International Journal of Open Source Software and Processes
Volume 10 • Issue 4 • October-December 2019

A Novel Approach to Optimize the Performance of Hadoop Frameworks for Sentiment Analysis

Guru Prasad, SDMIT, Ujire, India
Amith K. Jain, SDMIT, Ujire, India
Prithviraj Jain, SDMIT, Ujire, India
Nagesh H. R., A.J. Institute of Engineering and Technology, Mangalore, India

ABSTRACT

Twitter is one among most popular micro blogging services with millions of active users. It is a hub of massive collection of data arriving from various sources. In Twitter, users most often express their views, opinions, thoughts, emotions or feelings about a particular topic, product or service, of their interest, choice or concern. This makes twitter a hub of gargantuan amount of data, and at the same time a useful platform in getting to know and understand the underlying sentiment behind a particular product or for that matter anything expressed in twitter as tweets. It is important to note here that aforesaid massive collection of data is not just any redundant data, but one which contains useful information as noted earlier. In view of aforesaid context, Sentiment analysis in relation to twitter data gains enormous importance. Sentiment analysis offers itself as a good approach in classifying the opinions formulated by individuals (tweeters) into different sentiments such as, positive, negative, or neutral. Implementing Sentiment analysis algorithms using conventional tools leads to high computation time, and thus are less effective. Hence, there is a need for state-of-the-art tools and techniques to be developed for sentiment analysis making it the need of the hour to facilitate faster computation. An Apache Hadoop framework is one such option that supports distributed data computing and has been commonly adopted for a variety of use-cases. In this article, the author identifies factors affecting the performance of sentiment analysis algorithms based on Hadoop framework and proposes an approach for optimizing the performance of sentiment analysis. The experimental results depict the potential of the proposed approach.

KEYWORDS

Big Data, Data Compression, Hadoop, Hadoop Distributed File System, MapReduce, Sentiment Analysis, Twitter, YARN

1. INTRODUCTION

In today's digital world social networking sites play a vital role and also have an influential say in modern way of life. Twitter is one among the most popular social networking sites with more than 100 million of daily active users. According to Statista survey, as of year 2017 Twitter had 328 million active users and the number is said to have increased and still increasing day by day (Andreas et al.,2017). In Twitter, registered users can read and post tweets; tweets are limited to 280 characters. They can also upload images and short videos of size not more than 5MB and 512MB respectively.

DOI: 10.4018/IJOSSP.2019100103

Millions of users express their views, opinions, thoughts, emotions, feelings about different products, events, people, etc., on the twitter platform.

Indian Premier League (IPL) is a popular, professional Twenty-Twenty (T20) cricket league played in India. It ranks sixth among all sports leagues across the world. As we already know cricket in India is not just viewed as a sport, but, a religion in itself. Due to its humungous popularity, unending reach along with an uncanny ability to arouse interest and then being able to follow it up with definite action, it evokes all sorts of emotions, feelings and what not among cricket viewers. The same goes true for IPL, its fans, and in general, viewers of IPL. In Twitter, IPL fans originating from various places express their views, opinions, thoughts, emotions or feelings about their favorite IPL teams and players. During IPL season millions of tweets get tweeted every day on a regular basis. Aforesaid live stream of data is considered to be a rich source of information for Sentiment analysis. Natural Language processing is used to mine people's opinions about IPL teams and players expressed in form of tweets. Sentiment analysis helps in classifying people's opinions as positive, negative or neutral Implementing Sentiment analysis algorithms using traditional data analytics tools seem unable to handle Twitter Big Data as data to be handled is humongous, changing at a fast pace and characteristically complex by nature. Big data analytics has modernized traditional data analytics by introducing new technologies that support distributed storage and processing of large amount of data. Today, Apache Hadoop has become a highly popular and powerful distributed computing framework to process large amounts of data. It is composed of Hadoop Distributed File System (HDFS), Yet Another Resource Negotiator (YARN) and MapReduce parallel programming model. The unique features of Hadoop that make it so attractive are ease of access, robustness, fault tolerance, scalability and ease of parallel programming. Using Hadoop framework, a lot of work has already been proposed on Sentiment analysis in relation to Twitter data. However, some parameters affecting the performance of Sentiment analysis remain a challenge on Hadoop framework. When working with large amounts of data sets, there will be challenges and difficulties such as data sets consuming more HDFS disk space, network related issues and high computation time. In this paper, the author identifies the factors affecting the performance of sentiment analysis algorithm based on Hadoop framework and proposes an approach for optimizing the performance of sentiment analysis. Experimental results obtained show that proposed novel approach effectively optimizes the HDFS disk space utilization, speeds up the data movement in the network and optimizes the computation time.

The rest of the paper is organized as follows: Section 2 comprises of literature survey in relation to the proposed topic; Section 3 presents the proposed framework and associated implementation so as to optimize the performance of sentiment analysis with regard to Twitter data; Section 4 substantiates aforesaid analysis by showcasing comprehensive experimental results; Finally, Section 5 delivers conclusion to the paper.

