TOWARDS AN EFFICIENT DATA ANALYTIC MODEL FOR REAL-TIME SOCIAL-SENSOR EMOTIONS DETECTION

Lamanguluka N.Henock,
M.Tech (Computer Science),
Department of Computer Science and Engineering,
Sri Venkateswara College of Engineering and Technology,
Chittoor, India.

P. Jyotheeswari,
Associate Professor,
Department of Computer Science and Engineering,
Sri Venkateswara College of Engineering and Technology,
Chittoor, India.

Abstract: In this digital world, many applications like mobile devices, organization's ERPs, sensors, social networks, generate enormous amount of data. The data is so huge that conversional systems are not sufficient to process them efficiently. Hence, so many faced challenges in processing and extracting useful information from it. Thus, great interest is shown in the development of big data analytic models to cater for both real-time and off-line data processing. Therefore, in this paper an Efficient Analytic Model for Processing Real-time data has been proposed. The model comprises of three main units, i.e. Data acquisition unit; data processing unit; and data analysis and decision unit. For proposed model elaboration, a detailed analysis on real-time Social Sensor data for emotions detection is performed, using Apache Spark. Additionally, algorithms are proposed for detecting Negative, Positive and Neutral emotions in tweet-texts and locating on the map where people are tweeting from.

Keywords: Big Data analytic, Data processing, Real-time, Data acquisition, visualization, Decision Making

I.INTRODUCTION

Recently there has been a major shift in the way data and information is generated and handled in society. This change derives due to various factors we see happening in today's era. On a daily basis, the digital world generate massive amount of data, this perhaps is because we are all connected by sharing data on social networks, blogging sites, mobile devices, emails, on-line transitions etc. thus, human beings are moving data generators or social sensors who create enormous data every day. There is a steady growth in the usage of various types of sensors for observing the earth and in nearly all types of industrial applications, smart city applications, intelligent vehicles and health-care applications. All these sensors create huge amounts of data which can be of great importance if effectively collected and aggregated to extract useful information.

The data generated is continuously stored in databases and grow rapidly to the point where it becomes too complicated to manage with the present conversional systems. Through the exponential growth of these huge amount of data evolves what we call "Big Data". Big data refers to a collection of huge datasets which cannot be captured, managed, and processed by traditional conversional systems within an acceptable scope [4]. It is therefore characterized by three major elements namely: Volume, Variety and Velocity. Whereas huge volume referring to massive amount of Terabytes, Petabytes, Exabyte's and Zettabytes of data. Variety referring to various data collected from different sources, in diverse formats of structured, semi-structured and unstructured. Velocity referring to the rate at which these are created and the speed at which data is transferred. Generally data is an extremely valuable element, even more so when effectively collected and processed to get insightful information which could be significant in decision making and improving productivity. However, with these extreme volume, velocity and variety of data, traditional relational database systems are not well up to the mark to handle these types of data [5], hence great interest is shown in the development of new technologies which are capable of handling this voluminous data.

Big Data as a high speed continuous real time streaming data or large amount of off-line data brought us a realm of faced challenges, such as the critical task of converting remotely sensed data into scientific understanding. Actually, data is gathered from various sources, in which case they can be very noisy and with no meaning in them. This is because the sensors basically collect all the information. However in order to extract useful information from this noisy data, it needs to be preprocessed and filtered. The main challenge faced here is the data accuracy, simply because the data generated by the remote sensors are not in the correct format for analysis and they may also be erroneous. Therefore they need to be extracted to pull out the useful data and convert them into formats suitable for analysis.

To address the earlier mentioned matters, this paper presents an efficient analytic model for processing real-time as well as offline data. Firstly, data pre-processing is applied to the data to ensure it is machine understandable. Then error free data is transferred to base station for further execution. In the base station two forms of processing can be performed, that is real-time or offline processing. As for offline, data is directly transferred to storage devices and can thus be kept for future use, whereas for real-time data is instantly filtered to get only the relevant ones and discard rest. Afterwards load balancing is performed for equal data distribution, there by balancing processing power and improving system efficiency. Furthermore, concurrent processing servers continues processing filtered data and obtained results are compiled and sent to results storage devices or directly visualized for real-time decision making.

