International Journal of Industrial Engineering & Production Research September 2020 Vol. 31, No. 3: 343-350
DOI: 10.22068/ijiepr.31.3.343

RESEARCH PAPER

Effective Sentiment Analysis of Twitter with Apache Spark

K. V. K. Sasikanth[*1], K. Samatha[2], N. Deshai[3], B. V. D. S. Sekhar[4] & S. Venkatramana[5]

Received 3 May 2020; Revised 24 August 2020; Accepted 9 September 2020; Published online 30 September 2020

ABSTRACT

Today's interconnected world generates a huge amount of digital data while millions of users share their opinions and feelings on various topics through popular applications such as social media, different micro blogging sites, and various review websites every day. Nowadays, applying sentiment analysis to Twitter data is regarded as a considerable problem, particularly for various organizations or companies who seek to know customers' feelings and opinions about their products and services. The nature, variety, and enormous size of the data make it considerably practical for several applications ranging from choice and decision making to product assessment. Tweets are being used to convey the sentiment of a tweeter on a specific topic. Those companies keep surveying millions of tweets on some kinds of subjects to evaluate actual opinions and know the customers' feelings. This paper aims to significantly collect, recognize, filter, reduce, and analyze all such relevant opinions, emotions, and feelings of people on different products or services which could be categorized into positive, negative, or neutral because such categorization improves sales growth of a company's products, films, etc. The Naïve Bayes classifier is the mainly utilized machine learning method for mining feelings from a large quantity of data, like twitter and other popular social networks, due to its higher accuracy rates. This study performs sentiment polarity analysis on Twitter data in a distributed environment, known as Apache Spark.

KEYWORDS: *Big Data; Machine Learning; SVM; Map Reduce Spark Framework; Naïve Bayes; Sentiment Analysis; Natural language processing.*

1. Introduction

Sentiment Analysis (SA) aims to recognize the relationships between words in different sentences relatively rather than simply finding words through a search engine. Nowadays, people easily extensively share their experiences, thoughts, opinions, and feelings through popular social networks and on different professional micro blogging platforms, specifically Twitter as the most fashionable platform [1]. Today, the digital world extremely dominates the spread of encapsulated news and different trending topics across the globe at a rapid velocity. Most popular micro blogging platforms, like Twitter which was established in 2006 by Jack Dorsey, Noah Glass, Biz Stone, and Evan Williams, have recently become more popular social media across the whole world.

Approximately 6,000 tweets are tweeted on average on Twitter each second, more than 350,000 tweet posts per minute, around 500 million tweets per day, and approximately 200 billion tweets every year. More than 321 million active users are using Twitter due to the widespread dominance and horizon of the Internet [2].

Twitter is known as the most popular micro blog that offers an amazing social group service and mainly supports customers while interacting through messages called "tweets". Tweets could be primarily restricted to 140 characters only; however, in 2017, this boundary was increased by two-fold to 280 for many languages except Chinese, Japanese, and Korean. While all subscribed customers could post, like, and re-tweet required tweets, but nonregistered customers can simply read them [3].

* **Corresponding author:** ***K. V. K. Sasikanth***
sasikanth@giet.ac.in

1. *Department of CSE, GITE, Rajahmundry, A.P, India*
2. *Department of CSE, JNTUK, Kakinada, A.P, India*
3. *Department of IT, SRKREC, Bhimavaram, A.P, India*
4. *Department of IT, SRKREC, Bhimavaram, A.P, India*
5. *Department of IT, SRKREC, Bhimavaram, A.P, India*

2. Related Work

In the recent decade, , there has been a rising debate on Sentiment Analysis (SA) and emotional evolution, which is essential to the modern world applications. In addition, a large amount of data is being daily generated, which is more accessible over the WWW, particularly by those who reveal customers' feelings, incidents, sentiments, and opinions. SA is performed in multiple dissimilar stages using an unsupervised learning technique, which can be used to analyze reviews, thus facilitating the document-level data categorization. This technique is more successful in the word and sentence stage, where societal opinions are analyzed [4]. Furthermore, the intensity of used expressions is measured by discovering a neutral stance or polarity of expressions. To this end, machine learning methods are commonly employed. Naive Bayes (NB) classifier, Maximum Entropy, and SVM were also employed to examine the customers' feelings about products and film critiques. In this regard, classification standards mine the sentiment out of multilingual web documents.