2. LITERATURE SURVEY

Andreas et al. (2017), has presented that Sentiment Analysis of Twitter data is certainly a challenging problem due to the sheer amount of volume, velocity and variety associated with the same. Sentiment analysis of aforesaid large quantity of information offers extensive potential in terms of sentiments present in this information leaning towards specific topics. Most of the existing algorithms with respect to sentiment analysis are limited to centralized computing platforms, and hence can handle at the most a few thousand tweets. This kind of computing platform is not fully representative when it comes to finding sentiment polarity regarding a specific topic, owing to the fact that huge number of tweets are being posted daily. In aforesaid paper, the authors developed two modules, MapReduce and Apache Spark framework for the purpose of Big Data Programming. Authors implemented Sentiment Analysis technique using Machine Learning algorithms and utilized Apache Spark framework. The proposed systems were trained to collect real-time Twitter Data and process the same in a distributed

manner. The experimental results show the quality of sentiment identification as compared with the same in conventional solutions.

Diamantini et al. (2017), has explained that Traditional sentiment analysis based on lexicon technique falters in identifying the right negation as they fail to make use of available efficient methods to identify the right negation window. In aforesaid paper, authors have addressed issues of instinctive resolve of scope of negation and then proposed dependency-oriented parse tree to identify negation. The proposed work is built upon proper utilization of semantical relations that exist between terms necessary for framing of a meaningful statement, and thus emphasis is put on finding out terms which are tormented by negation. Furthermore, aforesaid technique has been combined with semantic disambiguation technique so that sentiment associated with a statement is properly recognized. Based on experimental results obtained on various sample sets, it is found that quality of aforesaid analysis is enhanced in the proposed technique. In future works, authors have planned to advance the correctness of the negation handling approach by determining more features, namely, effect of various conjunctions, or the usage of punctuation marks. Along with this, authors have also planned to address the problem of use of emoticons in sentences, which are being enormously used in social networks, specifically on Twitter.

Araque et al. (2017), has expressed that Sentiment Analysis is in dire need of deep learning techniques so as to facilitate automatic feature extraction, enabling of richer representation capabilities, along with offering better performance than conventional feature-based techniques (i.e., surface methods). In this paper, the authors developed an enhanced, deep learning-oriented Sentiment Analysis system based on a word's embedding method, along with linear machine learning approach. Several experiments were conducted wherein performance of presented technique was analyzed with the baseline deep learning technique, and experimental results depict that the performance of the proposed model is enhanced than the baseline model.

Clavel, et al. (2016), has expressed that Sentiment Analysis has seen an enormous surge in interest owing to the availability of huge amount of Social network data. It is also expressed that Sentiment analysis is instrumental to and is being pursued by lot of emerging research areas. Development of Embodied Conversational Agents (ECA) to interact with humans is one among them. The human-agent interaction communities and opinion mining enthusiasts are presently addressing sentiment analysis from completely different views that consists of, on the one hand, disparate sentiment-related phenomena and procedure representations, and on the opposite side, various detection and dialog management ways. In aforesaid paper, the authors identified, and later discussed the upcoming challenges in multidisciplinary fields. Authors proposed different potentialities for mutual profit, specifying a lot of analysis tracks and presenting open queries and prospects. To conduct proposed experiment, job interviews and conversation with regular visitors who attend museums have been considered as test cases.

Doan et al. (2017) described that most of the times people associated with business regularly want to know customer's opinion regarding standard of their service, so as to improve and hence, increase profit. Sentiment analysis of customer reviews holds and plays important effect on a business's improvement techniques. Sentiment analysis based on offline solutions comprise of training data being collected beforehand, and model being built later. This leads towards model being trained again and again. To avoid retraining of complete model from time to time, incremental learning offers itself as the best alternate solution and is the need for proposed task. In this paper, the authors proposed an alternate online random forests algorithm to accomplish sentiment analysis with regards to respective customers' reviews. The proposed work has been able to achieve more accuracy in comparison to traditional works.

Jose et al. (2015) proposed a Sentiment Analysis system to predict Delhi assembly election result through collection and analysis of twitter data. The author presented that sentiment analysis is the computational learning of information present in opinions, emotions, feelings, attitudes, and views, expressed in the form of text. It denotes a classification problem with major emphasis put on being able

to forecast the polarity of words and later classify them as positive, negative or a neutral sentiment. In Twitter, users express their views, opinions, thoughts, emotions, feeling, etc., towards variety of topics, including political party and politicians. In this paper, the authors introduced Sentiment Classifier using Word Sense Disambiguation (WSD), based on lexicon technique. The proposed system is implemented in the data pre-processing step using a negation handling to achieve more accuracy and hence proper classification of tweets as positive, negative or neutral. Experimental outcomes depict that the proposed approach attained high accuracy. However, limitation of the proposed work lies in the fact that collected data size was way too small to come at a meaningful conclusion.

Bharti et al. (2016), has explained that Sarcasm is the kind of sentiment where people express their negative emotions using positive words. While talking, people generally use heavy tonal stress and certain gestural intimations like moving of the eyes, hand movement, and so forth to reveal sarcasm. In textual data, these tonal and gestural intimations of information are missing, making sarcasm recognition exceptionally difficult for a normal human. Because of these difficulties, researchers have shown interest in sarcasm identification of social media text, particularly in tweets. In aforesaid paper, the authors presented a sarcastic sentiment detection model based on the Hadoop framework. Proposed model captured real-time tweets and analyzed it with a set of various approaches, such as TCTDF, TCUF, IWS, PSWAP, PBLGA, and LDC to identify sarcastic sentiment effectively. Aforesaid approaches were implemented with and without Hadoop framework. The experimental result shows that the elapsed time for analyzing and processing twitter data is more significant in the Hadoop framework as compared to conventional methods.

