Journal of Theoretical and Applied Information Technology
31st March 2018. Vol.96. No 6
© 2005 – ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

M-SANIT: A FRAMEWORK FOR EFFECTIVE BIG DATA SANITIZATION USING MAP REDUCE PROGRAMMING IN HADOOP

[1]Y.SOWMYA, [2]Dr. M NAGARATNA, [3]Dr.C SHOBA BINDU

[1] Research Scholar, JNTUA, ANANTAPURAMU. Computer Science Engineering, India
[2]Associate Professor, JNTUH, HYDERABAD. Computer Science Engineering, India
[3]Professor, JNTUA, ANANTAPURAMU. Computer Science Engineering, India
E-mail: [1] yalla.soumya.reddy1980@gmail.com, [2]mratnajntu@gmail.com,[3]shobabindhu.cse@jntua.ac.in

ABSTRACT

Sanitization of big data before it is subjected to mining or publishing is very important for privacy reasons. Though sanitization is not new, sanitization of big data based on a measurability score is a novel idea. We proposed a framework known as M-Sanit to realize this idea. The framework is meant for big data sanitization prior to processing it. We proposed an extended misusablity score function that can return misuse probability of given dataset. This score plays an important role in determining the level of sanitization needed. This kind of sanitization provides expected level of anonymity and protects data from privacy attacks. The rationale behind this is that outsourced data may be misused by insiders. To get rid of this problem, the data is subjected to sanitization after finding measurability score. Our contributions in this paper are two-fold. First we provided mathematical model for extended measurability score. Second we proposed an algorithm to utilize the measurability score to determine the level of sanitization. We built a prototype application using locally configured Hadoop in clustered environment to demonstrate proof of the concept. Our results revealed the utility of M-Sanit for protecting big data from privacy problems.

Keywords: *Big data, Hadoop, Map Reduce, Misusability measure, sanitization*

1. INTRODUCTION

When data is outsourced for storage or data mining, it may be subjected to misuse. Due to the exponential growth of data in enterprises it became difficult to store and process data in the local machines. For this reason it is important to outsource data to cloud. Therefore it is important to protect data from privacy attack from malicious insiders or outsiders. Malicious insiders are the personnel of an organization who have intentions to misuse data or get monetary benefits from the data. When data is misused for inferring sensitive information, it causes loss of privacy. Non disclosure of sensitive data is nothing but preserving privacy. Privacy has to be given high importance as data leakage causes many problems to data owner. It is more so in cloud computing as people outsource their data to a remote service provider. The data centres where outsourced data is stored are untrusted from the view point of consumer as they are in remote place and maintained by cloud service provider. Leakage of data can lead to problems such privacy, security and loss of valuable information. Therefore identifying sensitive data and sanitizing it is very important problem. Sanitization is the process of changing data to make it not suitable for privacy or inference attacks. Generally data is available in tabular format. Such data is a set of attributes and corresponding values. Attributes in a given dataset can be of different types in the context of preserving privacy. They are known as key attributes, sensitive attributes and quasi identifiers. Key attributes, as the name imply, can show identity directly. For instance, Social Security ID (SSID) and name of a person are known as key attributes. When identity of a tuple in the dataset is known to either malicious insiders or attackers, it does mean that privacy is lost.

Attributes that are exploited to match with external data using certain attacks and establish identity information are known as quasi identifiers. Gender, birth date, zip code etc. are examples for quasi identifiers. Obtaining sensitive information from quasi identifiers is done through inference attacks. Inference attack, as the name implies, is the attack that that infers sensitive information from the data available. To overcome this problem data needs to be anonymized [1]. Anonymization is the process of changing original data so as to ensure that identity of any record is not disclosed. In the literature, plethora of anonymization techniques are found. However, most important and widely used ones are k-anonymity [14], l-diversity [13] and t-closeness [15]. Background knowledge and heterogeneity are attacks made on k-anonymity. It does mean that k-anonymized data can be subjected to such attacks. To overcome this problem, another anonymization technique known as l-diversity came into existence where such inference attacks are not possible. However l-diversity exhibits tradeoff between privacy and data management effectiveness. The attribute disclosure issues of l-diversiity is overcome by t-closness.

These anonymity algorithms, though they are effective, they cannot be used with big data directly [6] where data has heterogeneity and parallel processing is needed with thousands of commodity computers are involved. In other words, big data is processed in distributed programming frameworks like Hadoop where MapReduce programming is used. Therefore, these algorithms are to be redesigned and implemented with the new programming paradigm. Big data is the data which is voluminous, containing multiple kinds of data such as structured, unstructured and semi-structured data and the data comes from different sources dynamically from time to time. MapReduce programming paradigm is used in distributed computing. In other words to process big data MapReduce kind of programming is essential. This new programming model has two important tasks such as map task and reduce task. It is best used to process big data where voluminous data needs to be handled. Hadoop [4] is the distributed programming framework which supports MapReduce programming paradigm. In the presence of data which exhibits exponential growth, privacy of data needs to be preserved [16]. Our previous work on parallelization of k-anonymity [24] and privacy preserving big data mining [24] provide useful insights in this area.

