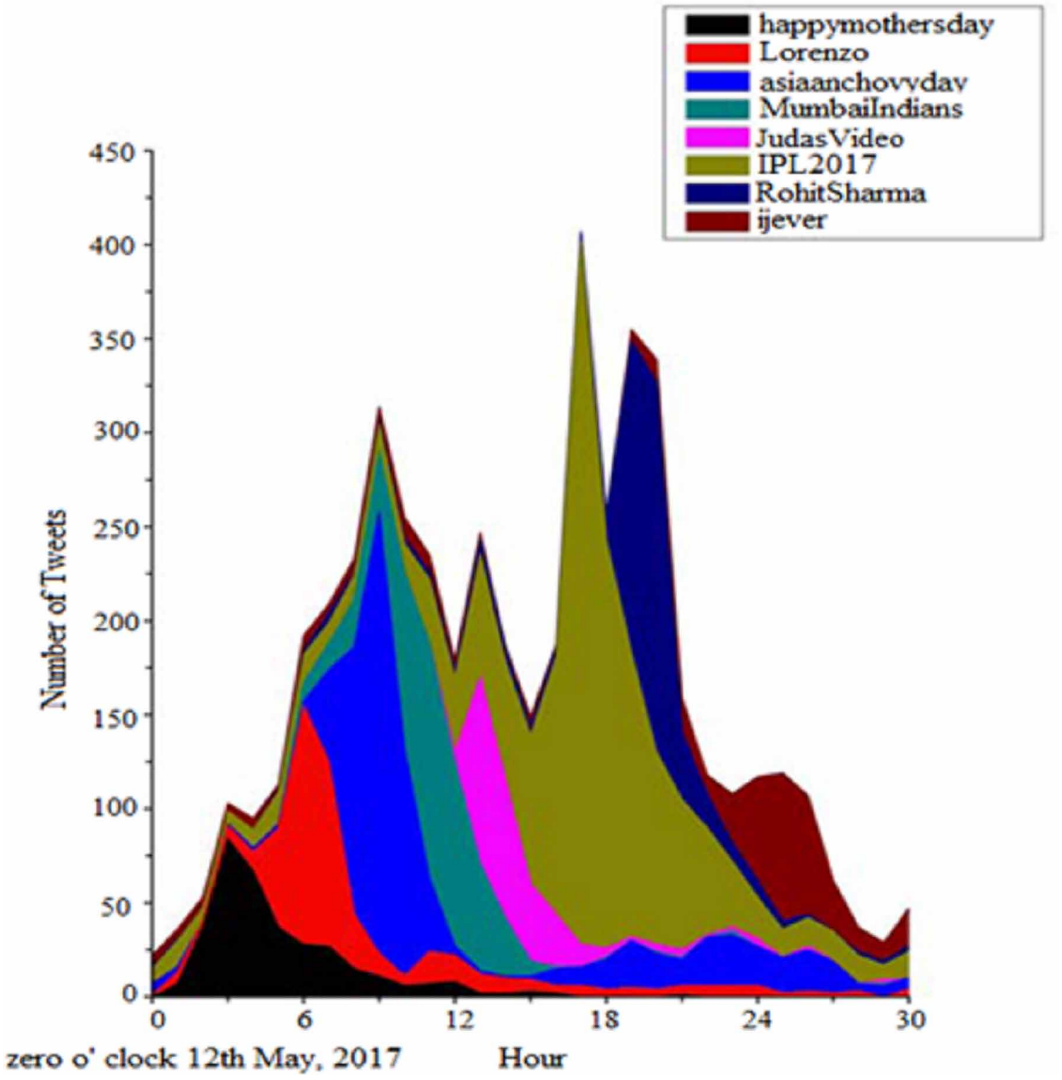
Trend analysis on twitter deals with the aspects of technical analysis which can be used to predict the future movements of emerging trends. This can benefit the companies and other organizations in planning marketing strategies and enhance their brand value to greater extent. Hence the aim of the proposed work focused mainly on implementing a framework for real-time twitter trend analysis and visualization with novel trend prediction method called TTM (Twitter Trend Momentum). The whole framework was divided into four major modules called Data Collection, Data Parsing, Data Analytics-Prediction and Data Visualization. The framework was implemented using Apache Storm module which added to the real-time analysis nature of the framework. Finally, a real-time web application called "TwitTrend" was implemented using a Dynamic Web Pages model D3.js and Python Micro server Flask over multimode. The results were tabulated by validating the TTM method against Fixed Point threshold method using 2 datasets over real-time and hence our method outperformed FT method. In order to check the performance of the framework Scalability and Throughput metrics were used and results were collected using multimode. The framework showed best scalability and

