Sentiment Analysis on Twitter Data using Apache Spark Framework

Hossam Elzayady
hossamelzaiade@gmail.com
Military Technical College

Khaled M. Badran
khaledbadran@mtc.edu.eg
Military Technical College

Gouda I. Salama
gisalama@mtc.edu.eg
Military Technical College

Abstract— **Sentiment analysis has become an interesting field for both research and industrial domains. The expression sentiment refers to the feelings or thought of the person across some certain issues. Furthermore, it is also considered a direct application for opinion mining. The huge amount of tweets jotted down daily makes Twitter a rich source of textual data and one of the most essential data volumes; therefore, this data has different aims, such as business, industrial or social aims according to the data requirement and needed processing. Actually, the amount of data, which is massive, grows rapidly per second and this is called big data which requires special processing techniques and high computational power in order to perform the required mining tasks. In this work, we perform a sentiment analysis with the help of Apache Spark framework, which is considered an open source distributed data processing platform which utilizes distributed memory abstraction. The goal of using Apache Spark's Machine learning library (MLIB) is to handle an extraordinary amount of data effectively. We recommend some Preprocessing and Machine learning text feature extraction steps for getting greater results in Sentiment Analysis classification. The effectiveness of our proposed approach is proved against other approaches achieving better classification results when using Naïve Bayes, Logistic Regression and Decision trees classification algorithms. Finally, our solution estimates the performance of Apache Spark concerning its scalability.**

Keywords— Big Data; Apache Spark; Twitter; Sentiment Analysis; Text Classification; Machine Learning techniques

I. INTRODUCTION

Now social networks sites make the entire world a small village, where users can share their views, feelings, experiences, advice through those sites so that others can get help from these [1]. Since many of us use social media daily, a huge quantity of comments, opinion, article have been created. Finding an automatic manner to investigate and classify users' opinions in social networks could be very essential. This is because it is the most empirical way to obtain the direct remarks or information from people. The method of classifying texts or documents in keeping with their polarity is referred to as Sentiment Analysis (SA) [24]. Sentiment analysis, which is part of Natural Language Processing (NLP), is set to extract the meanings from a text in a way to find the polarity of the text [1-6]. For the sentiment analysis, we focus our attention in the direction of the Twitter, a micro-blogging social networking website, where users can communicate with each other or share their opinions in short blogs. Large different number of text posts exists on twitter which increases every day, the rapid enormous data growth make the existing databases unable to handle an extensive amount of data in a short time. Also, these databases type designed to process structured data but there is a limitation on it when dealing with huge data. So the conventional solutions are not helpful for organizations to manage and process unstructured or large data [2]. Frameworks, such as Hadoop, Apache Spark, Apache flume and distributed data storages like Hadoop Distributed File System (HDFS), Cassandra and HBase are being very widespread, as they are designed in a manner which facilitates the process of huge amounts of big data and makes it almost effortless [3,15,12]. One of the most effective manners is the parallel computing techniques when dealing with big data, which include multicore processors, distributed computing, and etc. Dividing problems into many sub-problems, to be processed using threads and machines in the cluster is the main feature of Parallel computing methods. One of the most important advantages of parallel computing is enhancing the performance of the algorithms to be faster than the serial manners [4,13]. In this paper our goal is to establish a Sentiment Analysis methodology of Twitter data dependent on Apache Spark platform, which using supervised learning techniques in order to perform tweets classification, we concentrate on The change shown on the accuracy as increasing the size of training dataset and reducing running time as a result of Spark's Cluster expansion, Section 2 shows the related work. Section 3 presents big data characteristics and Spark architecture, while in Section 4 classification algorithms used in our suggested system are briefly discussed. Steps of training are explained in Section 5. Furthermore, Section 6 shows the results extracted from our conducted experiment. Eventually, Section 7 briefly summarizes the overall conclusion and displays directions for future work.