What are the benefits of having misusability measure based architectural framework for big data sanitization? It is an important question which probes into the need for sanitization in the context of big data. This paper answers this question. The motivation behind this work is that big data when outsourced to public cloud is exposed to internal and external threats. Especially privacy threat is there with internal members who are supposed to use the data for mining purposes. It is essential to overcome misusability problem by protecting privacy of such data using a framework. Our work in this paper provides such framework for big data sanitization based on misusability measure. Once the degree of misusability is known, it is employed to provide appropriate level of sanitization.

Our contributions in this paper are as follows.

- We implemented extended misusability score function which measures given data set and returns probability of misue. The result is a value between 0.0 and 1.0.
- We proposed and implemented a framework known as M-Sanit. The framework facilitates sanitization of given big data for which misusability score is known. Based on the score it applies level of sanitization.
- We proposed an algorithm for determining the level of sanitization based on the misusability score.
- We made experiments with Hadoop cluster configured locally.

The remainder of the paper is structured as follows. Section II provides review of literature. Section III presents the proposed system in detail. Section IV presents experimental results while section V concludes the paper.

2. RELATED WORKS

This section reviews literature on privacy issues, anonymization or sanitization techniques that are used for preserving privacy of data.

Privacy Preserving Data Publishing/Mining

Traditionally data is published or given to third party for data mining. Such data needs to be protected from privacy attacks. Therefore it is important to anonymize sensitive data before giving to third party for publishing or mining. Data anonymization with popular techniques such as k-Anonymity [12], l-diversity [13] and t-closeness [15] could prevent privacy attacks on data. “A release of data is said to have the k-anonymity property if the information for each person contained in the release cannot be distinguished from at least k-1 individuals whose information also appear in the release”. It is the definition of k-anonymity which has US patent. To overcome the drawbacks of k-Anonymity (vulnerable to background knowledge and heterogeneity attacks), l-diversity performs group based anonymization with reduced granularity. It gains more privacy and has trade off with data mining effectiveness. This disadvantage of l-diversity is overcome with t-closeness where values of an attribute are treated based on the distribution of values. Therefore t-closeness is used as a measure to gain knowledge on how the distribution of values with respect to sensitive attribute.

Horizontally distributed databases are studied in [19] for Privacy Preserving Data Publishing (PPDP) with a set of decentralized anonymization protocols. These protocols dealt with multiple sites where data is partitioned horizontally. The result of each site is piped as input for another site and sanitization of data is made using k-anonymity. Recent improvements in the area of PPDM are explored in [20]. It reveals four attacks models that are used frequently for privacy disclosure. They include probabilistic attack, table linkage attack, record linkage attack and attribute linkage attack. They also covered various privacy models and studied their performance with the attack models aforementioned. On the other hand various distributed data mining protocols are studied in [21]. They include distributed k-anonymization, BN learning and BN techniques.

Privacy Preserving Data Publishing/Mining for Big Data

Privacy preserving data mining or data publishing with big data is relatively new phenomenon. It is carried out in distributed programming frameworks like Hadoop. In [18] PPDM techniques are explored including randomization, secret sharing, homomorphic encryption, and multi-party computation. PPDM in Hadoop MapReduce programming is studied in [1]. They presented architecture for PPDM that exploits parallel processing power of Hadoop. A set of privacy preserving techniques with MapReduce programming paradigm is presented in [2]. Their framework is known as Sedic that exhibits automatic partitioning and effective security model for job computations. They named it as data-intensive and privacy-aware computing. They exploited file system of MapReduce with some strategic changes for storing sanitized data and original data. Public cloud is preferred for storing sanitized data while sensitive data is stored in private cloud. To this effect, Sedic needs labelling of data which is sensitive.

Big data anonymization is studied in [3] by proposing an algorithm that partitions given data and anonymize it using MapReduce programming model. In [5] tamper resistant hardware is used in order to achieve privacy is query execution. Securing SQL queries is made in presence of an attacker model. To this effect, they proposed a protocol compatible with MapReduce programming paradigm. Local-recording problem is investigated in [7] for overcoming privacy breaches. Their method is scalable and named as t-ancestors clsutring that performs clustering in two phases. It makes use of an algorithm known as agglomerative clustering which is proximity-aware.

For PPDM in hybrid cloud, Prometheus is the system proposed in [8] which uses many techniques for data sanitization using MapReduce programming. Quasi identifiers are effectively sanitized. A vision for PPDM with scalability, security and adaptability is presented in [9]. Insider attacks are studied on big data which is outsourced to cloud platforms in [10]. They proposed a privacy model known as m-adversary in the presence of colluding data providers and data owners. They explored the concept of collaborative data publishing in which data comes from different parties.