Cunha et al. (2015) described that social media advancements coupled with exponential increase in volume and complexity of data that are being generated by Internet services have made analysis difficult not only technologically, but also in view of emerging trends. In this paper, the authors proposed an all-purpose functional architecture based on Hadoop MapReduce platform and Mahout for storing and distributed processing of large data that can be applied in various situations. To prove its potential, strength, advantages, and applications, authors considered Twitter data related to health as a case study. Experimental outcomes of data analysis on Twitter health data demonstrated the potential of the proposed architecture.

3. FRAMEWORK AND IMPLEMENTATION

In the proposed work, a novel architecture to optimize the performance of sentiment analysis of Twitter data has been proposed. The proposed architecture i.e., SHOC (Sentiment analyzing Hadoop framework Optimized through Compressing data) is as shown in Figure 1, which expands the baseline of Hadoop framework.

Detailed step by step description of the SHOC is as follows.

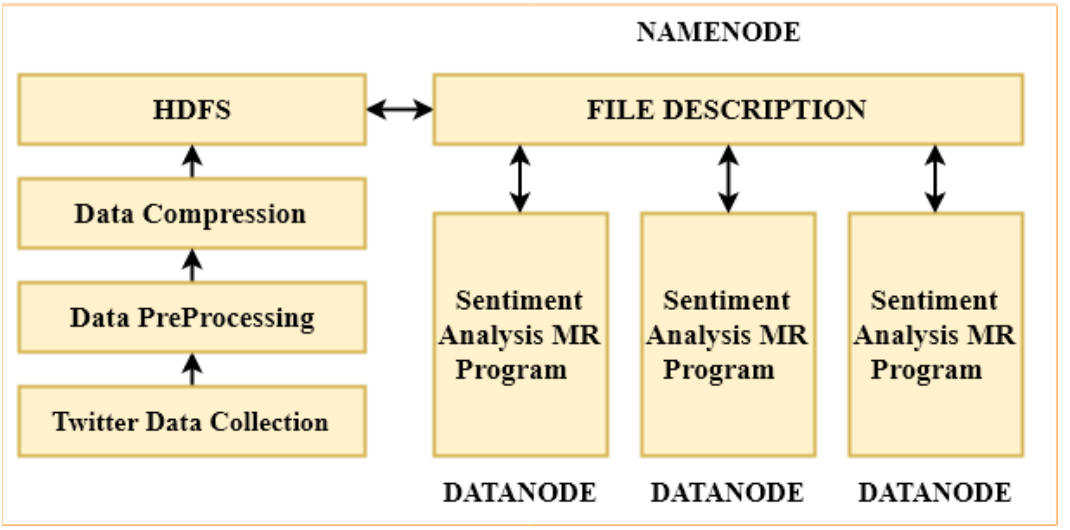
3.1. Twitter Data Collection

Using Twitter API, one can collect live data directly from twitter. In order to access such live Twitter data, the developer has to create a Twitter application, and the same has been created. In the proposed work, Twitter data concerning Indian Premier League (IPL)-2017 has been collected in JSON format for the duration 12-March-2017 to 22-May-2017.

3.2. Data Pre-Processing

The raw data so collected from Twitter is often inconsistent, sometimes irrelevant and may comprise of noisy tweets. Hence, data needs to be pre-processed. Data pre-processing is a process that involves transforming raw twitter data into an understandable format. It involves removing URL and hashtags, spelling corrections, and replacing emoticons.

Figure 1. SHOC architecture



3.3. Removing URL and Hashtags

While tweeting, sometimes a user mentions another user as @username and shares some useful information using URL and hashtags. Most of the times, the URL and hashtags which come along with the information hold little value to the recipient, and thus can be treated redundant. The proposed system eliminates such redundant information in connection with sentiment analysis of tweets like user information, time, date, URL, hashtags, duplicates of tweets, and hence reduces memory consumption.

3.4. Spelling Correction

Users, sometimes knowingly or unknowingly misspell the words, resulting in same word being written in so many variations such as good being written as gud, gooood, gd and so on. The problem with aforesaid variation in representations is that identification of said word for all its appearances becomes extremely difficult. Hence, spelling correction will be of great help for effective analysis.

3.5. Replacing Emoticons

Users sometimes use emoticons to express their views, emotions, and feelings. Emoticons play a major role in sentiment analysis. For efficient sentiment analysis, we can replace these emoticons by respective appropriate words.

3.6. Data Compression

The large data volumes that exist in a typical Hadoop framework demands and makes compression very essential. Data compression will definitely save large amount of storage space and is guaranteed to speed up the movement of data, throughout the Hadoop cluster. Data compression techniques that are supported by Hadoop frameworks are as follows.

3.6.1. Gzip

Gzip is a compression utility used for the purpose of file compression and decompression. It is based on the DEFLATE algorithm, which is a combination of LZ77 coding and Huffman Coding. The Gzip file format comprises of:

1. A 10-byte header providing version number and timestamp;
2. Optional extra headers which provide original file name;
3. A body, containing a DEFLATE-compressed payload;
4. An 8-byte footer, containing a CRC-32 checksum, along with length of the original uncompressed data.

By default, Gzip is supported in Hadoop ecosystem. The file extension of Gzip compressed file is .gz. Gzip does not support file splitting and parallel processing of files.

3.6.2. Bzip2

Bzip2 is an open source data compression technique based on Burrows-Wheeler algorithm. It provides a high degree of compression, with its downfall being low compression speed. Bzip2, as such is naturally supported in Hadoop ecosystem. The file extension of Bzip2 compressed file is .bz2. It supports file splitting and parallel file processing.

