Taylor & Francis
Taylor & Francis Group

Applied Artificial Intelligence
An International Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

Privacy Preserving Parallel Clustering Based Anonymization for Big Data Using MapReduce Framework

Josephine Usha Lawrance & Jesu Vedha Nayahi Jesudhasan

To cite this article: Josephine Usha Lawrance & Jesu Vedha Nayahi Jesudhasan (2021) Privacy Preserving Parallel Clustering Based Anonymization for Big Data Using MapReduce Framework, Applied Artificial Intelligence, 35:15, 1587-1620, DOI: 10.1080/08839514.2021.1987709

To link to this article: https://doi.org/10.1080/08839514.2021.1987709

Published online: 17 Oct 2021.

APPLIED ARTIFICIAL INTELLIGENCE
2021, VOL. 35, NO. 15, 1587–1620
https://doi.org/10.1080/08839514.2021.1987709

Taylor & Francis
Taylor & Francis Group

Privacy Preserving Parallel Clustering Based Anonymization for Big Data Using MapReduce Framework

Josephine Usha Lawrance[a] and Jesu Vedha Nayahi Jesudhasan[b]

[a]Department of Information Technology, St. Xavier's Catholic College of Engineering, Chunkankadai, Tamil Nadu, India; [b]Department of Computer Science and Engineering, Anna University Regional Campus – Tirunelveli, Tirunelveli, Tamil Nadu, India

ABSTRACT

Big data refers to a massive volume of data collected from heterogeneous data sources including data collected from Internet of Things (IoT) devices. Big data analytics is playing a crucial role in extracting patterns that would benefit efficient and effective decision making. Processing this massive volume of data poses several critical issues such as scalability, security and privacy. To preserve data privacy, numerous privacy-preserving data mining and publishing techniques exist. Data anonymization utilizing data mining techniques for preserving an individual's privacy is a promising approach to prevent the data against identity disclosure. In this paper, a Parallel Clustering based Anonymization Algorithm (PCAA) is proposed, and the results prove that the algorithm is scalable and also achieves a better tradeoff between privacy and utility. The MapReduce framework is used to parallelize the anonymization process for handling a huge volume of data. The algorithm performs well in terms of classification accuracy, F-measure, and Kullback–Leibler divergence metrics. Moreover, the big data generated from heterogeneous data sources are efficiently protected to meet the ever-growing requirements of the application.

ARTICLE HISTORY
Received 31 January 2021
Revised 23 September 2021
Accepted 27 September 2021

1.Introduction

Due to recent technological development, the amount of data collected from cyber, physical and human worlds is growing day by day. Nowadays, connecting of people with each other through the usage of various cyber society components leads to the generation of large volume of data. As indicated by the report of International Data Corporation (IDC) (Sweeney 1997; Orsini et al. 2017), the size of the data is from exabyte to zettabyte, expanding at regular intervals. Almost every researcher in information sciences, policy and decision makers in government and enterprises is trying to explore this huge amount of data for making critical decision and planned business moves.

CONTACT Josephine Usha Lawrance ✉ josephineusha@gmail.com 🖃 Department of Information Technology, St. Xavier's Catholic College of Engineering, Chunkankadai, Tamil Nadu, India

In a patient monitoring system, personal health record (PHR) acts a crucial platform for health information exchange. Data mining techniques helps investigate hidden relations in the data and assists the data owner with useful insights. Furthermore, this should be shared with outside parties for more investigations. It leads to significant loss in information privacy. The most important technical, legal, ethical and social challenge in information privacy is to prevent the exposure of personally identifiable information (PII) while sharing the information to the other. According to the privacy law (HIPAA 1999), the health records kept at any medical clinic ought to be kept confidential.