Data release with privacy-preservation is made in [11]. They used the technique known as Statistical Disclosure Control (SDC) and distance-based record linkage. With respect to big data, privacy and security issues are presented in [12]. They also made review of techniques for biometric authentication, collaborative data mining with privacy preserved and data matching with privacy ensured. PPDM techniques with union, sum and intersection are the main focus in [17]. They applied the techniques for different applications such as Expectation Maximization (EM), horizontal partitioning of data and association rule mining.

The techniques found in the literature are meant for privacy preserving data mining. However, when privacy needs to be preserved through sanitization techniques, we found that misusability score obtained given data set can help in determining the level of sanitization. When the sanitization level is known, it is possible to ensure that the data is sanitized and also has not lost its utility for data mining and data publishing purposes. In this regard, the proposed framework contributes. The rationale behind this is that there is relationship between data sanitization and utility. When sanitization degree is more it affects utility of data as well.

3. MAPREDUCE PROGRGMMING PARADIGM

MapReduce, as explored in [22], is a distributed programming framework that is fault tolerant, scalable and supports parallel processing. Thus it can leverage Graphics Processing Unit (GPU) used in commodity computers in cloud computing environment. MapReduce was first introduced by Google in 2004 for processing voluminous data. In other words, it is used to process big data which has the characteristics such as volume, variety and velocity. The processing is carried out by thousands of worker nodes that are associated with cloud eco-system. This new programming paradigm has attracted industry and academia. The MapReduce framework has two distinct functions known as map function and reduce function. The data is stored and processed in the form of key-value pairs. Figure 1 illustrates the functioning of Map and Reduce for processing big data. The input and output data is stored Hadoop Distributed File System (HDFC) associated with Hadoop.

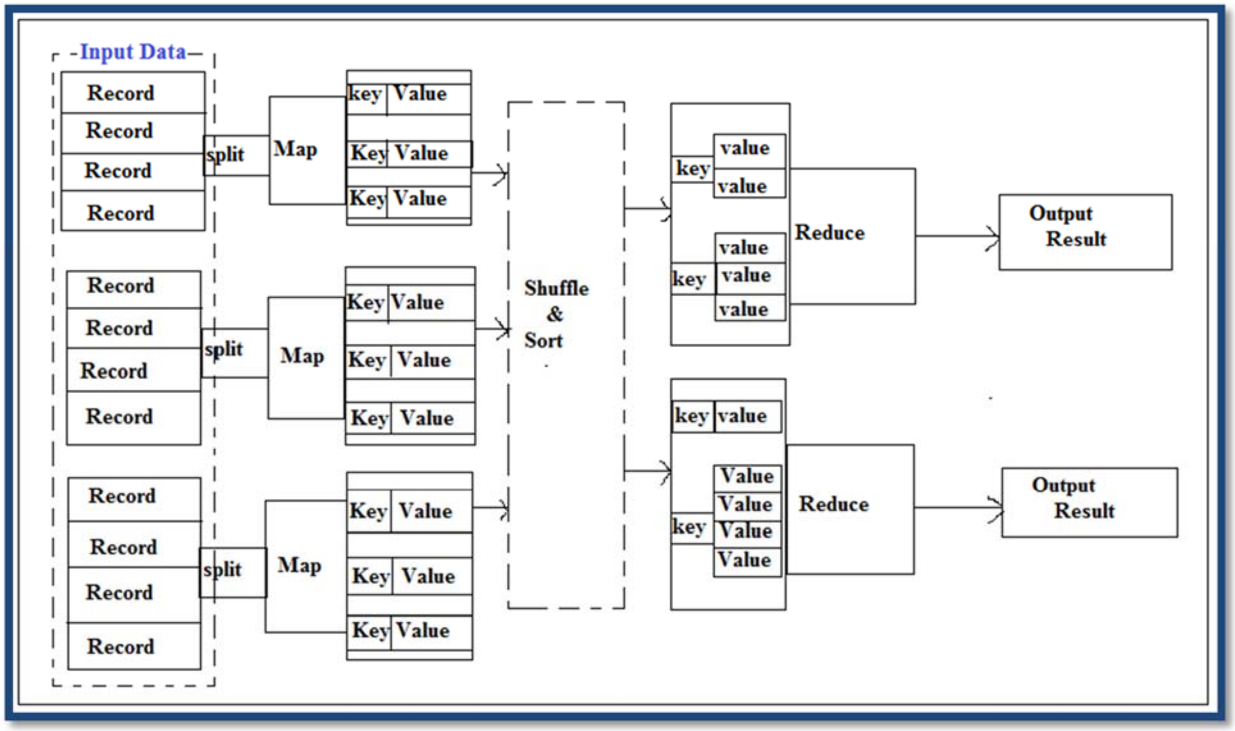

Figure 1: Execution Flow of MapReduce Programming Paradigm

The input data obtained from HDFS is split into multiple parts or chunks. For each chunk of data a map task is executed. The result of map task is nothing but key-value pairs. The output of the map function may be subjected to shuffling or sorting. It results in keys to be sorted. Afterwards, the reduce task is executed and generates final result which is stored in HDFS again. The final result is nothing but key-value pairs which is said to be the result of MapReduce job.

