Preprocessing Big Data for Efficient Storage and Research

Melbin J Reena, A. Shajin Nargun

Abstract--- Big Data refers to large datasets and so it is not possible to store, manage and analyze it using commonly used software systems. The emergence of smart phones, social networks and online applications has led to the generation of massive amounts of structured, unstructured and semi structured data. Big data analytics has received sizeable attention since it offers a great opportunity to uncover potentials from heavy amounts of data. Data preprocessing techniques, when applied prior to analytics, can substantially improve the overall quality of the patterns mined and/or the time required for the actual mining. Thus this paper presents an efficient method for preprocessing data and also partitioning big dataset based on sensitivity parameters. The partitioned dataset can be uploaded to public and private cloud based on the importance of data in the partition. Thus hybrid cloud storage and processing of big data is supported by this approach. The experimental results show that the proposed method preprocesses and partition data with high accuracy and reduced processing time.

Keywords--- Big Data Analytics, Preprocessing, Partitioning, Hybrid Cloud, Cloud Storage.

I. INTRODUCTION

Big data is the current trend and it refers to large volume of unstructured data that it is not possible to use traditional databases or software tools to manage and mine data. The big data domain includes categories such as big data science and big data frameworks. The term science refers to the study of things as they are, so as Big data science is the study of techniques dealing with the collection, cleaning, and mining of big data. Big data frameworks are software libraries and integrated tools and procedures that supports in the distributed processing and evaluation of big data by using large set of geographically separated computers. The mostly used big data processing frameworks are Hadoop, Spark, Flink, Storm, and Samza.

Big data analytics refers to the process in which valuable hidden data patterns are uncovered and presented as reports that helps in decision making and prediction. It necessitates the use of mining algorithms and strong supporting frameworks. Based on the time of processing after arrival of data, big data analytics can be categorized into two alternative paradigms: Streaming Processing, and Batch Processing. The streaming processing paradigm emphasizes data freshness and it analyzes data as soon as possible after generation. It assumes that the derived results are valid only if data is analyzed immediately after its arrival. In the batch-processing paradigm, data are first stored and then analyzed. MapReduce has become the dominant batch-processing model. The basic idea of MapReduce is that data are first divided into small chunks. Next these chunks are processed in parallel and in a distributed manner to generate intermediate results. The final result is derived by aggregating all the results.

The proposed work in this paper deals with partitioning data after preprocessing and thus supports MapReduce paradigm. The digital data life cycle of a big data system includes several functions to deal with different phases. Big data system life cycle includes four phases: Data generation, data acquisition, data storage, and data analysis. Data generation deals with the ways how large amount of data is generated from various domains. Data acquisition refers to the process of gathering information from disparate sources. Data acquisition includes data collection, data transmission, and data pre-processing. First, dedicated data collection technology is needed for gathering raw data that may come from specific data production environment. Second, a high-speed transmission mechanism is needed to move collected raw data to proper big data storage system for further analysis. Finally, collected datasets must be preprocessed to remove meaningless data, which simply waste the amount of storage space and affects the consequent data analysis. Data storage deals with permanently storing and managing large-scale datasets. A big data storage system includes hardware infrastructure and data management. Hardware infrastructure consists of a set of shared information and communication technology resources that are used to maintain and access big data on demand. Data management software is used along with hardware infrastructure to maintain large scale datasets. Data analysis consist of analytical methods or tools to inspect, transform, and model data to extract valuable information. The types of analytics are Structured data analytics, Text Analytics, Multimedia analytics, Web Analytics, Network analytics, and Mobile analytics. To perform any kind of analytics on big data, data must be preprocessed first to derive accurate results.

II. BIG DATA AND CLOUD

Cloud-based platforms are playing an increasingly significant role in big data analytics and storage applications. Usage of cloud computing ideas in big data analytics provides advantages such as scalability, flexibility, agility, energy efficiency, and cost effectiveness [8]. Enhanced security and privacy mechanisms must be in place to cope up with the rising need of correlating patterns from big data and for maintaining large-scale cloud infrastructures. The existing methods for securing data is applicable for only small-scale data and they doesn't suits well for big data.