Privacy-preserving data publishing (PPDP) is a concept that provides various tools and techniques for preserving data privacy while publishing the data over the Internet (Aggarwal and Yu, 2008; Fung et al. 2010; Fahad et al. 2014). The significant strategies utilized in the field of privacy-preserving data mining or data publishing are data anonymization (Sweeney 1997, 1998, 2002a, 2002b; Samarati and Sweeney 1998; Samarati 2001), data randomization (Chen and Liu 2009, 2011; Chen, Sun, and Liu 2007; Islam and Brankovic 2011) and cryptography (Liu, Huang, and Liu 2015; Pinkas 2002). Data anonymization is a promising methodology for anonymizing the records with the end goal that k individuals become indistinguishable from one another. It can be achieved by two approaches called generalization and suppression (Reddy and Aggarwal 2015). Many variations of anonymization exist, namely, *k*-anonymity (Sweeney, 2002b), *l*-diversity (Meyerson and Williams 2004), *(α, k)*-anonymity (Chi-Wing Wong et al. 2006; Wong et al. 2006; Raymond et al, 2009), and *t*-closeness (Li et al. 2007; Wong, Li, and Fu et al. 2009). When the values of sensitive attributes have a little variety, the k-anonymized data set is vulnerable to extreme privacy attacks, including similarity invasion, homogeneity attack, background knowledge attack, and probabilistic inference attack.

l-diversity (Machanavajjhala et al. 2006, 2007; Hongwei and Weining, 2011) is a group-based privacy anonymization, where both generalization and suppression methods achieve privacy such that every given record is similar to other records in the table of at least k-1. For the sensitive attribute, it needs at least *l* distinct values in each equivalence class. The technique of *l*-diversity suffers from an attack of skewness and similarity, so it is insufficient to avoid disclosure of attributes.

To overcome the limitations of the principle of *l*-diversity, the *t*-closeness (Li et al. 2007) principle is introduced. Here, the sensitive attribute distribution in each equivalence class is similar to sensitive attribute distribution in the original data set. The main benefit of using *t*-closeness is that it solves the problem of attribute disclosure. Traditional strategies are vulnerable to common attacks and struggle to achieve a better trade-off between privacy and utility as well. Another important issue is scalability while handling large volume of data (Abid 2016; Mehmood et al. 2016). *k*-anonymization and cryptographic techniques would not be scalable for big data, whereas perturbation techniques (Chen, Sun, and Liu 2007) ensure scalability by compromising the data utility.

A hybrid technique based on anonymization and clustering is most appropriate for ensuring the privacy of big data. (G,S) clustering (Nayahi and Kavitha 2015) overcomes the risk of a similarity attack but results in a long execution time when the number of cluster increases. The modified version is called KNN-(G,S) clustering (Nayahi and Kavitha 2017), where the k-nearest neighbors technique is used to achieve high degree of privacy with minimal information loss by finding the K clusters with S diverse sensitive values for its sensitive attribute.

In general, anonymization is one of the earliest approaches and it is more promising to preserve the identity of individuals against attacks. However, these algorithms should be modified to run in a parallel manner. Hadoop MapReduce (Al-Zobbi, Shahrestani, and Ruan 2017; Dean and Ghemawat 2004; Lammel 2008; Panagiotis et al. 2019; Qian et al. 2018; Shafer, Rixner, and Cox 2010; Shvachko et al. 2010) simplifies the processing of big data in parallel in a scalable, efficient and fault-tolerant way on large scale clusters of commodity hardware. In this paper, a Parallel Clustering based Anonymization Algorithm (PCAA) is introduced, which modifies the existing KNN-(G,S) clustering algorithm to parallelize the anonymization process by utilizing the Hadoop MapReduce framework. According to the experimental results, the proposed algorithm performs well in terms of scalability and achieves better trade-off between privacy and utility while handling massive amount of data.

The rest of the paper is arranged as follows: in Section 2, the related works are given. In Section 3, different methods of privacy protection and their comparisons are discussed. Section 4 introduces the proposed parallel clustering based anonymization algorithm for privacy preservation. The experiment outcomes of the proposed algorithms and their comparisons are illustrated in Section 5. Finally, Section 6 provides the concluding remarks.

Related Work

In this section, different techniques used for ensuring the privacy of personally identifiable information and their merits and demerits are identified. The merits and demerits of the clustering-based anonymization algorithm and some of the new strategies based on anonymization and data mining approaches are also discussed for ensuring the privacy of information.

Definition 1: (*k*-Anonymity). It is the process of transforming the records with the goal that *k* individuals become indistinguishable from one another, where *k* is the anonymization threshold.