II. RELATED WORK

In the previous years, studies of Sentiment Analysis and emotional models had a wide attention. The reason for that is basically due to the recent enlargement of data which exists on the social networks, particularly of those that describe people's point of view, thoughts and comments [5]. Walaa Medhat et al. [6] presented and discussed in brief details different types of sentiment analysis and its applications. Algorithms and their originating references of various SA techniques are categorized and shortly explained. Yang et al. [7] introduced the common sentiment analysis methods from the perspective of machine learning technologies, which encompass Naive Bayes technique, Maximum Entropy method, Support Vector Machine technique, and Artificial Neural Network method and performance assessment and difficulties. Pang et al. [8] were the first to apply Machine Learning for sentiment mining on movie reviews corpus, many classification algorithms were used, whereas unigram and bag of words are utilized for obtaining features. The ratio of accuracy differs according to what they applied for example it was 82.9% by applying Support Vector Machines, while it was78.7% 78.8 by applying Naive Bayes classifier. Wang et al. [9] used training dataset which contain 17000 Tweets to come up with a real time Twitter Sentiment Analysis System regarding to U.S. voting Presidential Election Cycle in 2012. In [1], authors introduced a new method combining the help of SentiWordNet alongside with an implementation of Naive Bayes; therefore more accuracy can be achieved. One of the possible techniques to get more accuracy of classification of tweets is applying SentiWordNet and Naive Bayes that give positive, negative and objective degree of the words exist in tweets. Bindal et al. [10] proposed a two-step system can be applied for sentiment classification of the tweet. During the initial step, sentiment lexicons are used to classify tweets, while the polarity of each tweet is also assigned by aggregating the scores of each token. During the next step, the SVM classifier receives all the tweets with low absolute scores to strengthen the whole accuracy. In [11], authors introduced a novel manner for Sentiment Learning depend on Spark platform; the hashtags and emotions within a tweet are exploited by the suggested algorithm, as sentiment labels, and continue to a classification step of various sentiment types using parallel processing methods. In [12], authors suggested a real-time solution using spark framework, for processing sentiment analysis Saudi dialect in twitter based on lexicon-based algorithm. In [13], authors recommended an efficient sentiment prediction technique in Big Data, using Spark. The outcomes got from the suggested work were subject to analysis to demonstrate high levels of scalability in relation to accuracy and time. It was noted that even with the growth of data volume, the processing time indicated very less variance. Authors in [14] suggested a system based on an SVM alongside with a rule-based classifier in order to enhance system accuracy. Authors in [2] suggested pre-processing of data to ignore noise and implemented sentiment analysis for movie dataset, on Hadoop framework and analyzed with a great number of tweets. Yan et al. in [4] proposed a microblog sentiment classification approach with parallel support vector machine (SVM) technique and Spark is utilized to improve the performance.

III. Big Data and Spark Architecture

Big data is defined as the process that is applied when conventional data mining and handling techniques are unable to get the insights and meaning of the underlying data. The relational database management systems engines cannot process unstructured or large data anymore [21]. So dealing with such data type needs a different processing approach called big data, which utilizes enormous parallelism on readily-available machines. Therefore, Apache Spark developed to make processing and analyzing the data easier. Big data has three main characteristics.

A. Big Data characteristics

- *Volume*

  Big data describes huge volumes of data. Creating and modifying data were one of the main employees' jobs. But Nowadays data is generated with the aid of networks, machines and human interaction on systems such as social media, so the quantity of data to be analyzed becomes vast [22]. New research statistics show that Twitter produces 12 terabytes of blogs within one day, also, Facebook declared that 2.7 billion "likes" and "comments" and messages were daily registered by the Facebook users [15].

- *Velocity*

  Today Data is generated too fast and also need to be processed fast. Social media is one of the most important means of providing data. Social sites are continuously producing a complex unstructured and semi-structured shape of data and presently created 90% of data in the last two years [15]. Increase the velocity of big data is based on using modern devices such as smart phones and televisions and more advanced technology [22]. The Internet is the primary factor for gathering enormous data. Amazon Web Services Kinesis is an example for data velocity handler streaming application.