Revised Manuscript Received on July 10, 2019.

Melbin J Reena, Research Scholar, Noorul Islam Centre for Higher Education. T.N, India.(e-mail: mjreena82@gmail.com)

Dr.A. Shajin Nargun, Director (Academics), Noorul Islam Centre for Higher Education, T.N, India. (e-mail: shajin@niuniv.com)

The cloud based environment is more vulnerable for security and privacy threats and need risk management procedures. The tremendous growth of cloud computing and cloud data stores is the major reason behind the emergence of big data.

Cloud computing has significant benefits over traditional deployment models. It saves computing time and resources by making use of standardized technologies. Cloud Service Providers come in all shapes and sizes and offer many different products for big data. Cloud computing provides enormous computing resources on demand and charges acquired based on usage only. Performance and Capacity are the two categories of cloud storage challenges in big data analytics. High performance platforms are necessary for managing and analyzing highly unstructured big data. Also, the cloud storage service for data analytics must be highly available, highly durable, and scalable in size. Thus cloud services become inevitable in Big Data analytics due to the large volume and variety of big data. All domains ranging from healthcare to e-commerce generates massive amounts of data frequently [11].

The three major categories of cloud deployment models developed over time are private cloud, public cloud and hybrid cloud. Private clouds are meant for one organization and suitable for organizations where data sensitivity in mainly concerned. As private cloud does not share physical resources secured storage and usage of data is facilitated. Also, the accidental or malicious access through shared resources is avoided.Public clouds share physical resources for data transfers, storage, and processing. Hybrid cloud may hold the answer for complicated big data analytics. As hybrid cloud is the combination of private and public cloud, user's sensitive information can be stored in private cloud and other data can be stored in public cloud. This hybrid model is more suitable for big data and it offers benefits like improved performance, cost effectiveness and enhanced security for businesses involved in big data analytics.

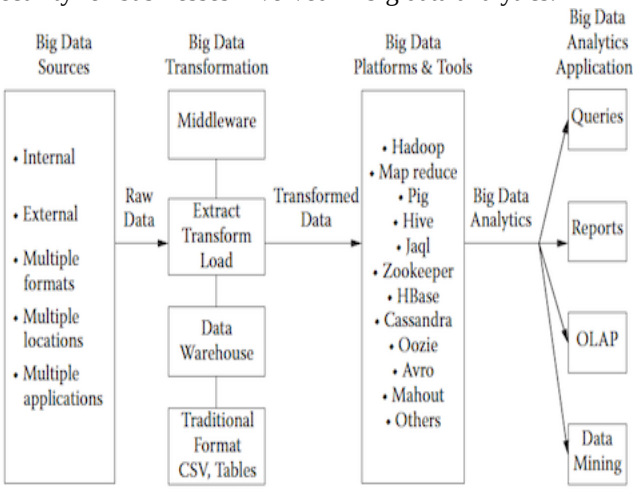

Figure 1: Big Data Analytics Framework

The varying nature of Big Data requires Elasticity and Infrastructure Flexibility. Through the use of hybrid cloud, down time and performance degradation due to increased workload can be minimized. Hybrid cloud model helps to allocate new workloads to powerful machines or will add more machines when there is a need for additional resources.In spite of all benefits of using cloud for big data processing, it should be noted that the transmission and remote storage of big data on cloud results in some problems as well. Securing tenants data, analyzing data, and sharing data only among authorized users remains as great challenge for cloud providers. In cloud computing environment, the tenants transfer and store their big data on cloud storage center directly, and may face some serious problems such as system crash or failure.

This paper proposes a preprocessing scheme, in which the big data of tenants will be divided into some smaller data blocks. These smaller data blocks will be stored in cloud storage media one by one, because these data blocks are smaller than the primitive big data, they are very efficient for remote-distance data transmission and storage. The proposed scheme mainly focuses on the storage and sharing for confidential big data in cloud.