3. Sentiment and Emotion Detection

NB job outperforms other machine learning methods in detecting sentiments, feelings, and emotion derived from a document. Hence, to facilitate this analysis, the Naïve Bayes classifier was employed to classify tweets into their equivalent feeling and sentiment classes. NB can also perform well on condition that the features are mutually dependent when every function is assumed to be autonomous in order to effectively measure the contingent probability [5]. This naive presumption could provide a decent trade-off between computation and output expenses. In this regard, it really is possible to interpret and clarify the resulting system. In addition, because of its intrinsic regularization, NB, much the same way as a generative classifier, might be suggested for tiny sample dimensions, increasing the chance of overfitting than allowed to discriminate classifiers.

Consequently, NB simply allows operating relatively well for issues with strong individual specific words and straightforward interactions between text features and their corresponding classes, e.g., for simple shapes of promotional information identification and two-class feeling categorization with strong polarization.

4. Naive Bayes

Naive Bayes is a pure multiclass model through Bayes' theorem. Every possible difficulty is primarily described as a feature vector, which points out that the value of each characteristic is independent of that of the remaining features. This algorithm is beneficial in that it is trained in an extremely efficient fashion, thus requiring only a single attempt at forming the entire training data. Further, the conditional probability distribution of every feature class is determined and then, Bayes theorem is utilized to predict the class label of every instance.

Tab. 1. Sample document with Customer's Tweets

DNO	Text	Class
1	I liked this Product	+
2	I Hated this Product	-
3	A great Product, Good Product	+
4	Poor Quality	-
5	A great Product, Good Quality	+

Tab. 2. Document with Positive and Negative Words

DNO	I	Liked	This	Product	Hated
1	1	1	1	1	1
2	1		1	1	
3				2	
4					
5					

Tab. 3. Document with Positive and Negative Words

DNO	A	Great	Poor	Quality	Good	Class
1	1	1	1	1	1	
2	1		1	1		
3				2		1
4						
5						1

Tab. 4. Document with Positive Words

DNO	I	Liked	This	Product	Hated	Class
1	1	1	1	1	1	+
2				2		+
3						+

Tab. 5. Document with Positive Words

DNO	A	Great	Poor	Quality	Good	Class
1						+
2	1	1			1	+
3		1		1	1	+

For this project, SA is applied to determine the approach of a speaker or a writer in terms of subject or the whole relative polarization of a text using Naive Bayes algorithm [6]. An example is given here to simplify the problem. There are five documents that contain movie reviews, three of which are positive and two are negative. There are a total of ten unique words in these documents. Here, all the unique words are listed out with their frequency in each document, as shown in Table 1.

Table 1 shows a document with 5 customers' tweets with a total of 10 unique words including I, Liked, This, Product, A, Great, Hated, Good, Quality, and Poor. Finally, this document is evaluated based on the number of unique words and the frequency of each word.

Let's look into the probability of a positive outcome. it comprises documents with positive reviews. Thus, the probability of achieving a positive value is three to five, e.g., 0.6. Then, the probability of each word being positive needs to be calculated using the NK formula, which is the frequency of the word happening N times, which is the total number of + ve words or -ve words as the total number of unique words while testing. in the case of the occurrence of an unknown word, NK equals 0 and its probability can be either +ve or -ve. The following steps demonstrate how to calculate the probabilities using neighbor approach.

At positive =3/5=0.6, the document is evaluated step by step using Naïve Bayes algorithm:

Step1:

P(I/+);P(Liked/+);P(This/+);P(Product/+);P(Hated/+);P(A/+);P(Great/+);P(Poor/+);P(Quality/+);P(Good/+);P(Class/+);

Step 2: P (WK/+) = (Nk+1)/n+|vocabulary|

Step 3: Nk: How many times the word K is seen in these cases (+)

Step 4: N: How many words are in (+) case: 14

Step 5: Vocabulary: the overall distinctive Words

Step 6: P (WK/+) = (nk+1)/n+|Vocabulary|

However, for unidentified words, nk=0 and its probability being either +ve or -ve is determined.

The flowchart of the entire project is given below.

Step 1: Training the classifier by training data with labels that are either positive or negative depending upon the review.

Step 2: Determining the test classifier

Step 3: Finding the sentiment about a given sentence or a word.