A utility-based anonymization (Xu et al. 2006a, 2006b) is a simple framework for producing a high-utility anonymized data set. It combines the features of both local recoding and global recoding for preserving privacy. The hybrid recoding technique (LeFevre, DeWitt, and Ramakrishnan 2005, 2006) takes advantage of both global and local recoding techniques to ensure the privacy. An enhanced model of *k*-anonymity called (α, k)-anonymity (Raymond et al. 2006) achieves privacy-preserving data publishing by considering both *k*-anonymization and *α*-deassociation property (Amit and Neeraj 2016). This not only protects individual identification but also protects sensitive relationships by hiding multiple sensitive values using a simple k-anonymity model. Achieving (α, k) anonymization itself is NP-hard (Raymond et al. 2006). A two phase clustering algorithm (Zhang et al. 2015) uses an anonymization technique to achieve the data privacy on cloud environment.

Homomorphic encryption (Lauter, Naehrig, and Vaikuntanathan 2011; Monique, Claude, and Pushkar 2013; Ogburn et al. 2013; Hayward and Chiang 2015; Potey et al. 2016; Rahul et al. 2017) enables the user to conduct encrypted data operations without exposing the original data so that both security and privacy are supported. Secured multiparty computation (Lindell and Pinkas 2009) is another cryptographic technique to accomplish the privacy of data. Identity-based anonymization (Govinda and Sathiyamoorthy 2012; Sedayao, Bhardwaj, and Gorade 2014) anonymize the data in a way to ensure the confidentiality of the user using the system.

MapReduce-based anonymization (Chamikara et al. 2019) uses a MapReduce framework for processing large volumes of data. The data is automatically split by the MapReduce (Dean and Ghemawat 2004) system into equal sized chunks. Each split can be assigned to separate mappers, where it is converted to key-value pairs. One reducer is assigned to a pair with the same key. The final result can be obtained from the reducer.

Parallelization (Jain, Gyanchandani, and Khare 2016) of k-anonymity can be achieved by using the Hadoop MapReduce concept. It works in a top down manner by splitting each data set into two equally sized data sets, then anonymizing the data sets by taking the attribute's maximum and minimum value and adjusting the explicit value to a min–max value. The methods such as suppression and sampling are introduced in k-anonymization to support the concept of differential privacy (Li, Qardaji, and Su 2012). Both differential privacy and k-anonymity are combined to enhance data utility in micro-aggregation-based k-anonymization (Soria-comas et al. 2014, 2015).

Recent research shows that with the introduction of data mining techniques in data anonymization, significant improvements in data utility can be achieved. A clustering-based anonymization algorithm called the (G,S) clustering algorithm (Nayahi and Kavitha 2015) safeguards the data against both identity disclosure and disclosure of attributes. It is modified using the *k*-Nearest Neighbors (*k*-NN) approach called KNN-(G,S) clustering algorithm (Nayahi and Kavitha 2017), protects sensitive information against various attacks and attains high degree of privacy with very low information loss. A multi-dimensional sensitivity-based anonymization (Al-Zobbi, Shahrestani, and Ruan 2017; Venugopal and Vigila 2018) utilizes SQL-like Hadoop ecosystems and Pig Latin and UDF anonymization tools to secure the big data and minimizes the information loss. A novel data anonymization algorithm based on chaos and perturbation (Eyupoglu et al. 2018) utilizes a chaotic function to ensure privacy and utility preservation while sharing the big data (Tankard, 2012). A clustering-based privacy preservation probabilistic model for big data (Saira Khan et al. 2019) utilizes k-anonymity, fuzzification and minimum perturbation concepts to achieve better privacy. An anonymization algorithm based on homeomorphic data space transformation (Anastasiia et al. 2021) uses feed forward artificial neural nets to learn the neural networks to protect the privacy of data.

Table 1 below shows the comparison between various privacy-preserving approaches with its merits and demerits. The approaches based on anonymization prevent the similarity attack and probability inference attack and those are not scalable. Hence, parallel computation of these algorithms would be beneficial to deal with scalability issues and also achieve a better trade-off between privacy and usefulness.

Recently, hybrid approaches that integrate anonymization with data mining techniques are appropriate for handling large volumes of data. MapReduce (Hadoop 2009; Shvachko at al. 2010; Shafer, Rixner, and Cox 2010; Gu et al. 2014) frameworks are also helpful to execute the privacy-preserving algorithms in a parallel manner, hence improving scalability.

Preliminaries

By distributing or publishing raw data containing un-aggregated information about individuals, many organizations are gradually sharing data. The data set released by organization can be described by different types of attributes. The various terms used in privacy preservation and its definitions are given below:

Definition 2: (Quasi-Identifier). These are attributes that when combined with other external data can be used to identify the individuals. For instance, attributes like age, gender, and zip code are *QI* attributes.