III. DATA PREPROCESSING

Nowadays, big data analytics become more important because of its applications in data modeling, mining, querying, and distributing large scale repositories [13]. Big Data are usually generated by online transaction, video/audio, email, number of clicks, logs, posts, social network data, scientific data, remote access sensory data, mobile phones, and their applications. The data generated from various sources are aggregated in data stores that grow greatly in size. As the size of data stores is very large, it is complicated to accumulate, refine, analyze, manage, share, and visualize data using traditional database tools.

Big data collection gathers raw data from data production environment and the collected datasets might contain many meaningless data, which unnecessarily increases the amount of storage space and affects the consequent data analysis. Thus it is necessary to preprocess data for efficient storage and mining. Data preprocessing techniques, when applied prior to analytics, can reasonably improve the overall quality of the patterns mined and/considerably reduce the time required for mining.

Data preprocessing techniques when applied carefully can improve the quality of data, thereby helping to mine data accurately and efficiently. Preprocessing is an important step in the analysis process, since quality decisions must be based on quality data. Data collected from various sources may have quality issues in terms of noise, missing values, and outliers. Some applications might have strict requirements on data quality and such applications require efficient preprocessing techniques.So that data preprocessing techniques that are designed to improve data quality must be enforced effectively in big data systems.

The three main concepts of data preprocessing are data integration, data cleansing, and redundancy elimination. Data integration combines data residing in disparate sources and provide users with the unified view of the data. The basic steps in data integration are extraction, transformation, and loading. The extraction step selects and collects only the necessary data for analysis. The transformation process applies techniques to convert the extracted data into standard format.

The load step imports extracted and transformed data into target data storage from which data can be retrieved and analyzed in future.

The data cleansing technique is the process of determining incomplete, inaccurate, or unreasonable data and then to cleanse it by using appropriate schemes. Data cleansing search and find errors and corrects them for missing values, outliers, and noise. Data preprocessing must be carried out in order to obtain accurate results in analytics. However, data cleansing is a complex process and it incurs lot of computation cost and impose delay overhead. It must be needed to maintain balance between the complexity of the data cleansing model and the resulting improvement in the accuracy analysis [12].

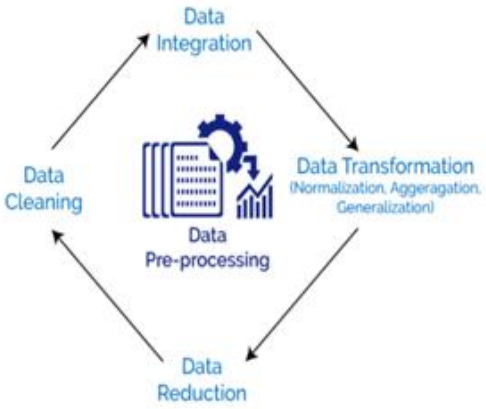

Figure 2: Data Preprocessing

Data redundancy is also an important issue in data preprocessing. An attribute can be considered as a redundant attribute if its value can be derived from another attribute. Storing data redundantly in data store merely waste valuable memory space. Redundancies among attributes can be identified by correlation analysis. The correlation between attributes A and B can be measured by

$$rA,B = \frac{\sum(A-\bar{A})\,(B-\bar{B})}{(n-1)\sigma A\sigma B}$$

Where n is the number of tuples, $\bar{A}$ and $\bar{B}$ are the respective mean values of A and B, and σA and σB are the respective standard deviations of A and B. if the result of the equation provides positive correlation, then one attribute can be removed as redundant attribute. In addition to detecting redundancies between attributes, duplication should also be detected at the tuple level [15].

However, no unified data preprocessing procedure and no single technique can be expected to work best across a wide variety of datasets. The attributes of datasets, the type of application, performance requirements, and other necessary features must be considered for selecting best preprocessing scheme. The proposed scheme preprocesses numerical and categorical data efficiently with reduced processing time and improved accuracy.

