**International Journal of Electrical and Computer Engineering (IJECE)**
Vol. 8, No. 6, December 2018, pp. 4524~4532
ISSN: 2088-8708, DOI: 10.11591/ijece.v8i6.pp4524-4532 

# Misusability Measure Based Sanitization of Big Data for Privacy Preserving MapReduce Programming

**D. Radhika[1], D. Aruna Kumari[2]**
[1]K L University, Computer Science Engineering, India
[2]K L University, Department ECM, India

**Article Info**

***Article history:***

Received Sep 1, 2017
Revised Feb 20, 2018
Accepted Jun 11, 2018

***Keyword:***

Big data
Misusability measure
Privacy Preserving Data Mining (PPDM)
Privacy Preserving Data Publishing (PPDP)
Sanitization

**ABSTRACT**

Leakage and misuse of sensitive data is a challenging problem to enterprises. It has become more serious problem with the advent of cloud and big data. The rationale behind this is the increase in outsourcing of data to public cloud and publishing data for wider visibility. Therefore Privacy Preserving Data Publishing (PPDP), Privacy Preserving Data Mining (PPDM) and Privacy Preserving Distributed Data Mining (PPDM) are crucial in the contemporary era. PPDP and PPDM can protect privacy at data and process levels respectively. Therefore, with big data privacy to data became indispensable due to the fact that data is stored and processed in semi-trusted environment. In this paper we proposed a comprehensive methodology for effective sanitization of data based on misusability measure for preserving privacy to get rid of data leakage and misuse. We followed a hybrid approach that caters to the needs of privacy preserving MapReduce programming. We proposed an algorithm known as Misusability Measure-Based Privacy Preserving Algorithm (MMPP) which considers level of misusability prior to choosing and application of appropriate sanitization on big data. Our empirical study with Amazon EC2 and EMR revealed that the proposed methodology is useful in realizing privacy preserving Map Reduce programming.



***Corresponding Author:***

D. Radhika,
K L University, Computer Science Engineering,
Guntur - 522502, Andhra Pradesh, India.
Email: radhikarajasekhar@yahoo.com

## 1. INTRODUCTION

Big data has become a known buzz word as it is well understood in the wake of new technologies like cloud computing and distributed programming frameworks like Hadoop [1] that supports new programming paradigm Map Reduce [2]. As this framework can leverage parallel processing and thus supports processing of massive data, enterprises started switching to cloud based storage and processing. This way cloud based data publishing and data mining became a reality. More information on big data and distributed programming frameworks can be found in our prior work [3]. With plethora of advantages such as on demand storage and computing without time and geographical restrictions and capital investment, in pay per use fashion, cloud also brought challenges. Leakage and misuse of sensitive data is one such challenge that needs more research. When data is outsourced for publishing and data mining privacy issues come into picture. These issues may lead to potential risk to customers and even raise legal hurdles to enterprises. Let us have some understanding on privacy in terms of attributes and sensitivity levels of data being published. Our focus is limited to data in tabular form only.

The attributes in any given dataset can be classified into quasi-identifiers, sensitive attributes and other attributes. Quasi-identifier is an identifier that does not reveal sensitive information directly but an attacker may be able to infer sensitive data from it. Sensitive attribute on the other hand has private data that

should not be disclosed. Non-disclosure of sensitive information is the aim of privacy preserving data publishing. Other attributes do not reveal sensitive data and attackers can never infer sensitive information from them. There are two sensitive attributes found in Table 2 which is derived from Table 1. They are account type and average monthly bill. The former shows importance of account while the latter shows spending patterns of customer. Adversaries can exploit such information.