Definition 3: (Sensitive Attribute). Attributes like disease and salary that are sensitive and hence protected from disclosure.

Definition 4: (Nonsensitive Attribute). It represents the attributes other than identifier, quasi-identifier and sensitive attribute.

Table 1. Comparison of privacy-preserving techniques.

Sl. No.	Frameworks	Method Used	Merits	Demerits
1.	Sweeney 1997	k-anonymity	Simple and easy to understand Protect identity disclosure.	Difficult to protect the sensitive relationships in a data set and disclosure of attributes. Susceptible to homogeneity and background knowledge attacks
2.	Machanavajjhala 2006	*l* – Diversity	Handle attribute disclosure.	Vulnerable to attacks like skewness and similarity invasion insufficient to prevent attribute disclosure
3.	Li et al. 2007	*t*-closeness	Handle attribute disclosure.	The probability of re-identification increases with size and variety of data increases.
4.	Nayahi and Kavitha 2015	(G,S) Clustering for privacy preservation	Resolve the probability of similarity attack. Achieves high level of privacy with limited loss of information.	Computational complexity possible with generalization.
5.	Jain, Gyanchandani, and Khare 2016	Differential privacy	Robustness against powerful adversaries Applicable to a wide range of data analysis problems.	Better privacy can be achieved with high distraction.
6.	Lauter, Naehrig, and Vaikuntanathan 2011	Partially homomorphic encryption	Computation is performed in the encrypted text. More secure.	Not more practical. Lot of computational overhead.
7.	Hayward et al. 2015	Fully homomorphic encryption	Computation is performed in the encrypted text. More secure.	Fully homomorphic encryption runs slow. Lot of computational overhead.
8.	Raymond etal. 2006	*(a, k)* – Anonymity	Protect both identity and sensitivity relationships in data.	NP-Hard problem. The execution time and distortion ratio depend on the value *a*.
9.	Amit et al. 2016	*k*-anonymity with privacy key	Achieves two levels of security.	More space is required to store privacy key.
10.	Sedayao, Bhardwaj, and Gorade 2014	Identity-based encryption	Ensures the confidentiality of the user.	Less efficient in terms of privacy and utility.
11.	Xu et al. 2006a	Utility-based anonymization	Produces high utility anonymized dataset. Less information loss.	Quality of analysis is compromised with the increased level of utility.
12.	LeFevre, DeWitt, and Ramakrishnan 2006	Mondrian multidimensional *k*-anonymity	Performs better in terms of runtime and less information loss.	NP hard.
13.	Can Eyupoglu et al. 2018	Anonymization based on chaos and perturbation	Robustness against powerful adversaries. Applicable to a wide range of data analysis problems.	Better privacy can be achieved with high distraction.

(*Continued*)

Table 1. (Continued).

Sl. No.	Frameworks	Method Used	Merits	Demerits
14.	Dean et al. 2004	*k*-anonymization with MapReduce	Supports parallelization of privacy-preserving algorithms. Improves scalability.	Lot of computational overhead.
15.	Goyal et al. 2006	Attribute-based anonymization.	Ensures fine-grained sharing of encrypted data over the Internet.	Complexity is increased.
16.	Soria-comas et al. 2015	*t*-closeness through micro-aggregation based anonymization.	Enhanced utility.	Supports only numerical data.
17.	Al-Zobbi, Shahrestani, and Ruan 2017	Multi-dimensional sensitivity-based anonymization using Pig Latin and UDF anonymization tools	Reduces loss of information.	Computational overhead.
18.	Venugopal et al. 2018	Multi-dimensional anonymization	Accuracy and execution speed are much faster.	More complexity.
19.	Saira Khan et al. 2019	*k*-anonymity and fuzzification	Achieves maximum privacy.	Accuracy is less concerned with the problem of reconstruction of original data.
20.	Anastasiia et al. 2021	Homeomorphic data space transformation uses feed forward artificial neural nets	Achieves better privacy.	More time to train the model.
21.	Chamikara et al. 2020	Optimal geometric transformations and *Φ–separation*	Resistance to data reconstruction attacks.	Not scalable

Table 2. Sample data set.