4. METHODOLOGY FOR REALIZING M-SANIT

In this paper we proposed a framework known as M-Sanit for PPDM in distributed programming environment using Hadoop. The framework is implemented in distributed programming paradigm MapReduce. It is meant for protecting big data before it is subjected to publishing or data mining. The overview of the M-Sanit framework is shown in Figure 2. It takes big data which is stored in HDFS as input. HDFS is associated with Hadoop's reliable, scalable and low-latency storage. M-Sanit needs big data and also the misusability score of the data in order to sanitize data based on the probability of misusability measured using extended misusability score function proposed in this section.

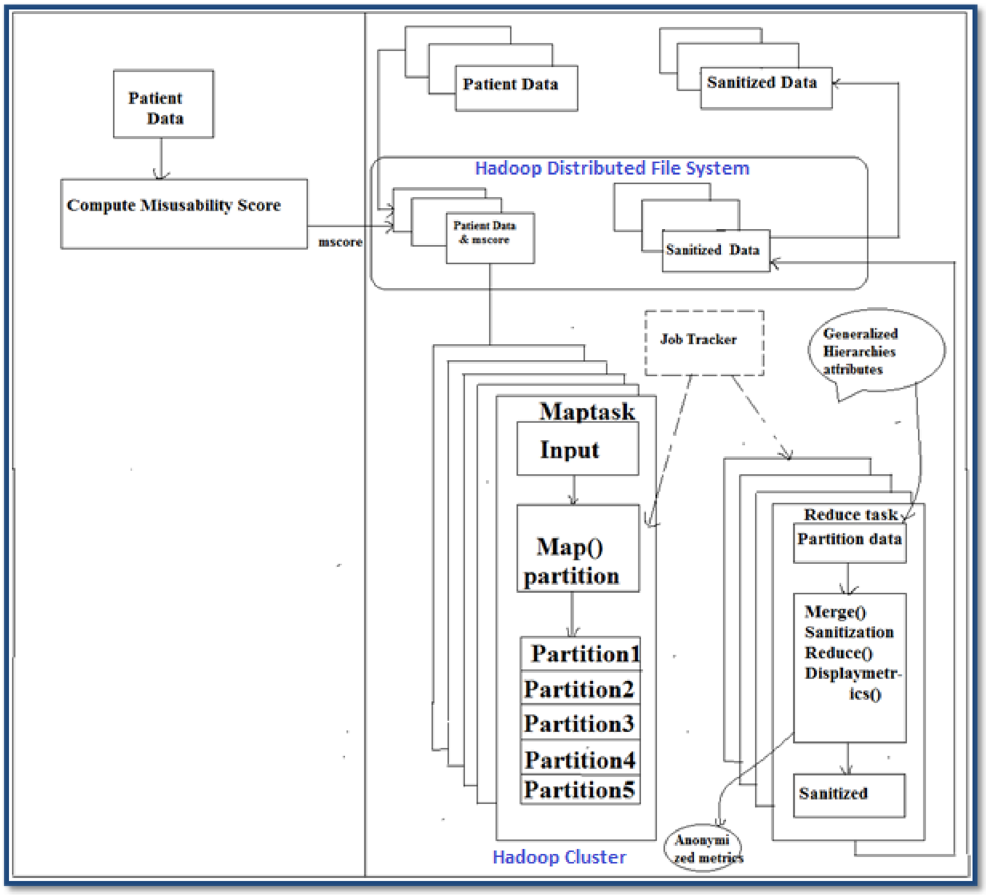

Figure 2: Overview of M-Sanit for Misuability Score Based Big Data Sanitization

As shown in Figure 2, big data is considered from healthcare domain where huge amount of data is accumulated from time to time. Electronic data of patients is the big data used as input to M-Sanit. Prior to using it, big data is subjected to extended misusability score which returns misuse probability of given data. This is very crucial part which is used to determine the level of sanitization using the proposed algorithm. The Map task in the framework partitions data into many chunks and output in the form of key-value pairs is given to reduce task. Actual sanitization is performed by the reduce task. Thus the MapReduce job is completed and big data gets sanitized before further processing. Misusability score function is used to find probable misuse of given big data and suitable sanitization is employed to have effective preservation of privacy of big data. Misusability score function details are provided in the ensuing section.