This paper is organized in the following manner: in Section 2, a brief summarization of previous work done has been provided.

IJCRCST © 2017|All Rights Reserved

Section 3 offers a comprehensive description of the exploited analytic model. Section 4 presents the model elaborative analysis discussion and obtained system results. Finally, section 5 gives conclusion and future directions of this paper.

II.RELATED WORK

Various researchers [1,4-6] emphasized on several issues and challenges in storing, processing and analyzing data in real-time. It is argued that to improve the accuracy of valuable information, highly efficient algorithms and technologies are needed. In addition, due to variety of heterogeneous data sources and the created volume, it is difficult to collect and integrate data with scalability from distributed locations. For instance, 170 million tweets can contain texts, images and videos which are generated by millions of accounts distributed globally. As stated, data obtained from remote locations are usually not in readily analyzable formats. Therefore, data extraction is applied, which deals with getting out effective data from the sources and delivering it in analyzable formats.

A four-layer Real-Time Big Data Analytic technology stack has been proposed. This model is utilized from the high-performance of data mining, predictive analytics, text mining, and data optimization to enhance the decision making [8]. Even though the stack was made for predictive analytics, still it serves as a good general model for real-time big data analytics [8]. Five phases of this model includes: Data extraction, development model, validation and deployment, real-time scoring, and model refresh.

Real-time Earth Observatory System is one of the extremely foreseen basic contributors of Big Data [1]. Although a single satellite's data may not be as important, together data generated by various satellites may contribute a vital sum of Big Data. Therefore, it's challenging when trying to get valuable information from the data. As a solution the authors proposed an application independent analytic architecture for processing remote sensing big satellite data.

HACE theorem has been proposed to model Big Data characteristics, and a Big Data processing model, from a data mining perspective has been recommended [3]. Moreover, the rapid growth of complex diversity and dimensionality of the Remote Sensor lies in collected metadata. Therefore, Parallel programming is required for Remote Sensor applications to predict accurate results [5].

III.PROPOSED MODEL

Irrespective of variety of technologies developed to handle big data efficiently, there has been a rise on the necessity for a big data model or architecture in scientific applications. Many efforts as stated in the literature, have already been applied towards the real-time analytic of big data. Amongst others, this paper exploited an efficient analytic model for processing real-time as well as off-line big sensed data in an efficient manner. The model is made up of three major components which are: Data Acquisition unit; Data processing Unit; as well as Data analytic and decision making unit. The following subsections further give a comprehensive description of the model and its components

A. Data Acquisition Unit

In this model, Data Acquisition unit that collects the data from different sources around the globe is presented. It is possible that the received raw data may be erroneous and noisy, therefore appropriate algorithms are applied to convert raw data in suitable formats. For example in remote sensing satellite, algorithms like Doppler or SPECAN [12], are used to convert raw data into image format. Hence for effective data analysis, remote data from different sources are integrated to reduce storage cost and also improve the accuracy of the analysis being performed. After acquisition the gathered data is transferred to a base station, there further processing can be performed on the data as required by the kind of analysis being taken. There are two ways these data can be processed further, that is, real-time Big data stream processing and off-line big data processing. In terms of off-line big data processing, the data is sent and saved to the off-line data storage devices and can be used for later/future processing when required. However, for real time, the data is directly processed as it arrives before being saved into storage devices; it is taken to filtration and load balancing server. This means that the data is being filtered on the fly to extract useful information from the big data and also the filtered data is distributed for parallel processing in a load based balanced way.