Now, this section presents the mining time in the trained classifier. *Stop words* represent a list of neutral words like nouns, articles, prepositions, etc. These stop words are eliminated and terms regarding the label are kept. Reviews are divided into +ve and -ve and the probabilities of every word occurring and dumping into the test classifier with pickle file are determined. This study presents input as the test data again and removes the stop words from the testing data for each of the remaining words. An attempt is made to verify whether the word is included in the training data; if included, the related probability should be calculated which could be either positive or negative, and if not, the probability is computed based on a Naïve Bayes Algorithm [7]. By repeating this process for all words, these probabilities are added individually for both positive and negative reviews along with prior probabilities. Now, if the positive probability is higher, then, it is positive, otherwise negative.

5. Sentiment Analysis with Apache Spark

Sentiment Analysis (SA) is described as the process of obtaining techniques for the identification and extraction of data from unstructured data through NLP and text analysis. The study of emotions has been frequently used in different projects that facilitates decision-making, where it allows making the right choice by gathering the thoughts of diffident people through their feedback and comments. Of example, on the basis of customers' feedback, several retailers sell the latest products or develop their new ones [8]. It is also used to assess the

assumptions of politicians in the elections based on an individual's feelings. The following paragraphs address the essential principles relevant to the study of emotions.

6. Spark-based Sentiment Analysis

The uniqueness of a large data quantity introduces the latest difficulties to sentiment analysis. Generally, the massive size, variety, and velocity of data are generated from various digital world applications. This is why more requirements turn out to be huge demands to utilize advanced big data and Hadoop frameworks for effective sentiment analysis [23,24,25]. Different types of research suggested improving the effectiveness of sentiment analysis in extracting people's feelings using large data frameworks. In the recent researches, based on machine learning, lexicon-based and hybrid methods are used. In machine learning methods including Naïve Bayes, support vector machine classifier is applied. Among these methods, the Naïve Bayes was the most frequently used technique due to its higher precision than other methods [26,27,28]. The scalability on Naïve Bayes to efficiently handle huge twitter datasets was evaluated within the Spark framework. Spark Streaming is an extension of the heart of Spark API that facilitates more scalable, large throughput, significant fault-tolerant stream processing across the digital world application live-data streams. Spark Streaming could be applied to a stream of live data and processing with more real times. While it arrives in Real-Time Analytics across the world datasets, Spark Streaming offers a particular platform to ingest data to significantly achieve quick and live processing [29]. Data Streaming is an essential technique for transporting data so that it could be prepared as a constant and endless stream. Streaming technologies are used to convert major leading and more demand with the extension of the digital world. Now, Apache Spark Streaming can be used to stream out the real-time data on multiple applications like Twitter, Stock Market, and Geographical Systems and play great analytics to the success of markets.

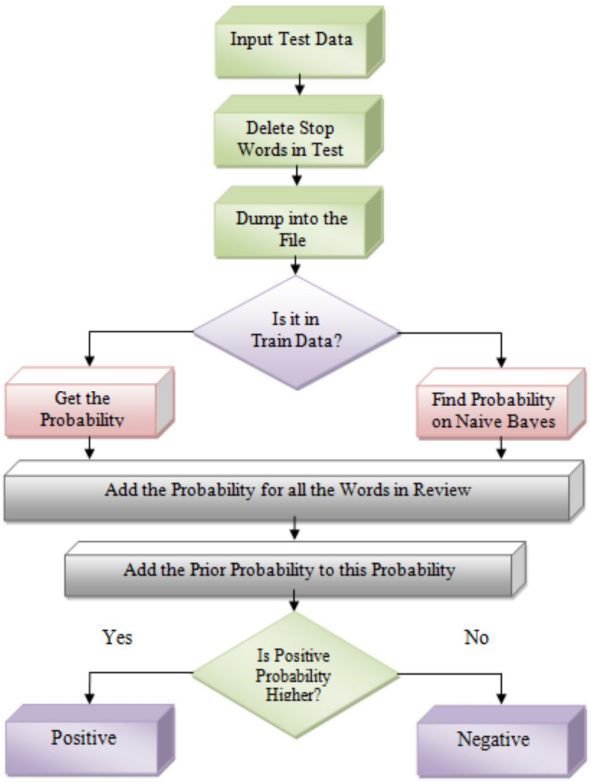

Fig. 2. Training Naïve Bayes Classifier

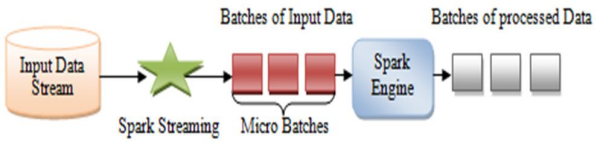

Fig. 3. Apache spark stream processing

The fundamental stream element is a stream which is mostly a sequence of RDDs to efficiently process the real-time data across the digital world applications, as shown in Figure 3.