Table 1. The Source Table

| Job | City | Sex | Account Type | Average Monthly Bill |
|---|---|---|---|---|
| Lawyer | NY | Female | Gold | $350 |
| Gender | LA | Male | White | $160 |
| Gender | LA | Female | Silver | $200 |
| Lawyer | NY | Female | Bronze | $600 |
| Teacher | DC | Female | Silver | $300 |
| Gardener | LA | Male | Bronze | $200 |
| Teacher | DC | Female | Gold | $875 |
| Programmer | DC | Male | White | $20 |
| Teacher | DC | Female | White | $160 |

Table 2. The Published Table

| Job | City | Sex | Account Type | Average Monthly Bill |
|---|---|---|---|---|
| Lawyer | NY | Female | Gold | $350 |
| Lawyer | NY | Female | Bronze | $600 |
| Teacher | DC | Female | Silver | $300 |
| Gardener | LA | Male | Bronze | $200 |
| Programmer | DC | Male | White | $20 |
| Teacher | DC | Female | White | $160 |

Our prior work [4] on Mining as a Service (MaaS) did not focus on privacy of data being published or mined with respect to Map Reduce programming. However, we understood that sensitivity level of data is important in making sanitization decisions. Sanitization is the process of hiding sensitive data by adding noise to data. Many anonymization techniques came into existence as explored in [5]. However, in the context of cloud and big data an integrated approach which takes care of privacy of data and publishing or mining of data based on level of misusability is missing. This is the motivation behind the work in this paper. Our contributions in this paper are as follows.

a. We proposed a comprehensive and integrated methodology for privacy preserving big data publishing or processing with respect to Map Reduce programming using Hadoop framework.
b. We proposed an algorithm known as Misusability Measure-Based Privacy Preserving Algorithm (MMPP) to determine level of misusability before applying appropriate sanitization technique.
c. We made an empirical study with Amazon EC2 and EMR. Amazon Simple Storage Service (S3) is used to store big data while Amazon Elastic MapReduce is used for implementation of privacy preserving big data processing with MapReduce programming paradigm.
d. We evaluated our methodology with big data (structured data) and the results revealed that proposed methodology is useful in realizing privacy preserving MapReduce programming.

The remainder of the paper is structured as follows. Section 2 reviews related works. Section 3 presents proposed methodology. Section 4 presents experimental results. Section 5 concludes the paper while section 6 provides directions for future work.

## 2. RELATED WORKS

This section provides review of literature on related works. Heatherly et al. [6] focused on inference attacks and the prevention of the same in social networks. They employed the notion of collective inference in order to discover sensitive attributes from given dataset. Acs et al. [7] proposed two sanitization techniques that make use of redundancy features of real world datasets. These techniques are used to have lossy compression of data before applying sanitization. Their firs scheme is optimization of Fourier Perturbation Algorithm (FPA) while the second scheme is based on clustering technique. Chen et al. [8] explored differential privacy model for transit data publication. They published large volumes of sequential data using their model based on differential privacy.

Askari et al. [9] proposed an information theoretic framework for privacy preserving data publishing. They evaluated their framework with two kinds of background knowledge. Original dataset

knowledge and the user's knowledge of dataset are the two kinds. Their work is meant for measuring privacy and utility of sanitization approaches in the confines of information theory.

Domadiya and Rao [10] proposed a heuristic based algorithm for hiding sensitive association rules for maintaining data quality and privacy. Their algorithm is known as Modified Decrease Support of RHS item of Rule Clusters (MSRRC). The algorithm modifies transactions in order to achieve sanitation. Canard and Lescuyer [11] proposed a novel approach for sanitizing personal data that makes use of anonymous credentials. Their framework does not support existing sanitization techniques as it is meant for different approach in terms of anonymous credential system. Lin et al. [12] followed a greedy-based approach for sanitization. They hide sensitive data by transaction insertion. Shar and Tan [13] proposed an approach for predicting web application vulnerabilities such as cross site scripting and SQL injection. To achieve this they used sanitization technique to hide code patterns. Xiao et al. [14] presented a data sanitization technique for inferring network's structure. This is done using differentially private fashion. Towards this end they employed statistical hierarchical random graph (HRG) model.