B. Data Processing Unit

This unit focuses on the execution and processing of the collected data from heterogeneous sources. In this unit, we have the filtration and load balancing server which perform two main tasks. Firstly, the data needs to be filtered by the filtration process. Secondly, the filtered data needs to be distributed to balance the processing power by the load balancing server. The Filtration finds the useful information from the big data and discards the remaining data. This varies from analysis to analysis, and the results of system performance are improved. Like filtering algorithm, the load balancing algorithm changes from analysis to analysis.

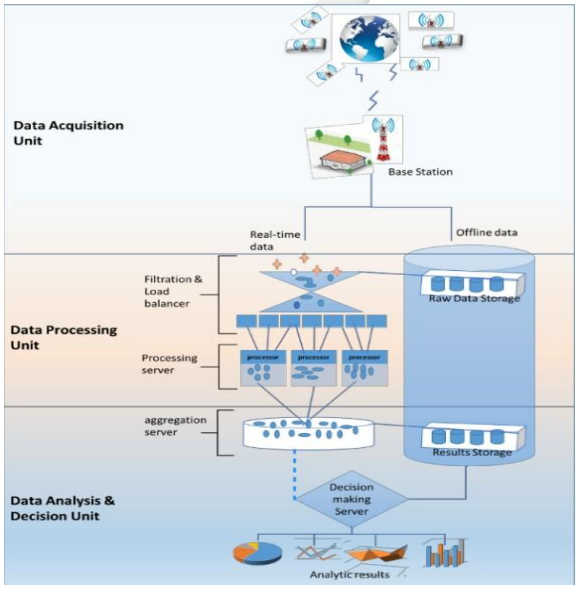

Figure 1 : Analytic model *Data Analysis and Decision Unit*

For example, if out of a big data from diverse sources we only need temperature and humidity data, then only the required data (temperature and humidity) will be filtered out of the big data and the rest will be discarded. The load balancing server

has capabilities of segmenting the whole filtered data into small chunks; each of these chucks will then be distributed across several processors (processing servers) available for parallel processing. Every processing server consists of its implementation algorithm for processing incoming segments of data from the Filtering and Load balancing server. This means that the processing servers performs some measurements, statistical calculations and makes other logical or mathematical operations to produce the intermediate results from every segment of data. Since each of the processing servers executes the tasks in parallel and independently, the proposed system drastically boosts the performance and the results of each processor are generated in real-time. The obtained outputs are then transferred to the compilation server where they are organized, compiled and stored or directly visualized as shown in figure1.

This unit comprises of three main components, "the aggregation and compilation server, results storage server(s), and decision making server". Once the outputs of each distinct processing server from the previous unit are all set for assembling, they are separately sent to "the aggregation and compilation server". At this point the results are not in an organized nor compiled form. Thus, all affiliated output has to be compiled and organized into suitable formats where further operations can be performed on the data. As proposed, "The aggregation and compilation server" is associated with supporting algorithms which again differs for each analysis requirements; this is because each analysis can have a different scenario. Once the results are compiled and organized, the Aggregation server send them for storage in the result's storage so that it can be accessed whenever its needed. Similarly, a replica of the results is send to the decision-making server so that in real-time it can also be processed further for making required decisions. The decision-making server like other severs also contains supporting algorithms and with these various queries can be parsed onto the processed data to return required results that will help in making decisions.

It is highly recommended that these algorithms be robust and accurate enough to accurately generate output in order to detect hidden information and help make right decisions. This unit of the model is very important and should be handled efficiently, because any slight error in the decision-making can reduce the accuracy of the entire analysis. The Data Analysis and Decision Unit finally broadcast the results and other applications (e.g. business software) can make use of those decisions in real time.

IV.RESULTS AND IMPLEMENTATION

A. Analysis Discussion

Using the proposed model for processing real-time as well as offline data, a simple analysis on streaming twitter data has been performed. In assumption that the data is massively and continuously coming from Twitter at a high speed rate, it is then difficult to handle it with a single processor. Thus, use of parallel processing will highly increase the system's efficiency. Therefore, to acquire process, analyze and make decisions from this big data, special algorithms are required. This section presents the analysis of Social Sensor data to detect positive, negative and neutral emotions in a tweet in real-time by following the proposed model and algorithms for filtering ,parallel processing and making decisions on the streaming data and display on the world map where people are tweeting from.