- *Variety*

  Sources and kinds of structured and unstructured data are multiple and various. Sources such as spread-sheets and databases have been used in the previous years to store and keep data. Now data comes in many different forms like photos, audio, videos, emails, PDFs, monitoring equipment, etc [22]. Due to the great diversity of unstructured data, many problems arise during storage, mining and analyzing data. All organizations, companies and manufacturers currently have nearly 85% of the data, structured and semi-structured and unstructured shape of data [15].

B. Spark framework

Apache Spark is one of the newer open-source, lightning fast big data distributed processing platform based on the same principles as Hadoop, which is developed in a manner to evolve the computational speed. Hadoop has some problems relating to performance in specific cases, as in iterative tasks or graph based algorithms. Also, Hadoop can't cache intermediate data for greater performance however rather; it flows the data to the disk between each step read and write. Spark minimizes the number of read and write cycle making it 10 times faster than Hadoop on disk in applications running; also, keeping intermediate data in-memory makes Spark 100 times faster when dealing with memory [18]. Resilient distributed data set (RDD) may be the key theoretical idea in spark, which represents a read-only collection of objects divided across a set of nodes, with the ability to rebuild them if arise an occurrence of losing any partition [16]. Spark is best known for its capacity for accomplishing batch, interactive, and machine learning in addition to streaming all in the same cluster. One of the main features of Spark is its scalability, so the cluster can be provided with n number of nodes. Spark offers Application Programming Interface (API) in Java, Scala, Python, and R programming languages. The architecture of Apache Spark Framework is depicted in Fig1 [19].

C. Spark cluster Processes

There are two essentials kinds of processes in the spark cluster which are a driver program and multiple executors. Those processes keep running within the same java virtual machine (JVM). In a cluster, these processes ordinarily run on separate nodes. The driver is considered the process through which the main () method of your program works and running the user code which is responsible for creating a Spark Context, RDDs, and executing all transformations and actions . Spark driver is responsible for changing a user program to units of physical execution, known as tasks. The responsibility of Spark executors, which are worker processes, is to run the individual tasks in a given Spark job [19]. The ability of cluster manager for being plugged in Spark permits Spark to work on top of various external managers, for example, YARN and Mesos, in addition to its built-in Standalone cluster manager. Fig 2 shows the architecture of Spark in distributed mode [23].

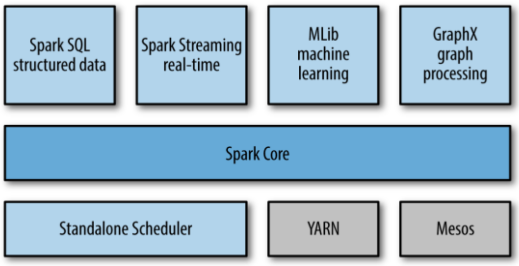

Fig. 1: The Spark stack.

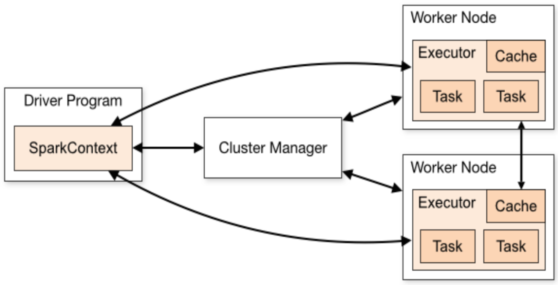

Fig. 2: The components of a distributed Spark application.

D. MLLIB

Spark MLlib is defined as a distributed machine learning framework on top of Spark Core. The Spark excels at iterative algorithms which makes it perfect for executing Machine Learning algorithms since the majority of Machine Learning techniques depend on repeated jobs. The main aim behind MLlib design is to make machine learning scalable and easier processing for data. MLlib has a high performance as a result of its design to run in parallel on clusters. MLlib has a lot of algorithms and uses for general Machine Learning techniques, including Clustering, Regression, and Classification[19].