Table 5. List of key users and keywords at rising peak hours

	12th hour	35th hour	92th hour	112th hour
Key Words	Kieron Pollard CaptainRohit rotation Opening cricket tomorrow	Stadium Mumbai Harbhajan Singhopeningday throwing Pepsi Ipl2017 Win Match	Kulwant Khejroliya Karn Sharma Saurabh Tiwary Pepsi Ipl2017 RohitSharma Win	RohitSharma Stadium Mumbai Ipl2017 Win Pepsi Match
Key Users	OneWorldWarrior BurrowDweller73 milddutystation dp57 VzSports Derek_Jeter_2 RohitFan _Cousin_ MsSweetyShay87 Classic_Jayy	Official_Saints NYCKING iKontra_ PresidentRemo loveJayR Osideous Shonny_28 News12BK FullMetal_Ninja VzSports	JavaJoeMyPlace DeanLand GuyOnTheCouchNY Iowakyjotex10 NYDNsports realSCORESarodatyahoo skullzboy252 TnTFan4life	Bronx_Bombers Eliezer_Dymania Yardseller_deals kdawg1313 Jary_JariDymfc quizGriffyJr xenadanielle IxelBomber CentralPennSEO

throughput results proving it is well-suited for Real-Time Bigdata Analytics. At last the reasons for trend variations were proposed and validated and hence our reasons for trend variations are accurate. In future, more rules can be considered for MACD stock market Indicator Analysis to carry out the Twitter Trend Analysis and Prediction.

Figure 11. Falling trend results



REFERENCES

Benhardus, J., & Kalita, J. (2013). Streaming trend detection in twitter. *International Journal of Web Based Communities*, *9*(1), 122–139. doi:10.1504/IJWBC.2013.051298

Bostock, M., Ogievetsky, V., & Heer, J. (2011). D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, *17*(12), 2301–2309. doi:10.1109/TVCG.2011.185 PMID:22034350

del Pilar Salas-Zárate, M., Medina-Moreira, J., Álvarez-Sagubay, P. J., Lagos-Ortiz, K., Paredes-Valverde, M. A., & Valencia-García, R. (2016, November). Sentiment Analysis and Trend Detection in Twitter. In *International Conference on Technologies and Innovation* (pp. 63-76). Cham: Springer. doi:10.1007/978-3-319-48024-4_6

Durantin, G., Scannella, S., Gateau, T., Delorme, A., & Dehais, F. (2014, August). Moving Average Convergence Divergence filter preprocessing for real-time event-related peak activity onset detection: Application to fNIRS signals. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 2107-2110). IEEE.

Extracting Information from Text. (2017, April). Retrieved from http://www.nltk.org/book/ch07.html

Garg, N. (2013). *Apache Kafka*. Packt Publishing Ltd.

Hawker, M. D. (2010). *The Developer's Guide to Social Programming: Building Social Context Using Facebook, Google Friend Connect, and the Twitter API*. Pearson Education.

Jain, A., & Nalya, A. (2014). *Learning storm*. Packt Publishing.

Kamal, R., Shah, M. A., Hanif, A., & Ahmad, J. (2017, September). Real-time opinion mining of Twitter data using spring XD and Hadoop. In *2017 23rd International Conference on Automation and Computing (ICAC)* (pp. 1-4). IEEE. doi:10.23919/IConAC.2017.8082091

Kanade, A., Gopal, A., & Kanade, S. (2014, February). A study of normalization and embedding in MongoDB. In *2014 IEEE International Advance Computing Conference (IACC)* (pp. 416-421). IEEE. doi:10.1109/IAdCC.2014.6779360

Lau, J. H., Collier, N., & Baldwin, T. (2012). On-line trend analysis with topic models:\# twitter trends detection topic model online. In *Proceedings of COLING '12* (pp. 1519-1534).