Name	Age	Gender	ZIP Code	Disease
Joy	33	F	45678	Stroke
Nalini	35	F	45678	Coronary disease
Shylu	36	F	45612	Blood pressure
Abijith	39	M	45615	Blood pressure
Jonna	44	M	45609	Gastritis
Balu	43	M	45606	Diabetes
Ratheesh	46	M	45605	Pneumonia
Jony	48	M	45607	Pneumonia

The sample data set used to demonstrate the concepts of anonymization is shown in Table 2. The data set has five features: name, age, gender, ZIP code, and disease, with name being the key or identifier attribute, disease being the sensitive attribute, and age, gender, and ZIP code being the quasi-identifier (*QI*) attributes. The attacker uses these *QI* attributes to reveal the sensitive attribute value of a person. Consider that there exists an intruder 'X' having knowledge of a male person of age 46 residing in his locality and get treatment from a particular hospital. Suppose he gets access to the hospital's published data, he may believe that the person may have pneumonia. This form of attack is known as the *disclosure of identity* or *linking attack*.

Definition 5: (Equivalence Class). It denotes a set of records in a table that have the same values for the *QI* attributes.

Before k-anonymization, the attribute '*name*' should be eliminated because it uniquely identifies the individual. Suppression and generalization are the two steps involved. Suppression would replace all the *QI* attribute values that do not have similar values to '*'. Generalization transforms the quasi-identifiers to more general values. The 2-anonymous table (Aggarwal et al. 2005) constructed from the given data set is shown Table 3, having four equivalence classes with identical *QI* attribute values; therefore, it is resilient to linking attack.

Definition 6: (*l*-diversity). It assures that in each anonymized group, the diversity of sensitive attribute values is at least *l*.

Table 3. 2-Anonymous group table.

Age	Gender	ZIP Code	Disease
30–35	F	4567*	Stroke
30–35	F	4567*	Coronary disease
36–40	*	4561*	Blood pressure
36–40	*	4561*	Blood pressure
40–45	M	4560*	Gastritis
40–45	M	4560*	Diabetes
45–50	M	4560*	Pneumonia
45–50	M	4560*	Pneumonia

In the above case, the sensitive attributes in the second and forth equivalence classes are identical, which leads to *homogeneity attack*. If the anonymized group contains identical sensitive attribute values for all records, homogeneity attack will happen. This can be eliminated by using the *l*-diversity principle. It assures that in each anonymized group, the diversity of sensitive attribute values is at least *l*. The four-anonymous three-diverse equivalence class groups, each with three different sensitive values ($l = 3$), are shown in Table 4.

Definition 7: (*S*-diversity principle). It ensures that sensitive attributes in all equivalence classes have all feasible diverse values, where '*S*' denotes the number of distinct sensitive attributes values.

Although the sensitive values in an anonymized group are not identical, they may be similar; it can lead to sensitive attribute disclosure. Table 4 shows that all records in the first equivalence class have the same values for sensitive attributes. An attacker can easily infer that the particular person has some problems associated with the heart. This form of attack is called *similarity attack*. The *S*-diversity principle (Nayahi and Kavitha 2017) efficiently overcomes the similarity attack by ensuring that all equivalence classes have all distinct sensitive attribute values.

Hadoop Distributed File System (HDFS)

Hadoop (Dean and Ghemawat 2004; Lammel 2008; Panagiotis et al. 2019; Qian et al. 2018; Saadoon et al. 2021; Shafer, Rixner, and Cox 2010; Shvachko et al. 2010) is an open-source implementation for reliable, scalable, distributed computing and data storage. It allows the user to write and execute applications that runs on large volumes of data. MapReduce is the programming paradigm that involves two processes, namely, Mapper and Reducer. The map task is an initial step that takes input data and converts it into a series of key/value pairs. Job Reduce combines those tuples of data into a smaller collection of tuples.

Table 4. Anonymization groups.