IV. RELATED WORK

Data cleansing and preparation efforts add significant overhead during analytics [17]. Xiaochun et al. proposed Range Aggregate Query approach that uses balanced partition algorithm for splitting big data into independent partitions with no overlapping of data, and then generates a local estimation sketch for each partition. The balanced partitioning algorithm works with a stratified sampling model [16]. This method splits data by considering attribute values of interest, whereas our proposed method divides big data by considering sensitivity parameter. Also, this work considers preprocessing of data before partitioning and making it ready for cloud upload. Muhammad et al. proposed real time analytical architecture for remote sensing application, which deals with dividing, load balancing, and parallel processing of only useful data. The filtration process separates useful data for analysis and discards incomplete and inaccurate data [13].

Xinhua et al. proposed a method for secure sensitive data sharing on a big data platform which needs sensitive data to be encrypted before cloud upload. For protecting user's sensitive data, Cheng et al. proposed a scheme in which big data is partitioned into sequential data parts according to certain principles such as same data type block or IP-resembled data packets. In the privacy-preserving public auditing approach proposed by Boyang et al. stated the importance of data partitioning in public auditing process of shared data in cloud [2]. [7] States that cloud users may also need to split big datasets into smaller datasets and store them in different physical servers for reliability, privacy-preserving or efficient processing purposes. Also, [12] states that partitioning datasets into smaller chunks helps in efficient scheduling. To better support security, auditing, scheduling, and hybrid cloud usage for big data processing, it is necessary to preprocess and partition big datasets [14]. According to [18], Shuffle and Chop are two data partitioning methods. Shuffle distributes data randomly, while Chop divides data by existing order. [19] adopts Shuffle method to partition data in their ensemble approach.

V. PREPROCESSING AND PARTITIONING

Preprocessing is necessary and it is the first step in any analytics. This paper proposes preprocessing and partitioning approach that can be well applied to big datasets consisting numerical and categorical data. The sample dataset from machine learning libraries are loaded in R. For preprocessing of numerical data, the mean value of every partition will be substituted for missing values. Also, the outliers will be identified and treated by considering mean and standard deviation. For preprocessing of categorical data, missing values will be filled by labels because most of the modeling tools neglect the records containing missing values. The dataset is partitioned into equal sized blocks before preprocessing and alternate blocks of data can be cleaned for missing values and outliers. The resultant value can be used for substitution in every alternate block thus reducing the amount of time and effort needed for processing entire dataset.

Partitioning of datasets can be performed by the user using partitioning techniques such as shuffle and chop [18]. In the shuffle method, the data items were selected one by one and assigned to different partitions randomly. Hence, it results in partitions containing equal amount of data.

One major concern about this approach is that the resultant partition is M-overlapping, so that a data item may appear in M partitions. Chop method partitions the datasets into equal sized blocks preserving the sequence of data. Data can be randomized before partitioning if needed; otherwise data is partitioned in the existing order.

In this proposed work, partitioning of big datasets is carried out by checking the sensitivity parameter mentioned in data dictionary. Every database available for experimentation is associated with data dictionary consisting description about data and their representation.

Parameters and variables:

S1, S2, S3,…SN are fixed size blocks of records. NT be the number of tuples/records. NCR be the number of columns in each record. BS is the block size of data.

If N is the number of sets/blocks, then N can be calculated as

$$N = \frac{NT \; X \; NCR}{BS}$$

Let SB be processed sample blocks and ASB be alternate unprocessed sample blocks.

SB: Processed Sample Block {B1, B3,B5….BN-1}

ASB: Unprocessed Sample Block{B2, B4, B6,……BN }

Let $\bar{X}$ Bi : Mean of sample values of block Bi, where i={1,2,3,4,…N}

$\bar{X}$ Bi = Sum of Values/ Size of Block

$$= \sum_{j=1}^{BS} \left(\frac{Vj}{BS}\right)$$

Vj: jth value of Block Bi, 1≤j≤BS

SDBi = $\sqrt{(\sum_{j=1}^{BS} vj - xBi)}$ /BS

Diff: Absolute difference between $\bar{X}$Bi and SDBi.

$$= | \bar{X} \text{ Bi} \; — \text{SDBi} |$$

Tval: Threshold value to check deviations from normal range.