Computing Misusability Score

This is a measure which is used to compute the probability of misuse for a given dataset. The extended misusability measure presented in this paper is an extension to the work of Harel et al. [23]. The extended misusability score function is utilized in this paper for effective sanitizatin of big data. Figure 3 shows a sequence of steps to be carried out to compute misusability score. There are many phases involved. They include computation of raw record score, computation of record distinguishing factor, computation of final record score and computationa of misusability score. This is done by taking big dataset as input and performing aforementioned steps that result in finding the probability of misuse of given dataset. This is very crucial information for effective big data sanitization which is exploited by the proposed algorithm.

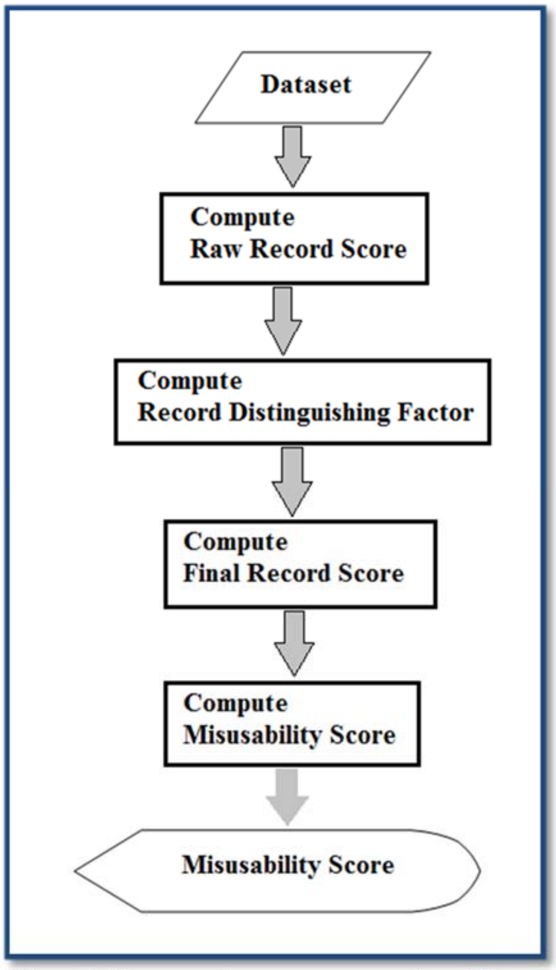
Dataset
Compute Raw Record Score
Compute Record Distinguishing Factor
Compute Final Record Score
Compute Misusability Score
Misusability Score

Figure 3: Sequence of steps to compute misusability score

Here are the steps described to understand the functioning of each phase in the process of computing misusability score.

Raw Record Score: This is nothing but the sensitivity score of a single record taken from the dataset which contains multiple attributes. The Raw Record Score (RRS) for a given record is computed as follows.

$$RRS_i=\min\left(1,\sum_{S_j \epsilon T} f\left(c,S_j[x_i]\right)\right) \quad (1)$$

As the RRS is computed as the sum of sensitive values of all attributes, its value is higher when a given dataset has more sensitive attributes. In the similar way, RRS is low when the number of sensitive attributes is less. However, the result of it should be 1 or less than 1.

Record Distinguishing Factor: It is used as a metric to find the ability of a quasi-identifier to disclose identity of given entity in the dataset. The result of this measure is between 0.0 and 1.0. Thus the function used to find distinguishing factor is as follows.

$$DF: \{\text{quasi-identifiers}\} \rightarrow [0,1] \quad (2)$$

Final Record Score: This is the measure which makes use of DF and RRS. A table containing number of records denotes *r*, can have the final record score computed as follows.

$$FRS=max_{0\le i\le r}(RS_i)= max_{0\le i\le r}\left(\frac{RRS_i}{D_i}\right) \quad (3)$$

For each record in the dataset, weighted sensitivity score is computed and then it is divied by DF. This operation results in maximal weighted sensitivity score for given table. It is nothing but FRS.

Misusability Score: This is the final measure derived which is used to determine the probability of misuse for given dataset. It is computed using FRS, number of records and importance of quantity factor denoted as x(x>=1).

$$MS = r^{\frac{1}{x}} \text{ X } FRS = r^{\frac{1}{x}} \times max_{0\le i\le r}\left(\frac{RRS_i}{D_i}\right) \quad (4)$$