Gambs et al. [15] proposed de-anonymization attack on massive amount of location data collected by GPS based systems. They implemented the attack using Mobility Markov Chain (MMC) model. This is done by observing mobility traces found in the dataset. Their attack was meant for measuring the strength of sanitization mechanisms. Zhang et al. [16] proposed a method to sanitize location based recommendations as they carry location related sensitive data. Their method is based on differential privacy. Sanchez et al. [17] focused on improving sanitization of textual documents. Their approach automatically finds sensitive terms in text documents and sanitizes them. Their approach significantly reduces the risk of disclosure of sensitive information. Sun et al. [18] employed sensitization routines for detecting vulnerability known as integer-overflow-to-buffer-overflow. Their technique is known as dynamic tracking technique. Li et al. [19] studied the need for sanitizing databases before outsourcing them, especially for software testing tasks. Heffetz and Ligett [20] contributed towards privacy based research which includes differential privacy and de-identification.

Clifton [21] explored the concept of distributed data mining with privacy preserving approaches. They discussed privacy preserving association rule mining, and component algorithm. A survey of privacy preserving data mining can be found with different techniques in [22]. Dwork et al [23] studied statistical validity while performing adaptive data analysis. They focused on accuracy guarantee analysis of statistics. Similar kind of work was found in [24]. Clifton et al. [25] presented tool for PPDDM (Privacy Preserving Distributed Data Mining). The tools include secure multi-party computation, secure sum, secure set union, secure size of set intersection, and scalar product. A survey on PPDDM is found in [26] other techniques like homomorphism encryption, secret sharing scheme, and randomization techniques are used for PPDDM.

Jurczyk and Xiong [22] developed many protocols in distributed environment for privacy preserving data publishing. Moreover thei7 work focused on horizontally partitioned distributed databases. Benjamin et al. [28] explored recent improvement in the area of PPDDM. They studied both privacy models and attack models in distributed environments. The attack models they found include probabilistic attack, table linkage, attribute linkage and record linkage. Kumar and Lavanya [5] focused on PPDM in the context of collaborative data publishing. They explored formal anonymity models such as k-anonymity, l-diversity and t-closeness. Besides they explored m-privacy algorithm for privacy in the presence of multi-party secure communication.

Bordoro et al. [29] reviewed big data platforms and techniques. Madhu and Ngachandrika [30] discussed missing value estimation using new paradigm with data imputation approach. Archana et al. [31] discussed about big data security by using data masking techniques. This paper has relevance with this as it exploits sanitization. More on big data and security can be found in the works of Arun et al. [32] and Madhavi an Ramana [33]. Wright et al. [34] reviewed distributed data mining protocols including Bayesian networks and BN learning protocol. Zaman and Obimbo [35] explored PPDP with respect to classification techniques. They developed a framework based on differential privacy. In this paper we focused on the privacy preserving data processing using MapReduce programming paradigm. Towards this end we proposed a methodology to sanitize data based on misusability level which is measure using misusability score computed.

## 3. PROPOSED METHODOLOGY FOR PRIVACY PRESSERVING MAPREDUCE PROGRAMMING

Here is the Comprehensive methodology for Exploring Privacy Preserving Data Mining for Big Data. It takes big data as input and produces sanitized data as output. After taking input, all attributes are considered and they are mapped to different kinds like sensitive, normal and quasi identifiers. Sensitive identifiers are identifiers that can directly disclose identity. Quasi identifiers are the identifiers prone to

inference attacks. The sensitivity level of attributes is considered. Then a misusability score is measured for all attributes to be sanitized. Misusability Score is a measure to know the vulnerability of an attribute against inference attacks. Once misusability measure is applied to attributes, the level of vulnerability against inference attacks is known. This information is used to have an adaptive and iterative process to sanitize the data. The approach is comprehensive as it can adapt to different sanitization procedures based on the misusability level. Thus it is a hybrid approach that can effectively deal with different attributes with appropriate sanitization method. As one size does not fit all the proposed methodology provides suitable sanitization mechanism for all attributes of the data set.