3.6.3. Lzo

Lzo is one such data compression technique which provides us with a modest degree of compression, along with high compression speed. Lzo, as such is naturally supported in Hadoop ecosystem. The file extension of Lzo compressed file is lzo. It supports file splitting and parallel file processing.

3.7. Sentiment Analysis

Sentiment analysis has been carried out by implementing SentiWordNet approach in MapReduce parallel programming model. SentiWordNet is a lexical resource for sentiment analysis and is one of the most popular approaches used for sentiment analysis. Aforesaid approach classifies the tweets as positive, negative or neutral, based on the context. The pseudo code of sentiment analysis is as shown below.

Algorithm 1. Sentiment analysis

```
Begin
• sentiment←0 (If there is no word related to a particular
sentiment in current tweet, then)
• ps←0 // positive sentiment
• ns←0 // negative sentiment
• nus←0 //neutral sentiment
• while there are words related to a particular sentiment in
current tweet, then
• if positive word is present in relation to the context, then
ps++
• else, if negative word is present in relation to the context,
then ns++
• else, if neutral word is present in relation to the context,
then nus++
• end while
• positive percentage=(ps/(ps+ns+nus)) *100
• negative percentage=(ns/(ps+ns+nus)) *100
• neutral percentage=(nus/(ps+ns+nus)) *100
• sentiment = (ps-ns)/(ps+ns)
End
```

4. RESULTS AND DISCUSSION

The Performance Analysis of Twitter data using Hadoop MapReduce framework with respect to data compression, time taken to load data from local file system to HDFS System, and data process time was initially benchmarked with original Hadoop and then compared with results obtained using Proposed Compression Approaches i.e. Gzip, Bzip2, Lzo.

4.1. Experimental Environment

Performance analysis is carried out on a test platform which contains Hadoop five node cluster with homogeneous hardware property, i.e.., Each node in the cluster has a 3.8 GB RAM, Intel® Core i5 3470 CPU @3.20GHz * 4 processor. A cluster has been set up on Ubuntu 16.03 with Hadoop 2.6.5 stable release using oracle jdk 1.8 and ssh configuration to manage Hadoop daemons. The cluster setup comprises of 1 NameNode and 5 DataNodes as part of the experiment. Configuration files such as mapred-site.xml, core-site.xml, hdfs-site.xml and yarn-site.xml are setup with default values, i.e.., replication factor of 2 and data block size of 128 MB.

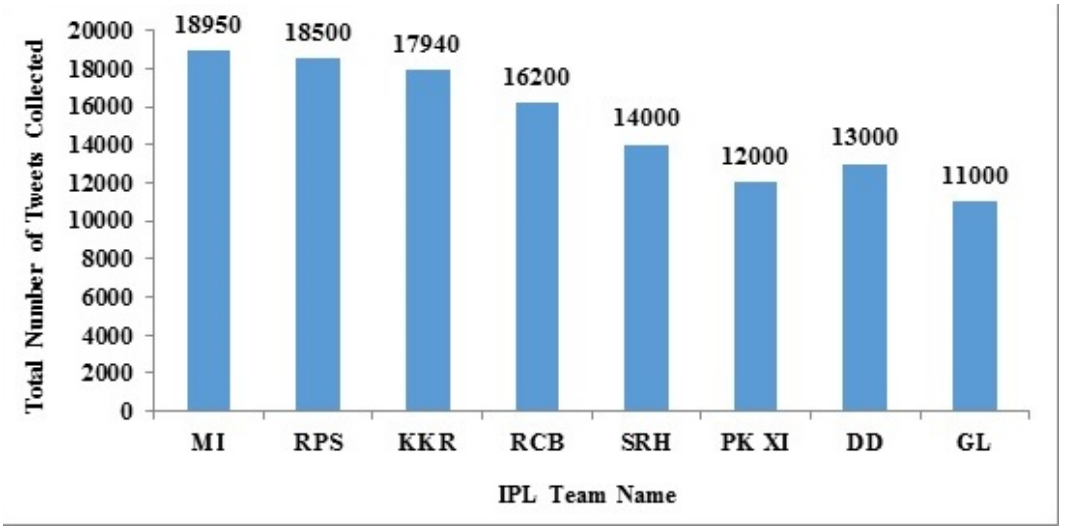
4.2. Sentiment Analysis

The total number of tweets collected for the purpose of sentiment analysis were 18,950 for Mumbai Indians (MI), 18,500 for Rising Pune Supergiant (RPS), 17,940 for Kolkata Knight Riders (KKR), 16,200 for Royal Challengers Bangalore (RCB), 14,000 for Sun Risers Hyderabad (SRH), 12,000 for Kings XI Punjab (PK XI), 13,000 for Delhi Daredevils (DD) and 11,000 for Gujarat Lions (GL). The same is plotted in Figure 2.

The so collected tweets were analyzed using SHOC. SHOC classifies the tweets as positive, negative or neutral, based on the context of the tweet. Total percentage of positive tweets, negative tweets and neutral tweets are calculated using Equations (1), (2) and (3) respectively. Table 1 demonstrates the percentage of Positive, Negative and Neutral Tweets with respect to IPL teams. Figure 3 depicts a graphical representation of data presented in Table 1:

$$P_{PT} = T_{PT} / T_{T} \tag{1}$$

Figure 2. Number of tweets collected in relation to IPL teams



$$P_{NT} = T_{NT} / T_T \tag{2}$$

$$P_{NUT} = T_{NUT} / T_T \tag{3}$$

where:

PPT = Percentage of Positive Tweets
PNT = Percentage of Negative Tweets
PNUT = Percentage of Neutral Tweets
TT = Total Number of Tweets
TPT = Total Number of Positive Tweets
TNT = Total Number of Negative Tweets
TNUT = Total Number of Neutral Tweets

4.3. Data Compression

The large data volumes that exists in a typical Hadoop framework demands and makes compression very essential. Data compression will definitely save large amount of storage space, and is guaranteed to speed up the movement of data, throughout the Hadoop cluster. Data compression techniques that are supported by Hadoop frameworks are Gzip, Bzip2, and Lzo.