NTval = Number of values in the block is greater than Tval.

If the number of records with outliers and missing values is negligible, that is less than five percent of block size, and then it can be simply discarded and not considered for analysis. The reason is that, in some cases the effort spent on cleaning data is more than performance gain after preprocessing data.

In the algorithm, Tval corresponds to threshold value to check deviations from normal range. It is calculated against each numeric value by considering mean and standard deviation. NTval specifies the number of values in the block that is greater than Tval and is used to decide whether to apply preprocessing algorithm or not.

Algorithm – Preprocessing

Input : Less structured Data, Data Dictionary

Output: Cleansed, Partitioned data blocks ready for cloud upload

1: Load dataset.
2: Read data dictionary for column names.
3: Assign names for each column.
4: Assign labels to categorical values.
5: Divide the dataset into fixed size (BS) blocks.
6: Make two samples of blocks so that only half of the part is processed.

SB = {B1, B3, B5….., BN-1}
ASB = {B2, B4, B6……, BN}

7: For each block Bi in SB, and for Numerical Values, Calculate

- $\bar{X}$ Bi (Mean)
- SDBi (Std Deviation)
- Diff
- NTval

8: If Diff>Tval, reset missing values and outliers with mean value.
9: Repeat the process (Steps 7 & 8) till all missing values and outliers are treated in Bi of SB.
10: For each block Bi in ASB, and for missing Numerical Values, substitute mean(Bi-1 and Bi+1)
11: Repeat the process till all missing values and outliers are treated in Bi of ASB.
12: Set the range for sensitivity parameters from data dictionary.
13: Parse the input block Bi into different partitions by the defined schema.
14: Assign Partition ID(PID) for data blocks.
15: Return partitioned set.

A fixed threshold value is considered to check number of records with outliers. If the number of records with outliers is less than threshold then it can be considered as negligible. Otherwise, the records with missing values and outliers must be treated. The mean value of numerical attributes of one block is used to treat missing values of the subsequent block. Thus, one half of the entire block set only need to be analyzed reducing the effort and time of processing to half.

After preprocessing, the entire dataset is divided into blocks based on the value of sensitivity parameter. Partition Id is also assigned to every partition in order to track it after cloud upload. Data preprocessing allows us to apply mining algorithms easier and quicker, obtaining more quality models in terms of accuracy and interpretability.

VI. RESULTS AND DISCUSSION

The Big Data preprocessing algorithm proposed in this work is evaluated with respect to accuracy and processing time. The preprocessed big data contains accurate values and so the analysis results of the dataset will also be accurate. For correcting missing values and outliers, only half of the big dataset is processed and the result is used in cleansing alternate blocks. Thus the preprocessing time is reduced to half. This type of preprocessing helps in batch processing of Big Data only, since data must be loaded entirely for preprocessing.

Table 1: Truth Table for Accuracy

TRUTH	FALSE	TRUE
Correct	22	158
Incorrect	264	14

Accuracy in preprocessing data can be calculated by:

$$\text{Accuracy} = \frac{(TP+TN)}{TP+TN+FP+FN}$$

The histogram obtained before preprocessing age factor shown in Fig.3 shows outliers, as the normal range of age in the dataset used varies from 20 to 70.