There are two phases in the proposed approach. First one is creating misusability measure and applying it to given dataset in order to obtain misusability score. Once misuability score is obtained it is given to the second phase which is execution model. In the execution model there are two steps involved. In the first step misusability score is used to know which level of sanitization is required. In the second step the determined sanitization technique is applied to given dataset(s) in order to generate fully sanitized dataset. Figure 1 depicst the approach.

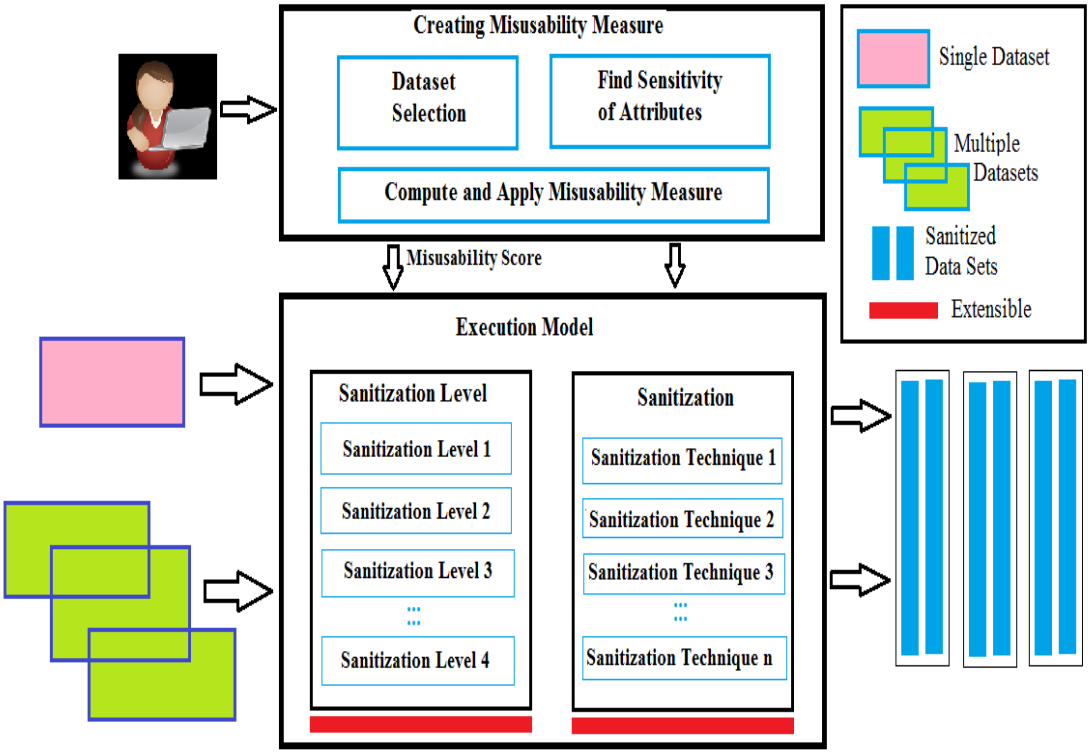


Figure 1. Architectural overview of the proposed approach

### 3.1. Creating Misusability Measure

Misusability measure is the measure used to know how much possibility is there to misuse the given dataset. This measure was first introduced by Harel et al. [6]. In this paper it is used as part of our comprehensive methodology used for protecting privacy of big data in the context of MapReduce programming paradigm. The misusability score is computed by using series of steps as shown in Figure 2.

The steps include computing raw record score (RRS), computing record distinguishing factor (RDF), computing final record score (FRS) and computing misusability score (MS). The mechanism illustrated needs a dataset as input and performs series of activities before it finally computes misusability score which is used in the proposed algorithm to determine the level of sensitization. Before employing sanitization, it is important to understand the misusability probability of given dataset to be published. Towards this end the steps are briefly described here.