1.A.1 *Dataset and tools used to perform the analysis*

Firstly, a previously labeled twitter dataset is needed. This will give basis from which to compute the probability that a certain tweet-text will fall into a specific class of emotion i.e. positive, negative or neutral. Therefore a training dataset containing more than 1 million labeled tweets, made available by sentiment140 is taken. Algorithms for handling processing, analysis and decision making to detect the emotions in tweets were proposed in accordance with the proposed model. Apache Spark framework with Scala programming on a single cluster node setup is used for the implementation of the algorithms for this analysis. Since Spark provides the capabilities for parallelism, in-memory data processing and high-performance computing, it is thus opted suitable for analyzing large amount of streaming data. The mechanism for load balancing used by Spark is also similar to that of the proposed model; therefore preference is given to Spark. Furthermore for offline data storage, Hadoop Distributed File System (HDFS) gained preference to store raw unprocessed data as well as processed results when required as proposed by the model. For visualization of data on the world map, processed results are published to Redis server – an in-memory data structure store, which is further subscribed by the webApp based on Datamaps for visualization.

Table1. Summary of frameworks / technologies used

Technology	Description
Apache Spark	Apache spark is a general engine that is used for processing big data on a large-scale at a very high rate. It contains stacks of libraries like SQL and DataFrames, MLlib for machine learning, GraphX, and Spark Streaming. For this application a combination of some of these libraries are used.
Apache Hadoop	Hadoop is an open source platform for massive datasets which supports distributed processing and distributed storage. It process very large data sets on clusters built from commodity hardware. In this application, HDFS – Hadoop Distributed File System has been used as an optional to save raw or processed data.
Datamaps	Datamaps is used to provide visualizations based on geographical data. It overly relays on D3.js (Data-Driven Documents)– a JavaScript library for visualizing data using web standards.
Redis Server	Redis which stands for Remote Dictionary Server, is an in-memory data structure storage which is used as a cache, database and message broker. It is free and open-source.

1.A.2 *Algorithms and descriptions*

a) Filtration and Load balancing

Input: Streaming Twitter Data

Output: Filtered Tweets divided in blocks, send each block to processing server

Steps:

1. Filter incoming tweets. Discard rest of unnecessary data.
2. Divide filtered data into blocks
3. Assign and transfer each block of filtered data to available processing servers.

Description:

The algorithm above takes in twitter data and then filter out all the unnecessary data. Based on our analysis, the algorithm takes in only those tweets with location; any other tweet without it is regarded as unnecessary data and is discarded. Filtered data is then divided into equal blocks or chunks for load balancing depending on the computation power.

b) Data Processing and execution

Input: Filtered blocks of data

Output: Statistical parameter results and transfers them to aggregation server.

This algorithm employs the Naive Bayes Technique, since it is faster to predict classes of data, performs well in multi-class predictions just as being performed in this analysis and thus, preference is given to it as good option for real-time detections.

Let:

- Feature(s) = word(s) in a tweet
- Frequency = frequency of word in a tweet
- Label = class of Negative, Positive, Neutral
- P = Probability

Steps:

1. For each Block, compute:

a) Frequency of feature

b) Likelihood of a features showing P of Label

2. Transmit results of each block to aggregation server

Description:

The data processing algorithm provides the main functions of operations to be performed. It computes results for each incoming block and sends them to the next unit for compilation. In our analysis, the frequency of words in a tweet is calculated and the likelihood of a tweet showing probability of being in a certain label is computed. This is done for each block and results are send to the next unit for further processing.

c) Results aggregation and compilation

Input: Processed results from each block

Output: Compiled, stored results and sending results to decision-making server

Steps:

1. Gather each block's results
2. Compile previous results by using following formula:

$$P(\text{label} \setminus \text{features}) = \frac{P(\text{label}) * P(\text{features} \setminus \text{label})}{P(\text{feature})} .(1)$$

3. Transmit results to next unit for decision making
4. Store processed results into HDFS result storage and send copy to decision server

Description:

This part of algorithm collects results of each block from previous unit, then compile , organize and stores results into HDFS (optionally), else if no storage is required, the result is directly send to decision server and then to other applications that requires it.

d) Decision Making

Input: Compiled final results

Output: Each tweet with allocated emotion

Steps:

1. Obtain posterior probability of a tweet for each label computed in previous unit.
2. Compare probability of all three labels
3. Label (i.e. negative, positive, neutral) with highest probability will be selected as emotion for that specific tweet.

Description:

Compiled results from the aggregation server are obtained and analyzed to determine whether a tweet belongs to a negative, positive or neutral category. Relative probability for all labels are obtained from previous unit, and then compared. The label with the highest probability out-powers the others and the tweet will be categorized as such label. Each incoming block of data will undergo all these processes for emotion detection in real-time. The above algorithms have been developed and tested on Intel Pentium 2.16GHz x4, Ubuntu 16.04LTS local machine with Apache Spark single cluster node setup having 4GB RAM and Scala Programming language.

(a)

(b)

Figure2. system results: (a) Positive detections (b) Neutral detections

(c) Negative detections

Efficiency measurements are taken by considering the average processing time to process a number of data of various batches. Since the data is streaming and processed in batches, we were able to measure the average processing time of about 93 batches with each batch containing between 800-900 events.

Spark jobs implementation of the analyzing algorithms takes less than 16 seconds to process one batch of events. The average processing time for 93 batches is about 14 seconds 680 milliseconds. We have experienced a total delay of about 40seconds 20milliseconds. Differences in processing time and delays may be caused by networking issues such as differences in bandwidths. The efficiency measurements are shown in the graph shown in fig.3.

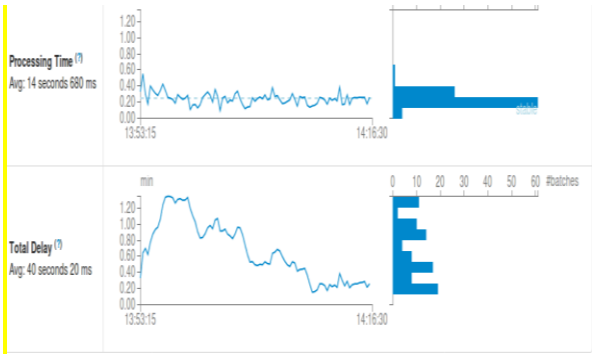

Figure3. Processing Time

Results : The analysis was done using Apache Spark and its components i.e. Spark streaming – connecting to twitter using API and streaming tweets in real-time, Spark MLib – for algorithms to predict and detect emotions in tweets based on the text. Spark takes in streaming tweets as input, filters tweets as they come in to consider only those tweets with location and discard the rest. This filtration will help facilitate visualization of tweets on the world map, by using latitude and longitude information contained in the tweet. Continuous streams of data flows into Spark, they are filtered and divided into sequence of abstractions called Resilient Distributed Dataset (RDD) representing fault tolerant arrays of objects distributed across cluster nodes. They are processed in parallel by Spark Engine systems and upon completion of processing every batch file results are compiled as a single file and saved in HDFS or directly transferred to requiring applications. We tested and evaluated our algorithms with respect to the accuracy and processing time using twitter streaming sets. Prediction accuracy evaluation is done using two parameters namely: the True Positive (TP) – which shows the % of correctly identified tweets; the False Positive (FP) – showing the % of incorrectly identified tweets. Our system detected real-time streaming social sensor data as negative, positive and neutral with an overall accuracy of more than 90% TP and utmost 10% FP. The accuracy results of detection in which positive, negative and neutral tweets are detected is shown in fig.5.