IV. Machine learning techniques

Within the next 10 years, machine learning algorithms may substitute nearly a quarter of the jobs across the world, this is a result of the availability of many easy and fast programming tools such as Python and R and massive increase in big data. Supervised and unsupervised techniques are two branches of machine learning techniques. Supervised machine learning techniques heavily based on training data which has labeled data, unlike the theory of unsupervised machines learning techniques. Depend on the provided training data, Supervised machine learning classifier will classify the rest data i.e. test data[17].

In this study, Naive Bayes, Logistic Regression and Decision Trees algorithms were utilized to execute the Sentiment Analysis.

A. Naïve Bayes

Naive Bayes is an easy technique for classification algorithm built on the application of Bayes' theorem, where every instance of the problem is shown as a feature vector, and it is presumed that the value of each feature is separate and independent from the value of any other feature [1]. For example, There may be a possibility that the fruit is an apple in the event that its color is yellow, round and about 3 inches in diameter. Regardless of whether these features based on each other, these properties may participate to the property that this fruit is orange and subsequently the name "Naive"[6]. The algorithm itself is derived from Bayes theorem, which is declared as follows:

$$P(H/X)= [P(X/H)P(H)]/P(X) \qquad (1).$$

Where H and X are events and $P(X) \neq 0$ [2]. It makes the hypothesis that words are created independently of word position. Naive Bayes has many merits, on top of which need of a small number of training data.

B. logistic regression

It is a regression analysis model and mostly used where the dependent variable is Binary which could take one out of a consistent number of values. It aims to measure and explain the relationship between the instance class and the extracted features from the input, not only it is widely utilized for binary classification (problems with two class values), but also it can be used to tackle multiclass classification problems [3].

C. Decision Tree

A decision tree is one of the machine learning algorithms, where it is used extensively as a result of its adaptation with almost any type of data. It is one of the methods of displaying a classification algorithm that is highly depend on a tree structure. In this structure, the leaves indicate class labels and branches point to the conjunction of features that result in the aforementioned classes. Fundamentally, it implements a recursive binary partitioning of the feature space. Each stage is chosen greedily, aiming for the optimal selection for the given stage by increasing the information gain [3,6].

V. PROPOSED SOLUTION

Our spark cluster consists of 4 virtual computing nodes, each one has 3.4GHz CPU processor Quad-Core, 100GB hard disk, 12GB of memory and the machines are connected by 1 gigabit Ethernet. On all machines, we install Ubuntu 16.04 operating system, Java 1.8.0_66 with a 64-bit Server VM, Hadoop 2.6.5 and Spark 1.6.0. One of the nodes acts as the master and the others act as the slave.

Some modifications are applied on the default Spark configurations, using 6 total executor cores (2 for each slave nodes), and setting an 8GB for executor memory and a 4GB for the driver memory.

PySpark was utilized as the favored language to implement Naive Bayes, Logistic Regression and Decision Trees algorithms.

The overall architecture of the suggested system is depicted in Fig 3, taking into consideration the previously discussed aspect.

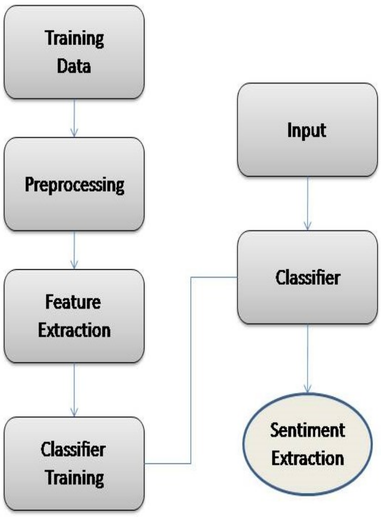

Fig. 3: Proposed system architecture

A. Data collection

Twitter's APIs for gathering data is categorized into The Search API is utilized to gather Twitter data based on hashtags while the stream API is utilized to stream real time data from Twitter [17].

B. Dataset

Dataset the Twitter Sentiment Analysis has 1.578.627 classified tweets. In each row, 1 stands for positive sentiment while 0 indicated the negative [3].