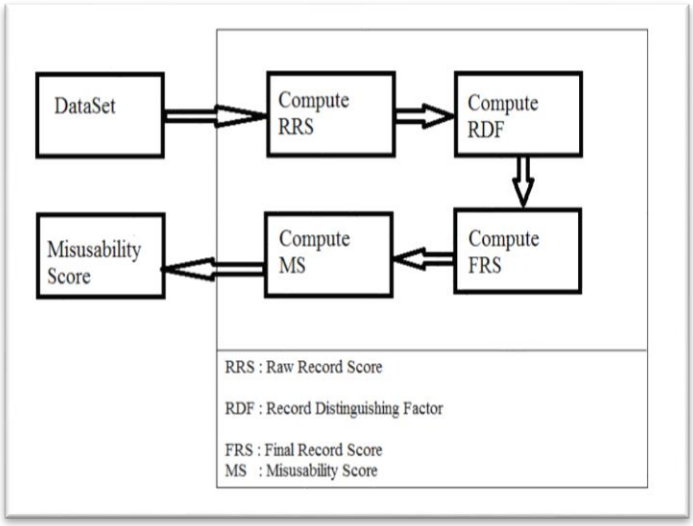


Figure 2. Overview of computing misusability score

### 3.2. Computing Raw Record Score

This is the sensitivity score of one record in the given dataset in the form of a table. For a single record i, the sum of all sensitive values is computed and that is denoted as RRSi. It is computed as follows.

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

The RRS is more when a table has more number of sensitive attributes. In the same fashion, when the table has less number of sensitive values, its RRS is low. The result of the RRS must be 1 or less than 1. It will not exceed the value 1.

### 3.3. Computing Record Distinguishing Factor

It is the measure to know how far a quasi-identifier in given dataset can reveal identify of the entity. Its value is in the range of 0.0 and 1.0. Therefore the distinguishing factor function is denoted as follows.

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

DF of a given record indicates the effort needed by an individual to know the about exact entity needed by the individual.

### 3.4. Computing Final Record Score

This measure makes use of a record's RRSi and Di. When a table is considered with r records, the final record score is computed as follows.

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

Weighted sensitivity score denoted as RSi is computed for each record. The RRSi is divided by distinguishing factor Di for doing this. Thus the maximal weighted sensitivity score FRS is computed for given table.

### 3.5. Computing Misusability Score

It is the measure needed finally which combines FRS which shows sensitivity levels of records, the number of records denoted by r, and the importance of the quantity factor x(x>=1).

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

FRS is the final record score and x is the given parameter while Di is he distinguishing factor.

### 3.6. Misusability Measure-Based Privacy Preserving Algorithm (MMPP)

We proposed an algorithm to realize the methodology presented in section 3. The algorithm reveals a hybrid approach that constitutes measurement of misusability, finding level of misusability and applying appropriate sanitization technique. This is an important step toward privacy preserving data mining on big data in distributed

---

**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 programming environment

The MMPP algorithm is takes dataset D as input and sanitizes it to produce D'. The dataset is subjected to computing misusability score so as to apply appropriate level of sanitization. After computing misusability score, the algorithm finds the level of sanitization needed. Based on the level of sanitization, specific sanitization method is employed.

## 4. EXPERIMENTAL RESULTS

The environment used for empirical study is Amazon EC2, Amazon EMR and Amazon S3. Amazon S3 is used for storing big data inputs and outputs. EMR is meant for performing MapReduce tasks which run on the EC2 instances in cluster environment.

### 4.1. Datasets Used

Four datasets are collected from UCI machine learning repository [36]. The datasets are manipulated to have more instances. The datasets collected are adult, breast cancer, census and diabetes is shown in Figure 3. As shown in Table 3, the datasets have different number of instances. The diabetes dataset is altered to have up to 200000 instances. As shown in Table 4, the memory consumption is influenced by the size of dataset. As the size increases, memory consumption is increased for processing data.