Lu, R., Xu, Z., Zhang, Y., & Yang, Q. (2012). Trends predicting of topics on twitter based on macd. *IACSIT*, *12*, 44–49.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations* (pp. 55-60). doi:10.3115/v1/P14-5010

Wikipedia. (n.d.). N-gram. Retrieved from https://en.wikipedia.org/wiki/N-gram

Portella, G., Rodrigues, G. N., Nakano, E., & Melo, A. C. (2017). Statistical Analysis of Amazon EC2 Cloud Pricing Models.

Roesslein, J. (2015). Tweepy (Python programming language module).

Statista. (2017, April). Number of monthly active Twitter users. Retrieved from https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/

Staubitz, T., Brehm, M., Jasper, J., Werkmeister, T., Teusner, R., Willems, C., & Meinel, C. (2016). Vagrant Virtual Machines for Hands-On Exercises in Massive Open Online Courses. In *Smart Education and e-Learning 2016* (pp. 363–373). Cham: Springer. doi:10.1007/978-3-319-39690-3_32

Taneja, S., & Gupta, P. R. (2014). Python as a tool for web server application development. *International Journal of Information Communication and Computing Technology*, *2*(1), 77–83.

Wikipedia. (n.d.). Tf–idf. Retrieved from https://en.wikipedia.org/wiki/Tf–idf

Github. (2014). Twitter Wordcloud Bot. Retrieved from https://github.com/defacto133/twitter-wordcloud-bot/tree/master/assets

Weller, K., Bruns, A., Burgess, J., Mahrt, M., & Puschmann, C. (2014). *Twitter and society*. P. Lang. doi:10.3726/978-1-4539-1170-9

Yamamoto, Y. (2010). Twitter4J: unofficial Java library for the Twitter API. Retrieved from http://twitter4j.org

Zhang, W., Yoshida, T., & Tang, X. (2011). A comparative study of TF* IDF, LSI and multi-words for text classification. *Expert Systems with Applications*, *38*(3), 2758–2765. doi:10.1016/j.eswa.2010.08.066

Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E. P., Yan, H., & Li, X. (2011, April). Comparing twitter and traditional media using topic models. In *European Conference on Information Retrieval* (pp. 338-349). Springer. doi:10.1007/978-3-642-20161-5_34

Jamuna S Murthy is currently working as an Assistant Professor in Department of Computer Science and Engineering at PES University, Bengaluru. She perceived her M.Tech degree from Ramaiah Institute of Technology, Bengaluru under Department of Information Science and Engineering. She received Best Project Award for her M.Tech thesis work under year 2017. Her research interests include Real-Time Data Analytics ,Machine Learning , Natural Language Processing , Bigdata and Cloud Computing . She also holds a good publication record of peer-reviewed International Conferences, Journals and Book Publications. She is a distinguished member of National Society of Professional Engineers (NSPE).

Siddesh G M is currently working as Associate professor in Department of Information Science & Engineering, M S Ramaiah Institute of Technology, Bangalore. He is the recipient of Seed Money to Young Scientist for Research (SMYSR) for FY 2014-15, from Government of Karnataka, Vision Group on Science and Technology (VGST). He has published a good number of research papers in reputed International Conferences and Journals. He is a member of ISTE, IETE etc., He has authored books on Network Data Analytics, Statistical Programming in R, Internet of Things with Springer, Oxford University Press and Cengage publishers respectively. He has edited research monographs in the area of Cyber Physical Systems, Fog Computing and Energy Aware Computing with CRC Press and IGI Global respectively. His research interests includes Internet of Things, Distributed Computing and Data Analytics.

Srinivasa K. G. is working at National Institute of Technical Teachers' Training & Research, Chandigarh as Professor and served as the principal investigator for many projects funded by UGC, DRDO, and DST. He has been a recipient of several awards, published more than 100 research papers, and authored two books. His research areas include data mining, machine learning, and cloud computing. He received his PhD in Computer Science and Engineering from Bangalore University.