Spark Streaming Features

Scaling: The ability to simply increase nodes to 100 in number and beyond.

Speed: attained with low latency.

Fault Tolerance: High capability to powerfully recover from failures.

Integration: Spark used to easily combine batch and micro-batch with real-time processing applications.

Business Analysis: Tracing the activities of each customer in the market [30].

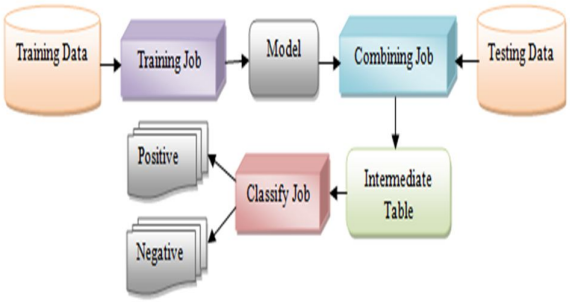

Fig. 4. Naïve bayes evaluation system on apache spark

The main objective here is to extrapolate the users' feelings by evaluating their tweet texts. In Fig. 4, the Naive Bayes Evolution System is applied to handle the world's large Twitter dataset and easily understand the consumers' feelings and opinions via taking some essential actions for an efficient sentimental analysis. Initially, twitter data collection is done; then, some preprocessing aspects are performed; the training data are stored for training jobs. Next, the Naïve Bayes classifier model is built based on top of Apache spark framework with high scalability [9,10,11,12]. The process involves combining jobs, finding each job accurately with the best classifier model, and preparing the testing data for identifying the best classifier based on precision and sentiments like positive and negative opinions [31]. The involved components include Work Flow Controller (WFC), compiler of content, terminals of the customer, and locator of the output. The rates of these components are obtained as follows: first, the WFC is used to develop the framework to conduct the learning job, as shown in Fig. 4. A combiner is used to integrate the testing data according to the standards, resulting in intermediate outcomes. The reports became eventually listed by measuring the probability of every report. The overall precision of the entire system is about 82%.

Variety of Naive Bayes (NB) and Support Vector Machines (SVM) are commonly seen as basic classification strategies; yet, their efficiency varies greatly based on the framework configuration, functionality used throughout different jobs, and repositories [13, 14, 15].

The results demonstrate that (I) the addition of the term token features offers consistent performance on sentiment analysis activities; (ii) for short fragment sentiment functions, NB is outperforming SVMs again (while the reverse results persist with lengthy files); (iii) a basic, yet modern, SVM variant uses NB log-count proportion as the feature level performs consistently well in both functions and databases [16, 17, 18, 19]. MNB is generally superior to and steadier than the multivariate Bernoulli NB and, gradually, produces more recognized outcomes; the binary MNB is rather more improved than normal MNB.

For the SVM, x (k) = ˆf (k) and w, b are achieved by reducing

wT w + C X I max(0, 1 − y (i) (wTˆf (i) + b))2

The L2-regularized L2-loss SVM were found highly reliable and the L1-loss SVM less stable. For lengthy texts, it was found that interpolation by MNB and SVM would be excellently performed for all documents and results are obtained and reported using this model:

$w0 = (1 - \beta)\ \bar{w} + \beta w$

where $\bar{w} = \|w\|1/|V|$ is the mean importance of w, and $\beta \in [0, 1]$ is the insertion parameter. This insertion can be seen as a model of regularization: consider NB as more reliable than SVM.

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

$$P(c|X) = P(x_1|c) * P(x_2|c) * \ldots * P(x_n|c) * P(c)$$

$$c^* = \arg\max_c P(c|d)$$

P (c|x) Posterior Probability.
P (x|c) Likelihood.
P (c) Class Prior Probability.
P (x) Predictor Prior Probability.

7. Experimental Outcome

The main objective of the testing is to evaluate the accuracy rate of popular MLlib's Naive Bayes and support vector machine to determine the viability of each classifier [21, 22, 23, 24, 25]. Accuracy is an excellent measure of determining which classifier represents the best to handle large databases. The precision metric is given as follows: *AAAAAAAAAAAAAAA* = (*TTTT* + *TTTT*)/ (*TTTT* + *TTTT* + *FFFF*+ *FFFF*), where TP, TN, FP, and FN are true +, true -, false +, and false -, respectively. The estimated outcomes are shown in Table 2. As shown in Table 2, the most excellent presentation was accomplished with Support Vector machine, followed by good results obtained from Naïve Bayes classifier and logistic regression classifier.