Age	Gender	State	Disease
30–40	*	456**	Stroke
30–40	*	456**	Coronary disease
30–40	*	456**	Blood pressure
30–40	*	456**	Blood pressure
40–45	M	4560*	Gastritis
40–45	M	4560*	Diabetes
45–50	M	4560*	Pneumonia
45–50	M	4560*	Pneumonia

Parallel Clustering Based Anonymization Algorithm (PCAA)

According to this algorithm, each record in the input data set *DS* can be represented in the Hadoop distributed file system as a <key, value> pair, where each *key* represents the combination of quasi-identifies(*QI*) and the *value* corresponds to the content of the tuple. Figure 1 shows the MapReduce framework (Panagiotis et al. 2019) for the first level mapper and reducer. The input data set is divided into DS_1, DS_2, . . ., DS_k chunks based on k-means clustering and broadcasted to all mappers (Sowmya and Nagaratna 2016). Let the number of mappers be *n* and the number of

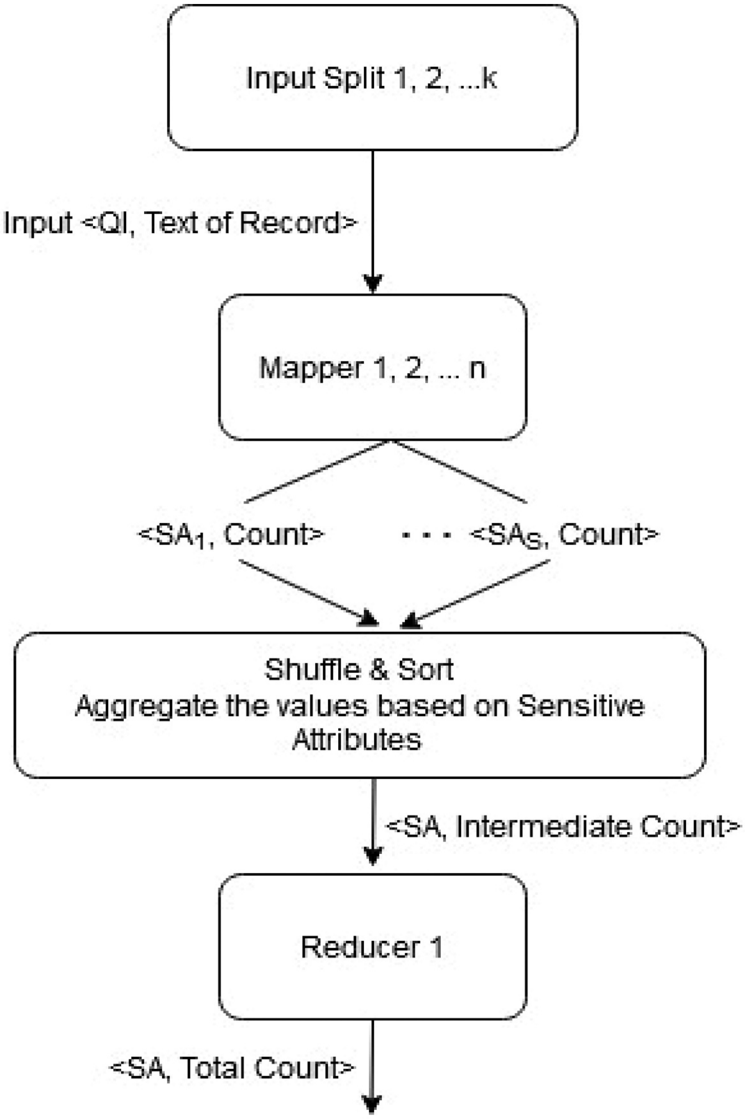


Figure 1. MapReduce framework.

reducers be *n*/2. The mapper will sort the chunks based on sensitive attribute values and form <*key-value*> pairs, where *key* denotes the sensitive attribute and *value* represents the sensitive attribute count. The reducer will merge all the sorted results and find the total count for each sensitive attribute.

Definition 8: (*NOC Cluster*). Let *DS* be the given data set, DS_1, DS_2, DS_3, . . . DS_k be the partitions based on sensitive attribute values, D_{min} be the partition with the smallest number of records, *k* be the anonymization parameter, and *S* be the number of distinct sensitive attributes in a data set. Then, the *NOC* number of nearest neighbor clusters C_i *(i=1,2,3 . . . NOC)* can be constructed by adding *KN* $(KN=|DS_i|/NOC)$ number of nearest neighbor instances from each sensitive value partitions, thereby satisfying the *S*-diversity principle.

Using the values of DS_{min}, *k*, and *S*, the number of clusters (*NOC*) formed is calculated. Then, the algorithm determines the chunks with which the cluster belongs and finds the *KN* records to be added to each cluster based on the value of the sensitive value subgroup (*DSi*) and the number of cluster (*NOC*). Accordingly, records with unique sensitive attribute values were fairly distributed among all equivalence groups, resulting in a cluster with *S* different sensitive values.