Experiments were conducted to compress 10 GB of twitter data using SHOC compression approaches i.e. Gzip, Bzip2, Lzo. Table 2 demonstrates the total time taken by proposed approaches to compress files along with compressed file size. Figure 4 depicts a graphical representation of total time taken by the proposed approaches to compress files and Figure 5 depicts a graphical representation of the compressed file size, upon compression.

4.4. Data Uploading

To process the input data, user needs to upload the same from local disk to HDFS. Proposed Approach compresses the input data and uploads the compressed data to HDFS. The formula to calculate data uploading time is as follows:

Table 1. Percentage of positive, negative and neutral tweets with respect to IPL teams

IPL Team	P_{PT} (%)	P_{NT} (%)	P_{NUT} (%)
MI	80.88	12.21	6.91
RPS	78.96	12.94	8.10
KKR	78.62	17.99	3.39
RCB	63.06	33.00	3.94
SRH	69.01	26.05	4.94
PK XI	60.14	32.93	6.93
DD	76.79	20.95	2.26
GL	55.98	38.25	5.77

Figure 3. Percentage of positive, negative and neutral tweets with respect to IPL teams

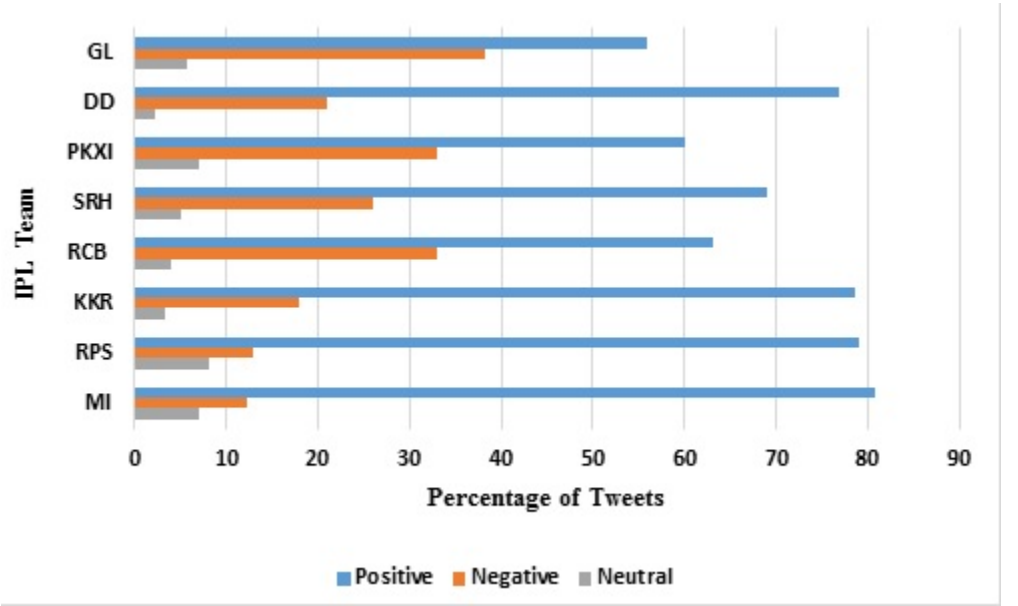


Table 2. Total time taken by proposed approaches to compress files along with compressed file size

Technique	Original File Size in GB	Total Time Taken to Compress File in Seconds	Compressed File Size in GB
Gzip	10	690	2.2
Bzip2	10	1305	1.4
Lzo	10	285	3.9

Figure 4. Total time taken by proposed approaches to compress files

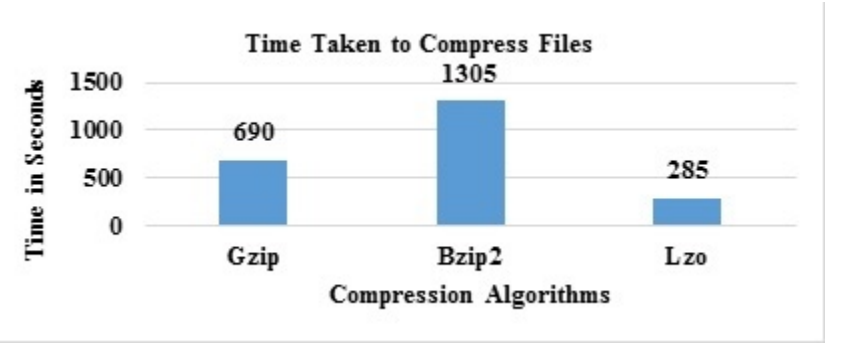
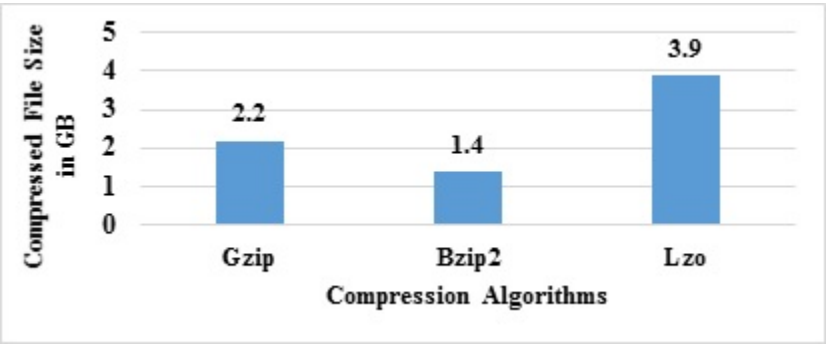