In the area of machine learning and particularly the issue of numerical identification, a confusion matrix, also recognized as an error matrix, is a particular table structure that represents the output of an algorithm. Generally, a supervised learning model is regarded as a match matrix in the unsupervised learning. While every row of the matrix describes the condition and status of a projected class, each column describes that of an actual class, and vice versa [21, 26, 27 ,28,29, 30, 31, 32, 33]. The name "confusion" suggests that the method can be misrepresented into two categories (i.e., generally misrepresenting one class as the other).

$$classify(f1, \ldots, fn) = \arg\max p(C = c) \prod_{i=1}^{n} p(Fi = fi \backslash C = c)$$

Tab. 6. Evolution Result

Classifier	Accuracy
Naïve Bayes	87%
Support Vector machine	85%

8. Conclusion

This study presented new insights by introducing various classified techniques, i.e., Naïve Bayes and Support vector machine, to characterize the opinions of large-scale data utilizing Spark's MLlib. Able to deal with a large quantity of data, Apache Spark Machine Library (MLlib) was employed in this study. The tests were performed using the dataset of Amazon reviews comprising four million ratings, 50,000,000 training reviews, and 40,000 test feedbacks. Several pre-processing procedures were implemented to prepare the information for classification. Two classifiers, Naïve Bayes and Support Vector Machine, were measured in terms of effectiveness. Test results revealed that the Naive Bayes classifier outperformed other classifiers.

Cotter, N.E., Guillerm, T.J., " The CMAC and a Theorem of Kolmogorov," Neural Networks, Vol. 5, 1992, pp. 221-228.

References

[1] Kaur, H., V. Mangat, V., Nidhi, N., "A Survey of Sentiment Analysis techniques," International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), (2017), pp. 921-925.

[2] Liu, B., Blasch, E., Chen, Y., Shen, D., Chen, G., "Scalable Sentiment Classification for Big Data Analysis Using Naïve Bayes Classifier," in IEEE International Conference on Big Data, USA, (2013).

[3] Edison, M., Aloysius, A., "Concepts and Methods of Sentiment Analysis on Big Data," International Journal of Innovative Research in Science, Engineering and Technology, Vol. 5, No. 9, (2016), pp. 16288-16296.

[4] Patel, P., Prabhu, G., Bhowmick, K., "A Survey of Opinion Mining and Sentiment Analysis," International Journal of Computer Applications, Vol. 131, No. 1, (2015), pp. 24-27- 2015.

[5] Madani, Y., Bengourram, J., Erritali, M., Hssina, B., Birjali, M., "Adaptive e-learning using Genetic Algorithm and Sentiments Analysis in a Big Data System," (IJACSA)

International Journal of Advanced Computer Science and Applications, Vol. 8, No. 8, (2017), pp. 394-403.

[6] Biltawi, M., Etaiwi, W., Tedmori, S., Hudaib, A., Awajan, A., "Sentiment Classification Techniques For Arabic Language: A Survey," in 7th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, (2016).

[7] Madani, Y., Mohammed, E., Jamaa, B.,"A Parallel Semantic Sentiment Analysis," in 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), Rabat, (2017).

[8] Appel, O., Chiclana, F., Carter, J., Fujita, H., "A hybrid approach to the sentiment analysis problem at the sentence level," Knowledge-Based Systems, (2016), pp. 110-124.

[9] Biltawi, M.,Al-Naymat, G., Tedmori, S., "Arabic Sentiment Classification: A Hybrid Approach," in International Conference on New Trends in Computing Sciences, Amman, Jordan, (2017).

[10] Huq, M., Ali, A., Rahman, A., "Sentiment analysis on Twitter data using KNN and SVM," (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 6, (2017), pp. 19-25.

[11] Jianqiang, Z., Xiaolin, G., "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis," in IEEE Access, (2017).

[12] Madani, Y., Erritali, M., Bengourram, J.,"Sentiment analysis using semantic similarity and Hadoop MapReduce," Knowledge and Information Systems, (2018), pp. 1-24.

[13] Haddia, E., Liu, X., Shi, Y., "The role of text pre-processing in sentiment analysis," in Information Technology and Quantitative Management (ITQM2013), (2013).