B. Misusability Measure-Based Big Data Sanitization (MMBDS)

The big data sanitization proposed with M-Sanit is realized using the algorithm MMBDS. This algorithm performs various functions including misusabilility score computation, finding level of data sanitization required and finally applying the sanitization to big data in MapReduce programming paradigm. Thus it achieves appropriate level of big data sanitization with privacy preserved.

```
Inputs: Dataset D
Output: Sanitised Dataset D'
Initialize level=0
Finding Score of Miusability for D
  Compute Raw Record Score rrs (1)
  Compute Record Distinguishing Factor rdf (2)
  Compute Final Record Score frs (3)
  Compute Misusability Score ms (4)
Finding Level of Sanitization Needed
  IF ms>=0.0 and ms<=0.3 THEN
        level = 1
Else IF ms>=0.4 and ms<=0.7 THEN
        level = 2
else
        level = 3
Sanitization
  D' = Sanitize(level)
  Return D'
```

Algorithm 1: MMPP algorithm

As presented in Algorithm 1, big data denoted as *D* is table as input and performs misusability score based optimal sanitization. It returns the sanitized big data *D'* which can be safely distributed for publishing or data mining with privacy preserved. Once misusability score is computed, the level of sanitization required is determined by the algorithm prior to application of sanitization technique to big data.

This paper presents a framework for better sanitization of big data. A misusability score is computed for given dataset prior to the sanitization process. Once the data is subjected to computation of misusability score, the data is given as input to M-Sanit which is a framework that is based on MapReduce programming paradigm. Along with the data misusability score is also provided. Based on this, the proposed methodology takes care of appropriate sanitization of the data. The algorithm proposed in this paper is the baseline algorithm used to know the level of sanitization and apply the same to given big data. However, it needs to be improved and evaluated further to have generalized conclusions and recommendations. These guidelines can be used to have appropriate sanitization to big data that serves dual benefit such as privacy preservation of data and also to ensure utility of the sanitized data. It strikes the balance between sanitization and the utility of the sanitized data.

6. EXPERIMENTAL RESULTS

We made experiments with Hadoop installed in the local system in cloud computing environment. The local machine has Linux operating system with 4 GB RAM with 1.70 GHz processing speed. Single machine Hadoop is used for experiments. The execution of the distributed programming framework Hadoop is made in the local machine and the proposed framework is evaluated against intended functionality.

We built a prototype application to demonstrate proof of the concept of M-Sanit. Information loss is computed and presented in Figure 4. The information loss is observed with different levels of sanitization. Four datasets are collected from UCI [26] and altered to have more instances. Adult dataset has 44453 instances, breast cancer dataset 36566, census dataset 73456, and diabetes dataset has 210234 records. With these datasets, two performance metrics such as execution time and memory consumption are used to evaluate the proposed methodology.

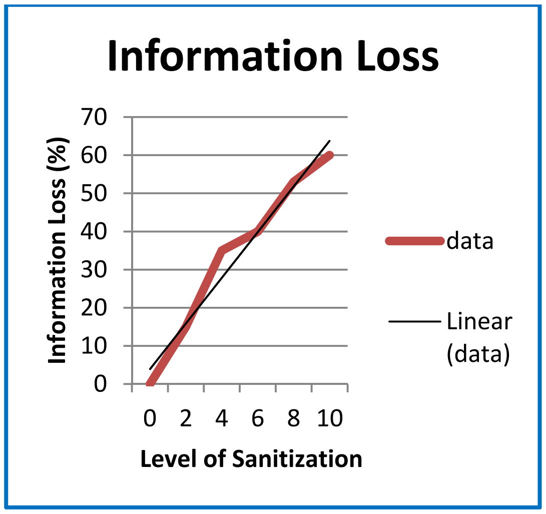

Figure 4: Information loss vs. level of sanitization

As presented in Figure 4, horizontal axis presents level of sanitization while the vertical axis presents information loss %. As revealed in the graph, the information loss is increased when level of sanitization is increased. Information loss is the loss is the loss of data when dataset is subjected to sanitization. Sometimes, it may lead to complete lack of utility a well. Therefore, it is important to ensure that the sanitized data has certain utility so as to perform data mining on it and discover business intelligence.

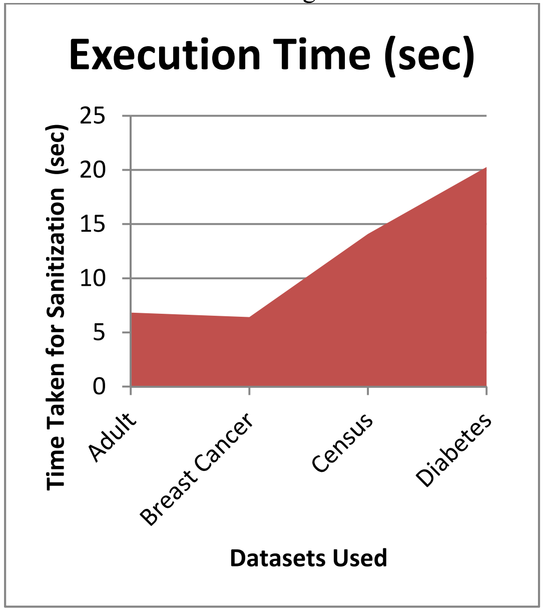

Figure 5: Performance in terms of execution time (seconds)

As shown in Figure 5, it is evident that the performance of proposed algorithm for different datasets is presented in terms of execution time. The diabetes dataset needed highest execution time while the breast cancer dataset needed least execution time.