Figure 5. Compressed file size in proposed approaches



$$T_{DU}^{Total} = T_{CF}^{Total} + T_{MF}^{Total} \quad (4)$$

where:

T_{DU}^{Total} = Total time taken for data uploading

T_{CF}^{Total} = Total time taken to compress files

T_{MF}^{Total} = Total time taken to move compressed files to HDFS

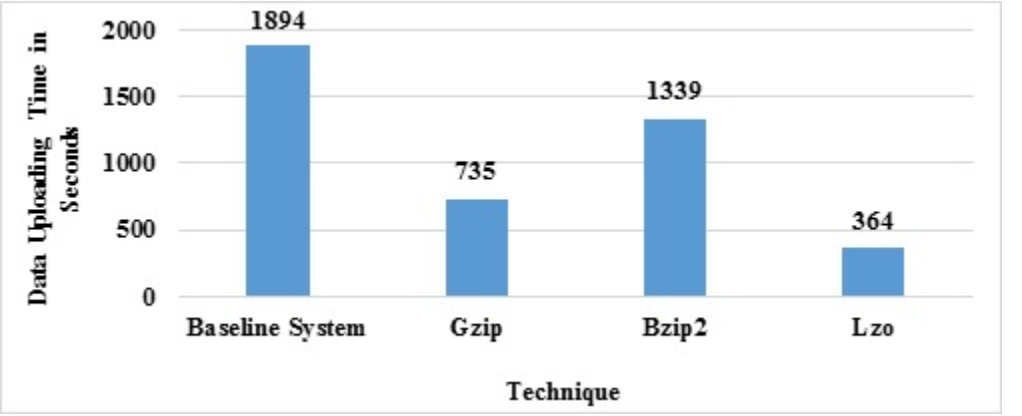
Experiments were conducted to test the data uploading time in Baseline System (default Hadoop) and Proposed Compression Approaches i.e. Gzip, Bzip2, Lzo. Table 3 demonstrates the time taken to upload data from local disk to HDFS by Baseline System and the Proposed Approaches. Figure 6 depicts a graphical representation of data presented in Table 3. With reference to Table 3, time required for Sentiment Analysis in Proposed Approaches i.e. Gzip, Bzip2, Lzo is optimized than in Baseline System by 61.19%, 29.30% and 80.78% respectively. The result obtained clearly indicates that the performance of Proposed Approaches is better than in the Baseline System.

Table 3. Comparative analysis of total time taken to upload data from local disk to the HDFS

Technique	T_{CF}^{Total} in Seconds	T_{MF}^{Total} in Seconds	T_{DU}^{Total} in Seconds	% Time Optimization
Baseline System	-	1894	1894	61.19%
Gzip	690	45	735	
Baseline System	-	1894	1894	29.30%
Bzip2	1305	34	1339	
Baseline System	-	1894	1894	80.78%
Lzo	285	79	364	

"-" relates to the fact that Baseline System does not compress files

Figure 6. Comparative analysis of total time taken to upload data from local disk to HDFS



4.5. Data Processing Time

Experiments were conducted to compute total time taken for Sentiment analysis in the Baseline System and Proposed Compression Approaches, i.e. Gzip, Bzip2, Lzo. Later, the same has been compared. The total time taken for Sentiment analysis can be calculated using Equation (5):

$$T_{SA}^{Total} = T_{m}^{Total} + T_{r}^{Total} \tag{5}$$

where:

T_{SA}^{Total} = Total time taken for Sentiment analysis

T_{m}^{Total} = Total time required for execution of map phase

T_{r}^{Total} = Total time required for execution of reduce phase

Table 4. Comparison of total time taken by baseline system with proposed approaches in relation to sentiment analysis

Technique	T_{m}^{Total} in Seconds	T_{r}^{Total} in Seconds	T_{SA}^{Total} in Seconds	% Time Optimization in Proposed Approaches When Compared With Baseline System
Baseline System	1165	148	1313	56.43%
Gzip	520	52	572	
Baseline System	1165	148	1313	69.38%
Bzip2	385	17	402	
Baseline System	1165	148	1313	47.90%
Lzo	608	76	684	

Figure 7. Comparison of total time taken to execute sentiment analysis by baseline system with proposed approaches