The main sources of the dataset are University of Michigan Sentiment Analysis competition on Kaggle and Twitter Sentiment Corpus by Niek Sanders[20]. Dataset which was described previously has been gathered by Twitter API.

C. Data Preprocessing

Data extracted from Twitter will need to be processed because it typically contains completely different non-sentiment terms like hashtag, website link, white spaces, emotions etc, Which ought to be ignored before processing it so that the sentiment produced are precise. Preprocessing includes:

- *Removal of URL's*
  There is a lot of information on twitter data, so in case of posting any link which is irrelevant to sentiment analysis, URL ought to be deleted from the tweet.
- *Removal of special symbol*
  Symbols which are utilized by the user like full stop (.), punctuation mark (!), etc. and doesn't have sentiment should be ignored from the tweet.
- *Removal of Username*
  Twitter provides each user with a special username which he uses to indicate his tweets and always starting with @ .It is identified as proper nouns for example, @username. This also should be deleted for accurate analysis.
- *Removal of Hashtag*
  Hashtag is preceded by hash symbol (#), and used for identifying topics or phrases that are currently in trend. For instance, #Egypt.
- *Removal of additional white spaces*
  Additional white space in the data affects the analysis, so it should be removed to get an effective analysis.
- *Removal of stop words*
  Stop words are the words that add nothing and give little information, so they are useless for taking them as features such as. “the”, “for”, “her”, “a”, etc. Stop words should be deleted from the tweets.
- *Lower casing*
  Lower casing is essential in order to ensure that the term (in our issue word) converted to respective feature, i.e. “Hello” and “HeLlO” ought to be converted to “hello”. This stage guarantees correlation within feature set.
- *Standardizing word*
  Sometimes words don't seem to be within the right formats. For instance: “I loooveeee you” need to be

“I love you”. Simple rules and regular expressions can assist solve these cases.

D. Unigram feature

The existence of single words or tokens in the text is considered the most common features of text classification since we take single words from the training dataset and make a frequency distribution of these words.

E. TF-IDF technique

The term frequency – inverse document frequency also called (TF-IDF), is a famous method to estimate the importance of a word in a collection or document by contrasting it with text in an immense document corpus. Its way of evaluating the importance of any keyword in the context depends on how many times it appears in the documents. This method is based not only on using the frequency levels of a word, but also the comparative frequency level by using the reference document corpus which is a great advantage. Therefore this method of operation differentiates between the common words and the important ones. So there is no use to keep the common words and they can be deleted, while the words of importance can be kept by providing an appropriate threshold level [13].

$$tfidf(t,d,D)= tf(t,d) \times idf(t,D) \qquad (2).$$

Term Frequency (TF) and the Inverted Document Frequency (IDF) are calculated using (2) and (3).

$$tf(t,d)= \frac{f(t,d)}{count(w,d)} \qquad (3).$$

where f(t,d) indicates the number of appearance of word t in the document d and count(w,d) indicates the number of words in document d.

$$idf(t,D) \ = log \frac{N}{|\{d \in D: t \in d\}|} \qquad (4).$$

Where, N is the total documents contained in the reference corpus, and |{d ∈ D: t ∈ d}| is the number of documents which includes the word t. It is also common practice to modify the denominator by adding 1 to it, 1+|{d ∈ D: t ∈ d}|. This assists in deleting the divide by 0 mistakes, if the word is not included in documents. This stage leads to the identification of TF-IDF vectors for all important words in the document corpus. This technique has another advantage, which is the deletion of all the common text in the corpus, resulting in a great downsizing of the processing data [13]. TF-IDF calculation is carried out through three classes: HashingTF, IDF and IDFModel.

VI. Results

In order to test our proposed solution and compare it with approach proposed in [3]; Experiments were implemented on dataset from Twitter that has 1.578.627 classified tweets as in (V.B). Data is stored into the HDFS for analysis. Before putting the tweets under the microscope and evaluating them, the original dataset is divided into random segments of 5.000, 15.000, 25.000, 50.000, 100.000, 200.000 tweets, to be pre-processed for getting rid of any noise from the data and each tweet was changed into a vector of unigrams that indicates the frequencies of each word in the tweets using (TF-IDF) technique unlike [3], taking into account that the number of positive records and negative ones is similar in each segment .