V.CONCLUSION

This paper proposed and exploited an analytic model for processing real-time big data. The proposed model has been tested on real-time Twitter social networking data and it has efficiently processed and analyzed the data to detect user's emotions in real-time and further located on the world map where people are tweeting from. The model comprises of three main units namely: Data acquisition unit, Data processing unit as well as Data analysis and decision unit. The units' implements algorithms to filter data process the data and make decisions respectively depending on the required analysis. The model is generic and can be used for any type of big data analysis. Furthermore the model's capability of performing filtration, load balancing and parallel processing of only useful data made it better choice for real-time data analysis. The provided algorithms in this paper are used to analyze social sensor data which helps in detecting people's emotions in real-time. Proposition of other algorithms for any type of real-time data analysis are welcomed as well as improvement in the accuracy of the above proposed algorithms.

VI.REFERENCES

[1]. M. Rathore, A. Paul, A. Ahmad, B. Chen, B. Huang, and W. Ji, "Real-Time Big Data Analytical Architecture for Remote Sensing Application", *Institute of Electrical and Electronic Engineering*, vol. 15, no. 2, pp. 809–818, 2015

[2]. J. Lu, B. Amen, "Sketch of Big Data Real-Time Analytics Model", *International Conference on Advances in Information Mining and Management*, pp. 48-52, 2015

[3]. D. Tamhane, and S. Sayyad, "big data analysis using HACE theorem", *International Journal of Advanced Research in Computer Engineering & Technology*, pp. 2278 – 1323, 2015

[4]. H. Hu, T. Chua, X. Li, "Toward Scalable Systems for Big Data Analytics: A Technology Tutorial," *Institute of Electrical and Electronic Engineering*, vol. 2, pp. 652-687, 2014.

[5]. Y. Ma, H. Wu, L. Wang, and B. Huang, "Remote sensing big data computing: Challenges and opportunities," *Future Generation Computer Systems*, vol. in press, 2014

[6]. K. Kambatla, G. Kollias, V. Kumar, and A. Grama, "Trends in big data analytics," *Journal of Parallel and Distributed Computing*, vol. 74, pp. 2561-2573, 2014.

[7]. J. Fan, and H. Liu, "Challenges of Big Data Analysis," *National science review*, vol. 1, pp. 293-314, 2014.

[8]. M. Barlow, "Real-Time Big Data Analytics: Emerging Architecture", pp. 24-26, 2013.

[9]. Z. Zou, F. Deng, Y. Bao, and H. Li, "An approach of reliable data transmission with random redundancy for wireless sensors in structural health monitoring," *Institute of Electrical and Electronic Engineering*, vol. 15, no. 2, pp. 809–818, 2015.

[10]. P. Chandarana, and M. Vijayalakshmi, "Big Data analytics frameworks*," in Proc. Int. Conf. Circuits Syst.Commun. Inf. Technol. Appl*., pp. 430–434, 2014

[11]. R. A. Dugane and A. B. Raut, "A survey on Big Data in real-time," *International Journal of Recent Innovation Trends Computing & Communication*., vol. 2, no. 4, pp. 794–797, 2014

[12]. W. Yang, X. Liu, L. Zhang and L. T. Yang, "Big Data Real-time Processing Based on Storm*," in proc. Int. Conf. Trust, Security and Privacy in Computing and Communications*, pp. 3-5, 2013

[13]. X. Yi, F. Liu, J. Liu, and H. Jin, "Building a network highway for Big Data: Architecture and challenges," *Institute of Electrical and Electronic Engineering*, vol. 28, no. 4, pp. 5–13, 2014.

[14]. F. Zhang, X. Li, and Y. Wang, "Research on Big Data architecture, key technologies, and it's measures," in Proc. *Institute of Electrical and Electronic Engineering*, Int. Conf. Dependable Auton. Secure Comput., pp. 3–7, 2013.