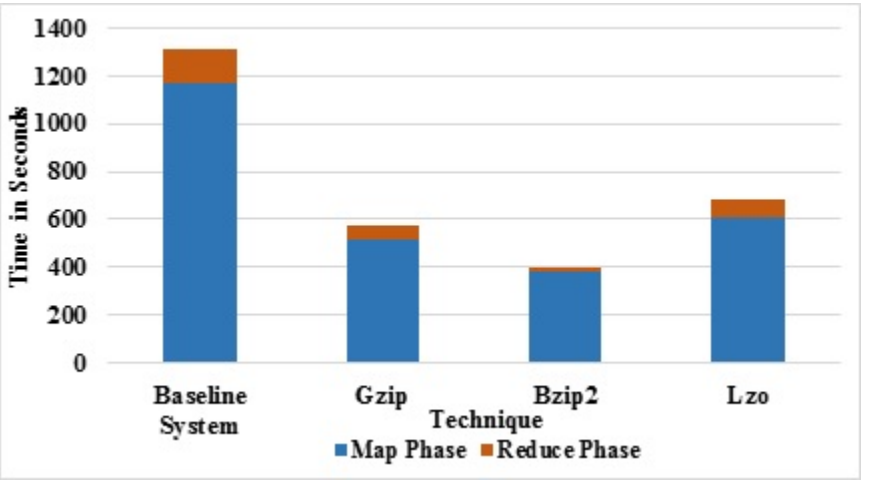


Table 4 demonstrates comparison of total time taken by Baseline System and Proposed Approaches in relation to Sentiment analysis. Figure 7 depicts the data presented in Table 4 in form of a chart. With reference to Table 4, time required for Sentiment Analysis in Proposed Approaches i.e. Gzip, Bzip2, Lzo is optimized than in Baseline System by 56.43%, 69.38% and 47.90% respectively. The result obtained clearly indicates that performance of Proposed Approaches is better than the Baseline System.

4.6. Overall Performance

Overall performance is defined as total time taken to complete the Sentiment analysis job. Experiments were conducted to compute the overall performance of Baseline System and the Proposed Compression Approaches i.e. Gzip, Bzip2, Lzo. Later, the same has been compared. The total time taken to complete Sentiment analysis job can be calculated using Equation (6):

$$T_{CJ}^{Total} = T_{DU}^{Total} + T_{SA}^{Total} \tag{6}$$

where:

T_{CJ}^{Total} = Total time taken to complete the Sentiment analysis job

T_{DU}^{Total} = Total time taken for data uploading

T_{SA}^{Total} = Total time taken for Sentiment analysis

Table 5 demonstrates the overall performance of Baseline System and Proposed Approaches. Figure 8 depicts a chart showcasing overall performance of Baseline System and Proposed Approaches. With reference to Table 5, overall performance of Sentiment Analysis using Proposed Approaches i.e. Gzip, Bzip2, Lzo is optimized than in Baseline System by 59.24%, 45.71% and 67.32% respectively. The result obtained clearly indicates that overall performance of Proposed Approaches is considerably optimized than in the Baseline System.

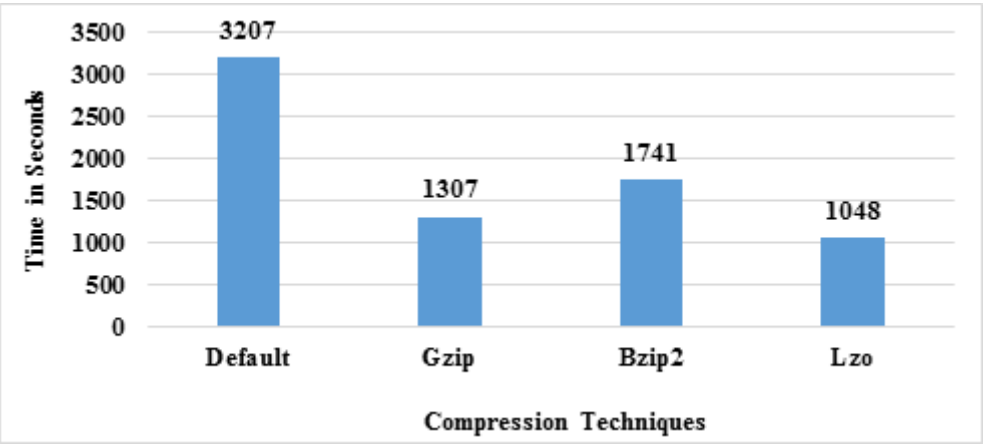
Table 5. Comparison of overall performance (with respect to sentiment analysis) of baseline system with proposed approaches

Technique	T_{DU}^{Total} in Seconds	T_{SA}^{Total} in Seconds	T_{CJ}^{Total} in Seconds	% Time Optimization in Proposed Approaches When Compared With Baseline System
Baseline System	1894	1313	3207	59.24%
Gzip	735	572	1307	
Baseline System	1894	1313	3207	45.71%
Bzip2	1339	402	1741	
Baseline System	1894	1313	3207	67.32%
Lzo	364	684	1048	

5. CONCLUSION

Twitter is one of the most popular microblogging platforms. Here, user's express their views, opinions, thoughts, emotions, feeling, etc., about any topic, product or service. Sentiment analysis is a good approach to classifying the opinions formulated by individuals into different sentiments such as, positive, negative, or neutral. Because of the exponential growth of Twitter data implementing sentiment analysis algorithms using traditional tools is not an effective way that results in high computation time. New distributed computing frameworks are required for faster computation. Presently, Hadoop is a popular distributed computing framework to process a large amount of data. With a comprehensive set of experiments, we identified that the parallel implementation of sentiment analysis algorithm in Hadoop framework is not best in terms of disk space utilization and execution time. In this paper, we proposed a novel approach to optimize the performance of Hadoop MapReduce framework for opinion mining. The experimental results depict that the proposed approach on Hadoop MapReduce framework provides the best performance in terms of execution time and disk space utilization as compared with the baseline Hadoop MapReduce framework.

Figure 8. Comparison of overall performance (with respect to sentiment analysis) of baseline system with proposed approaches



ACKNOWLEDGMENT

We would like to thank every member of the faculty at SDMIT, Ujire for their guidance and support, which has helped us, complete this research project successfully.