It is obligatory to split the datasets into training and test sets in order to validate the trained model. The division for this work is carried out in the ratio of 3:1. i.e. 70% of the data is utilized for training and 30% of the data is separated for testing.

To ensure obtaining a precise accuracy of the classifier model, 5-fold cross-validation is implemented. the processes of training and testing continue in the execution till each part of the five testing set is used. We have 5 testing results, so we consider the average value as the accuracy of the model.

One of the evaluation metrics of the different algorithms that we use is F-Measure. According to (Table 1), it is obvious that Naive Bayes and Logistic Regression results are close and perform better than Decision Trees.It could be noticed that, the results obtained using our approach, gives better results than [3] when using the three classifiers.

Finally, we investigate the impact of adding multiple computing nodes to Spark Cluster in regards to running time. Three various cluster configurations are examined, each cluster composes of N ∈ {1,2,3} slave nodes are being checked each time. Fig 4 reveals that the total running time for three algorithms which were utilized in sentiment prediction for Datasets (100000, 200000) among the three scales of Spark cluster, It is clear to us that the total running time is going down by adding more nodes to the cluster.

Table 1 : F-Measure of segments

Dataset size	Naïve Bayes	Logistic Regression	Decision Trees
5000	0.69	0.68	0.63
15000	0.72	0.71	0.63
25000	0.74	0.73	0.65
50000	0.76	0.75	0.65
100000	0.77	0.77	0.67
200000	0.78	0.78	0.68

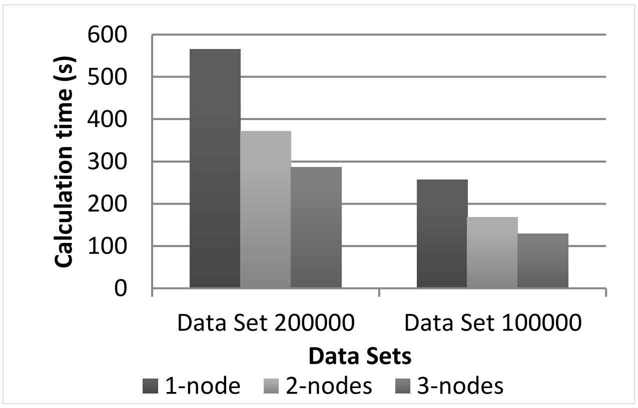

Fig. 4: CALCULATION TIM

VII. Conclusions and Future Work

Twitter Data in the form of reviews, thoughts, opinion, comments, feedback and grievance are treated as big data and it cannot be interpreted directly; it should be preprocessed in order to be suitable for mining tasks. In this research, we propose an effective sentiment prediction technique in Big Data, using Spark; the Apache Spark's Machine Learning library is utilized to execute all classification algorithms. The results indicate a significant enhancement in the accuracy of Naive Bayes and Logistic Regression with respect to increasing the volume of dataset, while the improvement is not strong in Decision Trees, also, experiment's results conclude that there is an inverse proportional relation between running time and the number of machines in the Spark Cluster, So in case of adding extra nodes in the cluster, higher performance capability will be obtained. From the former outcomes, our system can be described as effective and scalable.

For near future, we aim to examine the impact of different features on the input vector and use larger datasets. additionally, our goal is to establish an online service that gets benefits from the Spark Streaming which is considered as an Apache Spark's library for managing streams of data that gives the users with a real time predictions and analytics about sentiments of needed subjects.

References

[1] Goel, Ankur, Jyoti Gautam, and Sitesh Kumar. "Real time sentiment analysis of tweets using Naive Bayes." Next Generation Computing Technologies (NGCT), 2016 2nd International Conference on. IEEE, 2016.

[2] Parveen, Huma, and Shikha Pandey. "Sentiment analysis on Twitter Data-set using Naive Bayes algorithm." Applied and Theoretical Computing and Communication Technology (iCATccT), 2016 2nd International Conference on. IEEE, 2016.