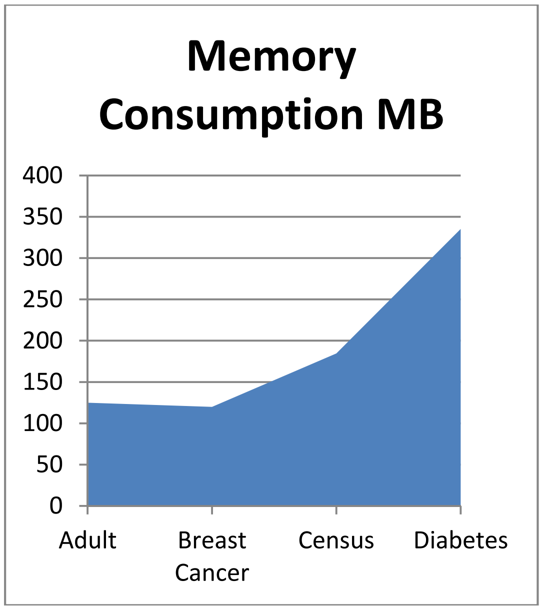

Figure 6: Performance in terms of memory consumption (MB)

As shown in Figure 6, it is evident that the performance of proposed algorithm for different datasets is presented in terms of memory consumed. The diabetes dataset revealed highest memory consumption while the breast cancer and adult datasets consumed least memory.

7. DIFFERENCE FROM PRIOR WORK

The prior work found in the literature is classified into privacy preserving data mining and privacy preserving data publishing. The works found in the literature dealt with data sanitization or data anonymity for privacy preservation. Since privacy is nothing but non-disclosure of data, it is important to have some sort of security mechanism to protect data from abuse. The works found in the literature have little information on the misusability measure based approach for sanitisation. Our framework provides the required components for ensuring big data sanitization to the level required besides ensuring the utility of the outsourced data in public cloud. This is the main difference between our work and the works found in the literature.

8. LIMITATION OF THIS WORK

Our work does have limitations. A framework is proposed and an algorithm is defined. However, experimental evaluation to know the tradeoffs between the level of sanitization and the degree of misusability. Therefore there is need to have more experimental evaluation with real cloud platforms. Another important limitation is to make use of truly big data instead of using just voluminous data. This work needs further improvement in terms of finding level of sanitization for given degree of misusability. Once it is done, it is possible to draw conclusions on the tradeoffs between sanitization and utility of sanitized dataset. When there utility of sanitised data, such data is useful for protecting privacy and also ensure utility of the data.

9. CONCLUSION AND FUTURE WORK

As the cloud computing brings many advantages to individuals and organizations, people started outsourcing their data and compute operations to cloud. As the cloud eco-system has inexhaustible computing resources, the trend of using it became more and more in the recent past. Cloud computing also brought privacy risks. Especially the data which is outsourced for publishing or data mining can be misused and privacy of the data may be lost. This kind of sensitive information disclosure is made by either internal or external attacks. Therefore it is essential to have anonymization techniques to prevent privacy attacks. There are two issues with existing anonymization techniques such as k-Anonymity, l-diversity and t-closeness. The first problem is that they cannot work with big data in MapReduce programming paradigm. Second problem is that they cannot determine the level of sanitization needed. To overcome these drawbacks, we proposed M-Sanit, a framework which computes misuse probability of big data prior to adapting to the level of ssanitization needed based on the misusability score. The M-Sanit framework is implemented using a prototype application that runs in Hadoop. Experiments are made in the local Hadoop environment in terms of information loss when different levels of sanitization is applied and the execution time and memory consumption for different datasets obtained from UCI machine learning repository. In future we intend to evaluate M-Sanit with well known metrics and standardise the proposed algorithm with more optimal thresholds for effective big data sanitization.

REFERENCES:

[1] Deepalakshmi V, Mayuranathan M and Balasubramani S. (2015). An Enhanced Data Anonymization Technique to Preserve Privacy in Big Data. International Journal of Advances in Engineering. 1 (2), p66 -71.

[2] Kehuan Zhang, Xiaoyong Zhou, Yangyi Chen, XiaoFeng Wang and Yaoping Ruan. (2011). Sedic: Privacy-Aware Data Intensive Computing on Hybrid Clouds. ACM, p1-11.

[3] Priyashree H.C and Mr. Justin Gopinath. (2014). Privacy Preserving and Scalable Processing of Data Sets Using Data Anonymization and Hadoop Map Reduce on Cloud. International Journal of Research in Information Technology. 2 (5), p1-9.