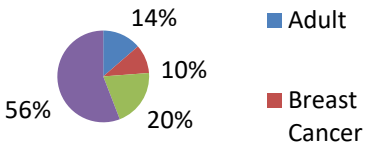


Figure 3. The datasets and percentage of instances in experiments

Table 3. Shows Datasets with Number of Instances

| Dataset | Adult | Breast Cancer | Census | Diabetes |
|---|---|---|---|---|
| No. of Instances | 48842 | 36369 | 72738 | 200000 |

Table 4. Shows Memory Consumption for Different Datasets

| | Adult | Breast Cancer | Census | Diabetes |
|---|---|---|---|---|
| Memory Consumption (MB) | 124.94 | 119.85 | 184.62 | 335.36 |

As shown in Figure 4, it is evident that the memory consumption is presented in vertical axis while the horizontal axis shows datasets used. There is clear increase in the memory consumption when number of instances increase in datasets. As shown in Table 5, the Diabetes dataset took more time for processing. In fact, it is the dataset which has highest number of instances. The results reveal that the size of dataset has its influence on the execution time. As shown in Figure 5, the Breast Cancer dataset took least time for processing. In fact, it is the dataset which has lowest number of instances. The results reveal that the size of dataset has its influence on the execution time.

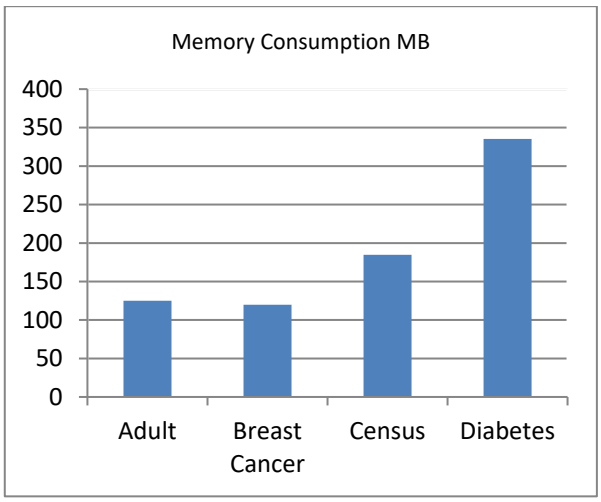


Figure 4. Shows Details of Memory Consumption

Table 5. Shows Execution Time (sec)

| Dataset | Adult | Breast Cancer | Census | Diabetes |
|---|---|---|---|---|
| Execution Time (sec) | 6.828 | 6.409 | 14.084 | 20.264 |

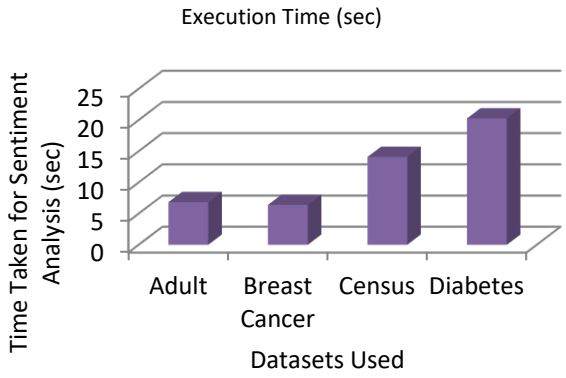


Figure 5. Execution Time for Algorithm

This paper has focused on the misusability measure based sanitization of big data. It considered different datasets and misusability score is computed as per the methodology provided. The execution time and memory consumption for each data set are provided. The results revealed that the Diabetes dataset took more time when compared with other datasets. Breast cancer dataset took least time for processing. In the same fashion, memory consumption is made. Both the metrics revealed that the data size is influencing the execution time and memory consumption. Different levels of sanitization are employed based on the misusability score computed. The results are not compared with other such works as we could not find references to misusability measure based sanitization. However, we understand that there is need for further experimental evaluation of the work. It needs to be explored with different misusability scores and sanitization levels with MapReduce programming paradigm. More evaluation and the discussion on the tradeoffs between misusability values and sanitization levels is left for our future work.