[3] Baltas, Alexandros, Andreas Kanavos, and Athanasios K. Tsakalidis. "An apache spark implementation for sentiment analysis on twitter data." International Workshop of Algorithmic Aspects of Cloud Computing. Springer, Cham, 2016.

[4] Yan, Bo, et al. "Microblog Sentiment Classification Using Parallel SVM in Apache Spark." Big Data (BigData Congress), 2017 IEEE International Congress on. IEEE, 2017.

[5] Pang, B., Lee, L.: Opinion mining and sentiment analysis. Found. Trends Inf. Retrieval 2(1–2), 1–135 (2008) .

[6] Medhat, Walaa, Ahmed Hassan, and Hoda Korashy. "Sentiment analysis algorithms and applications: A survey." Ain Shams Engineering Journal 5.4 (2014): 1093-1113.

[7] Yang, Peng, and Yunfang Chen. "A survey on sentiment analysis by using machine learning methods." Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2017 IEEE 2nd Information. IEEE, 2017.

[8] Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, 2002.

[9] Wang, Hao, et al. "A system for real-time twitter sentiment analysis of 2012 us presidential election cycle." Proceedings of the ACL 2012 System Demonstrations. Association for Computational Linguistics, 2012.

[10] Bindal, Nimit, and Niladri Chatterjee. "A Two-Step Method for Sentiment Analysis of Tweets." 2016 International Conference on Information Technology (ICIT). IEEE, 2016.

[11] Nodarakis, Nikolaos, et al. "Large Scale Sentiment Analysis on Twitter with Spark." EDBT/ICDT Workshops. 2016.

[12] Assiri, Adel, Ahmed Emam, and Hmood Al-Dossari. "Real-time sentiment analysis of Saudi dialect tweets using SPARK." Big Data (Big Data), 2016 IEEE International Conference on. IEEE, 2016 .

[13] Nirmal, V. Jude, and DI George Amalarethinam. "Real-Time Sentiment Prediction on Streaming Social Network Data Using In-Memory Processing." Computing and Communication Technologies (WCCCT), 2017 World Congress on. IEEE, 2017.

[14] Chikersal, Prerna, Soujanya Poria, and Erik Cambria. "SeNTU: sentiment analysis of tweets by combining a rule-based classifier with supervised learning." Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). 2015.

[15] Sehgal, Divya, and Ambuj Kumar Agarwal. "Sentiment analysis of big data applications using Twitter Data with the help of HADOOP framework." System Modeling & Advancement in Research Trends (SMART), International Conference. IEEE, 2016.

[16] Zaharia, Matei, et al. "Spark: Cluster computing with working sets." HotCloud 10.10-10 (2010): 95.

[17] Desai, Mitali, and Mayuri A. Mehta. "Techniques for sentiment analysis of Twitter data: A comprehensive survey." Computing, Communication and Automation (ICCCA), 2016 International Conference on. IEEE, 2016.

[18] Apache Spark. Available online: http://spark.apache.org.

[19] Hamstra, Mark, and Matei Zaharia. Learning Spark: lightning-fast big data analytics. O'Reilly & Associates, 2013.

[20] Twitter Sentiment Analysis Training Corpus (Dataset). Available online:http://thinknook.com/Twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/

[21] Bhadani, Abhay Kumar, and Dhanya Jothimani. "Big Data: Challenges, Opportunities, and Realities." Effective Big Data Management and Opportunities for Implementation. IGI Global, 2016. 1-24.

[22] Bhadani, Abhay Kumar, and Dhanya Jothimani. "Big Data: Challenges, Opportunities, and Realities." Effective Big Data Management and Opportunities for Implementation. IGI Global, 2016. 1-24.

[23] Perwej, Yusuf, et al. "An Empirical Exploration of the Yarn in Big Data."

[24] Hammad, Mustafa, and Mouhammd Al-awadi. "Sentiment analysis for arabic reviews in social networks using machine learning." Information Technology: New Generations. Springer, Cham, 2016. 131-139.