[14] Hassan, S., Miriam, F., Yulan, H., Harith, A., "On stopwords, filtering and data sparsity for sentiment analysis of Twitter," LREC, Ninth International Conference on Language Resources and Evaluation, (2014), pp. 810-817.

[15] Singh, T., Kumari, M., "Role of Text Pre-processing in Twitter Sentiment Analysis," Procedia Computer Science, (2016), pp. 549-554.

[16] Sharif, W., Samsudin, N., Deris, M., Nase, R., "Effect of Negation in Sentiment Analysis," in Sixth International Conference on Innovative Computing Technology (INTECH), Dublin, Ireland, (2016).

[17] Saif, H., He, Y., Fernandez, M., Alani, H., "Semantic Patterns for Sentiment Analysis of Twitter," in International Semantic Web Conference, (2014).

[18] Tartir, T., Abdul-Nabi, I., "Semantic Sentiment Analysis in Arabic Social Media," in Journal of King Saud University – Computer and Information Sciences, (2017).

[19] Biltawi, M., Etaiwi, W., Tedmori, S., Shaout, A., "Fuzzy based Sentiment Classification in the Arabic Language," in Intelligent Systems Conference, London, UK, (2018).

[20] Chauhan, V., Shukla, A., "Sentimental Analysis of Social Networks using MapReduce and Big Data Technologies," IJCSN International Journal of Computer Science and Network, Vol. 6, No. 2, (2017), pp. 120-130.

[21] Parveen, H., Pandey, S.,"Sentiment Analysis on Twitter Data-set using Naive Bayes Algorithm," in 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Bangalore, India, (2016).

[22] Ramesh, R., Divya, G., Divya, D., Kurian, M., Vishnuprabha, V., "Big Data Sentiment Analysis using Hadoop," IJIRST – International Journal for Innovative Research in Science & Technology, Vol. 1, No. 1, (2015), pp. 92-96.

[23] Deshai, N., "Performance and Cost Evolution of Dynamic Increase Hadoop Workloads of Various Data centers," Springer Nature Singapore, Vol. 105, No. 1, (2019), pp. 505-516.

[24] Deshai, N.,"A cross study on apache hadoop and yarn," International Journal of Engineering & Technology, Vol. 7, No. 4, (2018), pp.4850-4855.

[25] Deshai, N.,"Study with Comparing Big Data Handling Techniques using Apache Hadoop MapReduce Vs Apache Spark," International Journal of Engineering & Technology,Vol. 7, No. 4, (2018), pp. 4839-4843.

[26] Deshai, N.," Big Data Challenges and Analytics Processing Over Health Prescriptions" Jour of Adv Research in Dynamical & Control Systems, Vol. 15, No. 1, (2017), pp. 650-657.

[27] Deshai, N.,"Big Data Hadoop MapReduce Job Scheduling: A Short Survey", Information Systems Design and Intelligent Applications, (2019), pp. 349-365.

[28] Deshai, N.,"A Study on Big Data Hadoop Map Reduce Job Scheduling," International Journal of Engineering & Technology, Vol. 7, No. 3, (2017), pp. 59-65.

[29] Deshai, N., "An advanced comparison on big data world computing frameworks", Journal of Physics: Conference Series,Vol. 1228, No. 1, (2019), pp. 12003-12011.

[30] Deshai, N.,"MLlib: Machine Learning in Apache Spark," International Journal of Recent Technology and Engineering (IJRTE),Vol. 8, No. 1S3, (2019), pp. 45-49.

[31] Deshai, N.,"Protect Internet from Intrusion with Advanced Spark Framework, International Journal of Recent Technology and Engineering (IJRTE),Vol. 8, No. 1S3, (2019), pp. 186-190.

[32] Deshai, N., "Processing Real-World Datasets Using Apache Hadoop Tools," International Journal of Recent Technology and Engineering (IJRTE), Vol. 8, No. 1S3, (2019), pp. 209-213.

[33] Deshai, N.,"A Study on Big Data Processing Frameworks: Spark and Storm," Springer, Vol. 160, (2019), pp. 415-424.

[34] Zhou, K., Doyle, J.C., & Glover, K., Robust and optimal control, Prentice-Hall, Englewood cliffs, NJ, (1996).

Follow This Article at The Following Site:

Sasikanth K, Samatha K, Deshai N, Sekhar B V D S, Venkatramana S. Effective Sentiment Analysis on Twitter with Apache Spark. IJIEPR. 2020; 31 (3) :343-350
URL: http://ijiepr.iust.ac.ir/article-1-1067-en.html