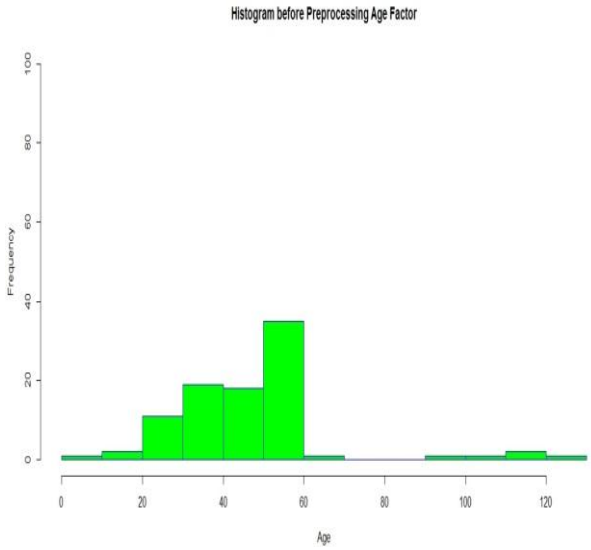

Figure 3: Histogram before preprocessing

After carrying out preprocessing of numerical data, the outliers are resolved and the resultant histogram consisting age range from 20 to 70 is shown in Fig.4.

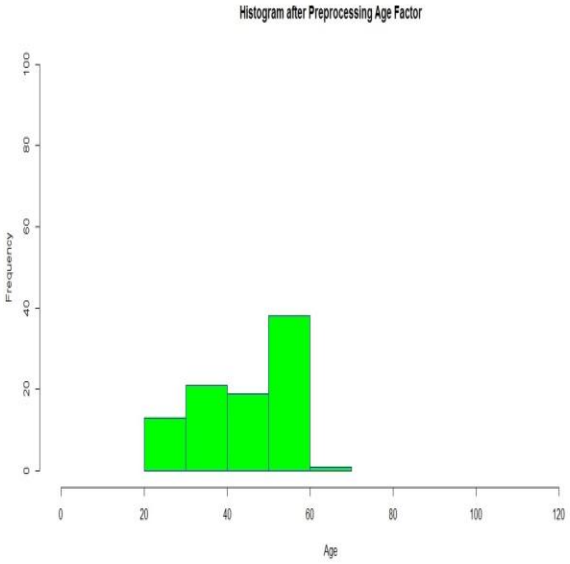

Figure 4: Histogram after preprocessing

The entire dataset is partitioned into blocks and every block's age factor is tested for accuracy before and after preprocessing. Comparative analysis of accuracy of age factor before and after preprocessing is shown in Fig.5.

Table 2: Accuracy Percentage

	Accuracy %	
Blocks	Before Preprocessing	After Preprocessing
B1	87	92
B2	89	95
B3	82	89
B4	90	96
B5	89	93

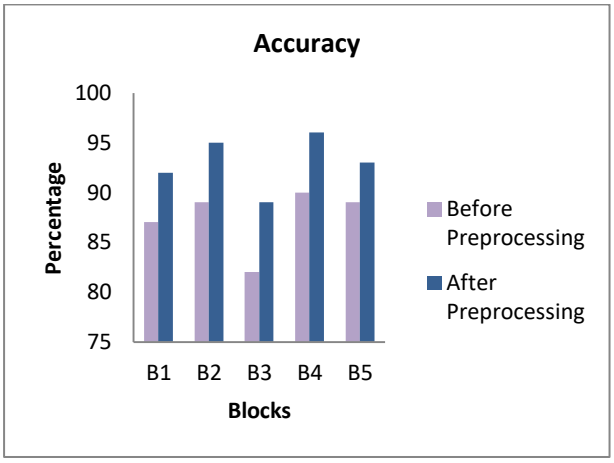

Figure 5: Accuracy Comparison

To compare processing time needed for cleansing data using the proposed algorithm, the preprocessing techniques are applied by partitioning and not partitioning datasets. The simulation result shows that the processing time is reduced if big datasets is preprocessed by partitioning and using mean value. Also, the difference in processing time is smaller for small datasets and increases with larger datasets. So for preprocessing datasets with less number of records it is not necessary to partition data. However with growing number of records, partitioning reduces considerable processing time.