REFERENCES

Andreas, N. (2017). Nikolaos, D. Tsolis, and G. Tzimas, "Large Scale Implementations for Twitter Sentiment Classificatio. *MDPI Alogorithms*, *10*(1), 1–21.

Araque, I., Corcuera-Platas, I., Sánchez-Rada, J. F., & Iglesias, C. A. (2017). Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems With Applications*, *77*(1), 236–246. doi:10.1016/j.eswa.2017.02.002

Bharti, B., Vachha, B., Pradhan, R. K., Babu, K. S., & Jena, S. K. (2016). Sarcastic sentiment detection in tweets streamed in real time: A big data approach. *Elsevier Digital Communications and Networks*, *2*(3), 108–121. doi:10.1016/j.dcan.2016.06.002

Clavel, C., & Callejas, Z. (2016). Sentiment Analysis: From Opinion Mining to Human-Agent Interaction. *IEEE Transactions on Affective Computing*, *7*(1), 74–93. doi:10.1109/TAFFC.2015.2444846

Cunha, J., Silva, C., & Antunes, M. (2015). Health twitter big data management with Hadoop framework. *Procedia Computer Science*, *64*, 425–431.

Diamantini, C., Mircoli, A., & Potena, D. (2016, October). A negation handling technique for sentiment analysis. *Proceedings of the 2016 International Conference on Collaboration Technologies and Systems (CTS)* (pp. 188-195). IEEE.

Doan, T., & Kalita, J. (2016, December). Sentiment analysis of restaurant reviews on yelp with incremental learning. *Proceedings of the 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 697-700). IEEE.

Jose, R., & Chooralil, V. S. (2015, November). Prediction of election result by enhanced sentiment analysis on Twitter data using Word Sense Disambiguation. *Proceedings of the 2015 International Conference on Control Communication & Computing India (ICCC)* (pp. 638-641). IEEE.

Guru Prasad M S, Assistant professor, Department of Computer Science & Engineering. Shri Dharmasthala Manjunatheshwara Institute of Technology (S.D.M.I.T), Ujire, Dakshinna Kannada. He got his Ph.D. in Computer Science Engineering branch from the Visvesvaraya Technological University, Belagavi. He got his M.Tech (Computer Science & Engineering) from NMAMIT Nitte. He has published 4 research papers in International Journals, 6 research papers in International Conference and 1 research papers in national Conference. He has delivered 15 invited technical talks on "Big Data Analytics and Hadoop Ecosystem" at various reputed institutes. His interested area is Big Data Analytics and Distributed computing.

Amith K Jain received the B.E. degree in Electronics and Communication Engineering from AIT, Chikmagalur in the year 2009, and M.Tech in Microelectronics and Control Systems form DSCE, Bangalore in 2011. His subjects of interest include Image Processing, Communication Engineering and Data Mining.

Prithviraj Jain received the B.E. degree in Computer Science and Engineering from Visvesvaraya Technological University, Belagavi in the year 2011, and M. Tech in Computer Science and Engineering from Visvesvaraya Technological University, Belagavi in the year 2015. His subjects of interest include Big Data Analytics and Machine Learning.

Nagesh H.R. is currently working as Professor and Head, Department of Information Science & Engineering, Chief Coordinator for R&D activities and Coordinator for First year has obtained his B.Tech.(Computer Engineering) from NMAMIT, Nitte M.Tech. (Computer Engineering) and Ph. D (Computer Engineering) from National Institute of Technology, Karnataka Surathkal. He has got more than 23 years of experience in the field of teaching/ research. His subjects of interest are computer networks, distributed computing, Cloud Computing, Big Data Analytics, cryptography and network security, systems programming, object-oriented programming, etc., His field of specialization is Group Communication Security. He has published more than 70 research papers in National and International Conferences and journals. He has delivered more than 30 invited talks in topics like Component Based Software Development, Internet Security, Web Security, Web Engineering, Information Security, Network Management, Promoting Global Cyber Security, Security issues in Distributed Systems, Digital library and Information Search, Information Security Management, Recent Trends in Information Technology, Emerging Challenges and Solutions for Multimedia Data Security, Big Data & Internet of Things and Cloud Computing. He has also chaired many sessions in International and National level conferences. He has also published one chapter titled Proactive models for Mitigating Internet DoS/DDoS Attacks, in Selected Topics in Communication Networks and Distributed Systems, World Scientific, London, April 2010. He had also worked as visiting faculty to National Institute of Technology Karnataka Surathkal and NITK Science and Technology Entrepreneurs Park, Karnataka, Surathkal. Published two books titled Fundamentals of CMOS VLSI Design for V Semester Electronics & Communication Engineering students of VTU Pearson Education & VLSI Design for V Semester Electronics & Communication Engineering students of JNTU Pearson Education. Member of BOS for PG studies in Computer Science at Mangalore University and Manipal Institute of Technology for PG studies in Computer Science & Engineering. Worked as Member of BOE for UG studies in Computer Science & Engineering at VTU Belgaum. Worked as a Member of BOS in Computer Science & engineering at VTU Belgaum for the year 2013 to 2016. He has guided 3 Ph.D. scholars and currently guiding 7 Ph.D. scholars. Vision Group on Science and Technology (VGST), Department of Information Technology, Biotechnology & Science and Technology, Government of Karnataka has funded rupees 40 lakhs for his research project titled Technology Aided Agriculture Optimization.