Algorithm 1 denotes the map function for the second level MapReduce program for anonymization. The input values such as *NOC* number of clusters, dataset, and quasi-identifiers are given before invoking the map function. The anonymized cluster can be found by replacing *QI* attributes into the centroid values of each *QI* attributes.

```
Algorithm 1: AnonymizationMapper
Input: Initial Clusters C, QI
Data set, DS
Output: A set of <key, value> pairs, with value indicating the Anonymized
        Cluster
For each record in Cluster C

(1) Parse the string value.
(2) Split the string into number of columns.
    (a) $\forall_{\text{attribute}}$ A in QI
          i. Calculate centroid of attribute A.
         ii. Compute the anonymized cluster by replacing quasi-identifiers
             into centroid value.
        iii. Save the result in the ‘value’ variable.
(3) Return the pair of key and value.
```

```
Algorithm 2: AnonymizationReducer
  Input: Anonymized Clusters
  Data set, DS
```

```
Output: A set of <key, value> pairs, with value indicating the privacy
        preserved data set ($DSp$).

(1) Sort the anonymized cluster based on ‘key’ value.
(2) Combine all anonymized clusters to form final privacy preserved data set.
(3) Save the result in the ‘value’ variable.
(4) Return the pair of key and value.
```

Algorithm 2 denotes the code for a second-level reduce function for anonymization. The output of the map function is given as the input for the reduce function. The reduce function acts like a combiner for producing a privacy preserved data set. Figure 2 shows the complete diagram of the parallel big data clustering algorithm.

Algorithm 3: Parallel Clustering based Anonymization Algorithm (PCAA)

```
Input: $DS$, $SensAttr$, $QI$, and $k$
Output: Output data set, $DSp$

(1) $S \leftarrow$ Unique Sensitive Attribute Count $SensAttr$.
(2) Apply K-means clustering and partition the data set $DS$ – >$DS_1$, $DS_2$, . . . .
    $DS_k$, where $k$ is the smallest sensitive attribute count.
(3) Assign $n$ mappers (M) and $n/2$ reducers (R).
(4) $\forall_{\text{mappers}} M_i$, assign $M_i \leftarrow DS_i$, $i = 1,2, \ldots, n$.
(5) $\forall_{\text{mappers}} M_i$

    (a) Sort the cluster $DS_i$ based on sensitive attribute values.
    (b) Form key-value pairs, where $key \leftarrow$ sensitive attribute, $value \leftarrow$ sensitive attribute count.
(6) $\forall_{\text{reducer}} R_i$, $i = 1,2, \ldots, n/2$.

    (a) Combine all mapper outputs $M_i$, $i = 1, 2, \ldots, n$.
    (b) Find the entire count of sensitive attribute in data set, $DS$.
(7) $DS_{min} \leftarrow$ Group that contains smallest count for the sensitive attribute.
(8) $DS_{rem} \leftarrow$ All the remaining groups.
(9) if $k \leq S$

        Invoke the procedure $ClusterFormation1(DS_{min}, DS_{rem}, k, S)$

10. else

        Invoke the procedure $ClusterFormation2(DS_{min}, DS_{rem}, k, S)$

(11) $\forall_{\text{clusters}} C_i$, assign $M_i \leftarrow C_i$, $i = 1, 2, \ldots, NOC$
```