Table 3: Processing Time

	Processing Time (ms)	
Records	Without Partitioning	With Partitioning
R100	8.7	7.8
R200	10.2	7.9
R300	13.1	9.1
R400	16.9	11.7
R500	17.8	12.3

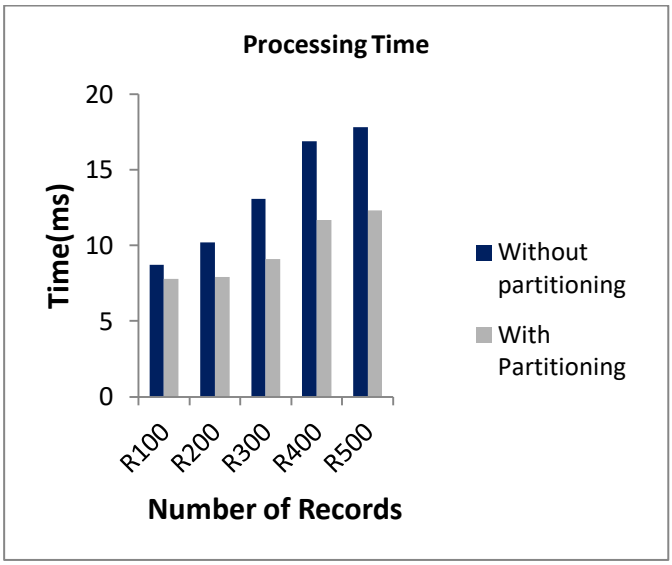

Figure 6: Processing Time Comparison

The performance of partitioning process is compared with the existing Shuffle and Chop techniques. The results shows that the proposed method of partitioning based on sensitivity parameter helps to protect data and also the partitioning time is reduced with increased number of partitions.

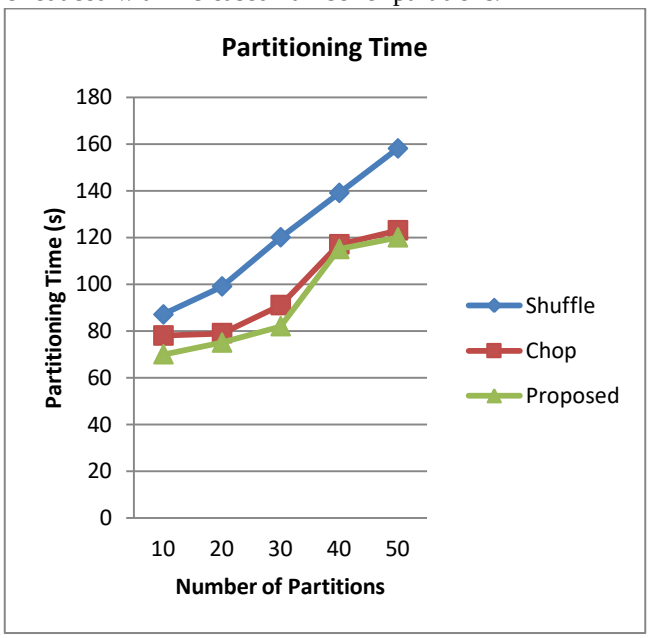

Figure 7: Partitioning Time Comparison

Thus, the proposed method can be used to partition dataset after preprocessing, so that queries over partitions can be executed in shorter period and also the partitions with sensitive data can be well protected.

VII. CONCLUSION

This paper proposes an effective preprocessing method for big data analytics with reduced processing time and high accuracy. Also, based on sensitivity parameters the big dataset can be partitioned further and is ready for upload on cloud. As every block of preprocessed data is assigned partition identifier, it is possible to distribute data among different servers by balancing workload. Sensitive data can be protected more precisely by using this type of preprocessing and partitioning method. This method works with big datasets consisting numerical and categorical values. This preprocessing algorithm cleanses big dataset and treats missing values and outliers effectively. In future, preprocessing of all types of big data including images, audio, and video can be considered.