[4] The Apache Software Foundation. (2016). Welcome to Apache™ Hadoop. Available: http://hadoop.apache.org/. Last accessed 01 December 2016.

[5] Quoc-Cuong. (2016). Privacy-Preserving Query Execution using Tamper Resistant Hardware. Université Paris-Saclay, p1-141.

[6] Avita Katal,Mohammad Wazid and R H Goudar. (2013). Big Data: Issues, Challenges, Tools and Good Practices. IEEE, p1-6.

[7] Xuyun Zhang, Wanchun Dou, Jian Pei,Surya Nepal,Chi Yang, Chang Liu, and Jinjun Chen. (2013). Proximity-Aware Local-Recoding Anonymization with MapReduce for Scalable Big Data Privacy Preservation in Cloud. IEEE TRANSACTIONS ON COMPUTERS, p1-14.

[8] Zhigang Zhou, Hongli Zhang, Xiaojiang Du, Panpan Li and Xiangzhan Yu. (2013). Prometheus: Privacy-Aware Data Retrieval on Hybrid Cloud. Proceedings IEEE INFOCOM, p1-10.

[9] Li Xiong,Slawomir Goryczka and Vaidy Sunderam. (2011). Adaptive, Secure, and Scalable Distributed Data Outsourcing: A Vision Paper. ACM, p1-6.

[10] Vishal Phadtare, Samir Kashid, Monika Dherange and Prof. Pramod Murkute. (2016). Privacy Preservation of Big Data Using Hadoop. IJSRSET. 2 (3), p1-9.

[11] Javier Herranz, Jordi Nin, Pablo Rodr guezb and Tamir Tassa. (2015). Revisiting Distance-Based Record Linkage for Privacy-Preserving Release of Statistical Datasets. Elsevier, p1-25.

[12] Elisa Bertino and Murat Kantarcioglu. (2014). Big Data – Security with Privacy Response to RFI for National Privacy Research Strategy. IEEE, p1-9.

[13] Ashwin Machanavajjhala, Johannes Gehrke and Daniel Kifer. (2010). ℓ-Diversity: Privacy Beyond k-Anonymity. IEEE, p1-12.

[14] Charu C. Aggarwal. (2005). On k-Anonymity and the Curse of Dimensionality. IEEE, p1-9.

[15] Ninghui Li, Tiancheng Li and Suresh Venkatasubramanian. (2007). t-Closeness: Privacy Beyond k-Anonymity and -Diversityt-Closeness: Privacy Beyond k-Anonymity and -Diversity. IEEE, p1-10.

[16] Chris Clifton. (2001). Privacy Preserving Distributed Data Mining. Elsevier, p1-10.

[17] Chris Clifton, Murat Kantarcioglu, Jaideep Vaidya,Xiaodong Lin and Michael Y. Zhu. (2002). Tools for Privacy Preserving Distributed Data Mining. Springer , p1-7.

[18] V. Baby and N. Subhash Chandra. (2016). Privacy-Preserving Distributed Data Mining Techniques: A

Survey. International Journal of Computer Applications. 143 (10), p1-5.

[19] Pawel Jurczyk and Li Xiong. (2008). Privacy-Preserving Data Publishing for Horizontally Partitioned Databases. ACM, p1-2.

[20] BENJAMIN C. M. FUNG,KE WANG,RUI CHEN and PHILIP S. YU. (2010). Privacy-Preserving Data Publishing: A Survey of Recent Developments. ACM Computing Surveys. 42 (4), p1-53.

[21] Rebecca N. Wright and Zhiqiang Yang and Sheng Zhong. (2006). Distributed Data Mining Protocols for Privacy: A Review of Some Recent Results. Springer , p1-13.

[22] Apache Software Foundation. (2016). MapReduce Tutorial. Available: https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html. Last accessed 01 December 2016.

[23] Harel, A., Shabtai, A., Rokach, L., and Elovici, Y. (2012). M-Score: A Misusability Weight Measure. IEEE Transctions on Depedeable and Secure Computing, 9 (3), p414-428.

[24] Y. Sowmya and Dr. M Naga Ratna, "A Review on Big Data Mining, Distributed Programming Frameworks and Privacy Preserving Data Mining Techniques," International Journal of Advanced Research in Computer Science. 6 (1), p1-6, 2015.

[25] Y. Sowmya and Dr. M. Nagaratna, "Parallelizing K- Anonymity Algorithm for Privacy Preserving Knowledge Discovery from Big Data," International Journal of Applied Engineering Research. 11 (2), p1-8, 2016.

[26] UCI (2017). UCI Machine Learning Repository. Available online at: https://archive.ics.uci.edu/ml/index.php. [accessed on: 20 April 2017]