## 5. CONCLUSION AND FUTURE WORK

The problem of misuse of sensitive data has increased significantly as enterprises opt to outsource their massive amount of data, big data, to cloud for data publishing and mining to extract business intelligence. Existing sanitization techniques can be applied when level of misusability is known. This is the motivation behind this research. We introduced a comprehensive and integrated methodology for privacy preserving MapReduce processing of big data. Our methodology considers sensitivity level of dataset in order to make sanitization decisions. We computed misusability measure originally introduced by Harel et al. for more appropriate sanitization of big data. We proposed an algorithm known as Misusability Measure-Based Privacy Preserving Algorithm (MMPP) which considers level of misusability prior to choosing and application of appropriate sanitization on big data. Since the level of misusability can reveal the needed sanitization approach, we incorporated misusability score into the algorithm. Our empirical study with Amazon EC2 and EMR revealed that the proposed methodology is useful in realizing privacy preserving MapReduce programming. This research can be extended further to evaluate the framework to analyze the dynamics of misusability measure and corresponding sanitization performance.

## REFERENCES


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

[2] Apache Software Foundation. *MapReduce Tutorial.* Available: https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html. Last accessed 01 December 2016 survey paper ref here

[3] D. Radhika and D. Aruna Kumari. A Framework for Exploring Algorithms for Big Data Mining. *Indian Journal of Science and Technology*, 2016. 9(17), p1-7.

[4] V. V. Nagendra kumar and C. Lavanya. Privacy-Preserving For Collaborative Data Publishing. *IJCSIT*. 2014. 5 (3), p1-4

[5] Raymond Heatherly, Murat Kantarcioglu, and Bhavani Thuraisingham. Preventing Private Information Inference Attacks on Social Networks. *Transactions on Knowledge and Data Engineering*. 2013; 25 (8), p1-14.

[6] Gergely Acs, Claude Castelluccia and Rui Chen. Differentially Private Histogram Publishing through Lossy Compression. *International Conference on Data Mining*, 2012: p1-10.

[7] Rui Chen, Benjamin C. M. Fung, Bipin C. Desai and Nériah M. Sossou. Differentially Private Transit Data Publication: A Case Study on the Montreal Transportation System. *ACM*, 2012: p1-9.

[8] Mina Askari, Reihaneh Safavi-Naini and Ken Barker. An Information Theoretic Privacy and Utility Measure for Data Sanitization Mechanisms. *ACM*, 2012: p1-12.

[9] Nikunj H. Domadiya and Udai Pratap Rao. Hiding Sensitive Association Rules to Maintain Privacy and Data Quality in Database. *IEEE*, 2012: p1-6.

[10] Sébastien Canard and Roch Lescuyer. Protecting Privacy by Sanitizing Personal Data: a New Approach to Anonymous Credentials. *ACM*, 2013: p1-12.

[11] Chun-Wei Lin, Tzung-Pei Hong, Chia-Ching Chang, and Shyue-Liang Wang. A Greedy-based Approach for Hiding Sensitive Itemsets by Transaction Insertion. *Journal of Information Hiding and Multimedia Signal Processing*. 2013; 4 (4): p1-14.

[12] Lwin Khin Shar and Hee Beng Kuan Tan. Predicting Common Web Application Vulnerabilities from Input Validation and Sanitization Code Patterns. *ACM*, 2012: p1-4.

[13] Qian Xiao, Rui Chen and Kian-Lee Tan. Differentially Private Network Data Release via Structural Inference. *ACM*, 2014: p1-10.

[14] SébastienGambs, Marc-OlivierKillijian and MiguelNúñezdelPradoCortez. De-anonymization attack on geolocated data. *Journal of Computer and System Sciences*, 2014: p1-18.

[15] Jia-Dong Zhang, Gabriel Ghinita and Chi-Yin Chow. Differentially Private Location Recommendations in Geosocial Networks. *IEEE*, 2014; p1-10.