REFERENCES

1. Kaitai Liang, Willy Susilo, and Joseph K. Liu, "Privacy-Preserving Ciphertext Multi-Sharing Control for Big Data Storage," IEEE Transactions on Information Forensics and Security, vol.10, no.8, 2015.
2. Boyang Wang, Baochun Li, and Hui Li, "Oruta: Privacy-preserving Public Auditing for Shared Data in the Cloud," IEEE Transactions on Cloud Computing, vol.2, no.1, 2014.
3. Cheng Hongbing, RongChunming, Hwang Kai, Wang Weihong, and Li Yanyan, "Secure Big Data Storage and Sharing Scheme for Cloud Tenants," China Communications, 2015.
4. Rongxing Lu, Hui Zhu, Ximeng Liu, Joseph K. Liu, and Jun Shao, "Toward Efficient and Privacy-Preserving Computing in Big Data Era", IEEE Network,2014.
5. Xinhua Dong, Ruixuan Li, Heng He, Wanwan Zhou, ZhengyuanXue, and Hao Wu, "Secure Sensitive Data Sharing on a Big Data Platform," Tsinghua Science and Technology, ISSN 1007-0214 08/11 ,pp 72-80, volume 20, number 1, 2015.
6. Zhiyuan Tan, Upasana T. Nagar, Xiangjian He, Priyadarsi Nanda, Ren Ping Liu, Song Wang, and Jiankun Hu, "Enhancing Big Data Security with Collaborative Intrusion Detection," IEEE Cloud Computing, 2014.
7. Chang Liu, Jinjun Chen, Laurence T.Yang, Xuyun Zhang, Chi Yang, Rajiv Ranjan, and RamamohanaraoKotagiri, "Authorized Public Auditing of Dynamic Big Data Storage on Cloud with Efficient Verifiable Fine-Grained Updates," IEEE transactions on parallel and distributed systems, 2014.
8. JoonsangBaek, QuangHieu Vu, Joseph K. Liu, Xinyi Huang, and Yang Xiang, "A Secure Cloud Computing based Framework for Big Data Information Management of Smart Grid," IEEE transactions on Cloud Computing, vol.3, no.2, 2015.
9. Guiyi Wei, Jun Shao, Yang Xiang, Pingping Zhu, Rongxing Lu, "Obtain Confidentiality or/and authenticity in Big Data by ID-based Generalized Signcryption," Elsevier Journal on Information Sciences, 2014.
10. JinboXiong, Ximeng Liu, Zhiqiang Yao, Jianfeng Ma, Qi Li, KuiGeng, and Patrick S. Chen, "A Secure Data Self-Destructing Scheme in Cloud Computing", IEEE transactions on cloud computing, vol.2, no.4, 2014.
11. MatturdiBardi, Zhou Xianwei, Li Shuai, and Lin Fuhong, "Big Data Security and Privacy: A Review," China Communications, supplement no.2, 2014.
12. Han Hu, Yonggang Wen, Tat-Seng Chua, and Xuelong Li, "Toward Scalable Systems for Big Data Analytics: A Technology Tutorial," IEEE Access, 2014.
13. Muhammad MazharUllahRathore et al. "Real-Time Big Data Analytical Architecture for Remote Sensing Application" IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Vol.8, No.10, October 2015.
14. Melbin J Reena, A. ShajinNargunam, "A Review on Cryptographic Approaches for Secured Processing of Big Data" IJCTA, 10(03), pp. 73-79, 2017.
15. J.Han, M.Kamber, and J.Pei. "Data Mining: Concepts and Techniques," San Mateo, CA, USA: Morgan Kaufmann, 2006.
16. Xiaochun Yun et al., "FastRAQ: A Fast Approach to Range Aggregate Queries in Big Data Environments," IEEE Transactions on Cloud Computing", Vol.3, No.2, April/June 2015.
17. Carlos E. Otero, Adrian Peter, "Research Directions for Engineering Big Data Analytics Software", IEEE, 2015.
18. C. Moretti, K. Steinhaeuser, D. Thain, N.V. Chawla, " Scaling up classifiers to cloud computers," Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pages 472-481, Washington, DC, USA, 2008.
19. Amy Xuyang Tan, Valerie Li Liu, Murat Kantarcioglu, Bhavani Thuraisingham, "A Comparison of Approaches for Large-Scle Data Mining," Technical Report, The University of Texas at Dallas, August 2010.