[16] David Sánchez, Montserrat Batet and Alexandre Viejo. Detecting Term Relationships to Improve Textual Document Sanitization. *Pacific Asia Conference on Information Systems*, 2013: p1-15.
[17] Hao Sun, Xiangyu Zhang, Chao Su and Qingkai Zeng. Efficient Dynamic Tracking Technique for Detecting Integer-Overflow-to-Buffer-Overflow Vulnerability. *ACM*, 2015: p1-12.
[18] Boyang Li, Mark Grechanik and Denys Poshyvanyk. Sanitizing And Minimizing Databases for Software Application Test Outsourcing. *IEEE*, 2014: p1-10.
[19] Ori Heffetz and Katrina Ligett. Privacy and Data-Based Research. *Springer*, 2014; p75–98.
[20] Chris Clifton.Privacy Preserving Distributed Data Mining. *Computer Sciences*. 2001: p1-10.
[21] S.Selva Rathna, Dr. T. Karthikeyan. Survey on Recent Algorithms for Privacy Preserving Data mining. *Computer scince*. 2015; 6(2): p1-6.
[22] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold and Aaron Roth. Preserving Statistical Validity in Adaptive Data Analysis. *Computer Sciences.* 2015: p1-29.
[23] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann PitassiOmer Reingold and Aaron Roth. the reusable holdout: Preserving validity in adaptive data analysis. *Computer Sciences*. 2015; 349: p1-4.
[24] Chris Clifton, Murat Kantarcioglu, Xiaodong Lin, Michael and Y. Zhu. Tools for Privacy Preserving Distributed Data Mining. *IEEE*. 2002; 4(2): p1-7.
[25] V. Baby and N. Subhash Chandra. Privacy-Preserving Distributed A Survey Data Mining Techniques. *International Journal of Computer Applications*. 2016; 143(10): p1-5.
[26] Pawel Jurczyk and Li Xiong. Privacy-Preserving Data Publishing for Horizontally Partitioned Databases. *IEEE*, 2008: p1-2.
[27] Benjamin C. M. Fung, Ke Wang, Rui Chen and and Philip S. Yu. Privacy-Preserving Data Publishing: A Survey of Recent Developments. *ACM*. 2010; 42 (4): p1-53.
[28] Salisu Musa Borodo, Siti Mariyam Shamsuddin and Shafaatunnur Hasan. Big Data Platforms and Techniques. *Indonesian Journal of Electrical Engineering and Computer Science*. 2016; 1, p191 -200.
[29] Madhu G and Nagachandrika G. A New Paradigm for Development of Data Imputation Approach for Missing Value Estimation. *International Journal of Electrical and Computer Engineering*. 2016; 6: p3222 – 3228.
[30] Archana RA, Ravindra S Hegadi and Manjunath TN. A Big Data Security using Data Masking Methods. *Indonesian Journal of Electrical Engineering and Computer Science*. 2017; 7 , p449 -456.
[31] Sachin Arun Thanekar, K. Subrahmanyam and A. B. Bagwan. Big Data and MapReduce Challenges, Opportunities and Trends. *International Journal of Electrical and Computer Engineering*. 2016. 6.
[32] Dasari Madhavi and B.V.Ramana. De-Identified Personal Health Care System Using Hadoop. *International Journal of Electrical and Computer Engineering*. 2015; 5: p1492-1499.
[33] Rebecca N. Wright, Zhiqiang Yang and Sheng Zhong. Distributed Data Mining Protocols for Privacy: A Review of Some Recent Results. *IEEE*. 2006; 0 (0), p1-13.
[34] A N K Zaman and Charlie Obimbo. Privacy Preserving Data Publishing: A Classification Perspective. *IJACSA*. 2014; 5 (9): p1-6.
[35] UCI. UCI Machine Learning Repository. Available online at: https://archive.ics.uci.edu/ml/index.php. [accessed on: 20 April 2017]