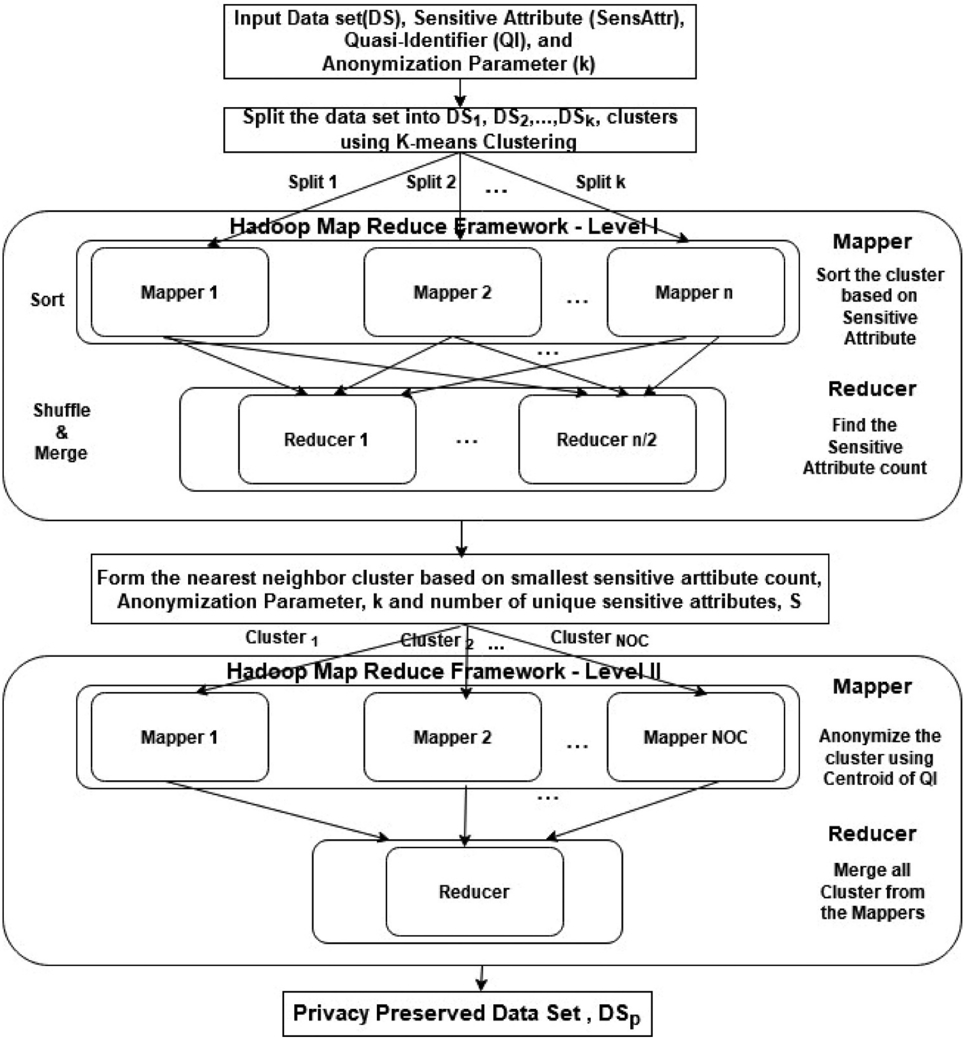

Input Data set(DS), Sensitive Attribute (SensAttr), Quasi-Identifier (QI), and Anonymization Parameter (k)
Split the data set into DS1, DS2,...,DSk, clusters using K-means Clustering
Split 1
Split 2
...
Split k
Hadoop Map Reduce Framework - Level I
Sort
Mapper 1
Mapper 2
...
Mapper n
Mapper
Sort the cluster based on Sensitive Attribute
Shuffle & Merge
Reducer 1
...
Reducer n/2
Reducer
Find the Sensitive Attribute count
Form the nearest neighbor cluster based on smallest sensitive arttibute count, Anonymization Parameter, k and number of unique sensitive attributes, S
Cluster 1
Cluster 2
...
Cluster NOC
Hadoop Map Reduce Framework - Level II
Mapper 1
Mapper 2
...
Mapper NOC
Mapper
Anonymize the cluster using Centroid of QI
Reducer
Reducer
Merge all Cluster from the Mappers
Privacy Preserved Data Set , DSp

Figure 2. Privacy-preserving MapReduce architecture.

```
(a) $\forall_{\text{mappers}}\ M_i$
    (i) $\forall_{\text{clusters}}\ C_i$ in $M_i$
        a. $\forall_{\text{attribute}}\ A$ in $QI$
            i  Calculate centroid of attribute A.
            ii Use centroid to replace all the values of A.
(b) $\forall_{\text{reducer}}\ Ri$
    (i) Combine all $C_i$ in $C$ to produce a data set $DSp$.
(12) Return $DSp$
```

Cluster Formation Case 1: ($K \leq S$)

Let DS_{min} be the minimum sensitive attribute count, and it determines the values of *NOC*. The *NOC* number of single element clusters, namely, C_1, C_2, . . ., C_{NOC}, is constructed by distributing one record from DS_{min} to all *NOC* number of clusters. One nearest neighbor instance from DS_{rem} is included in every cluster. It is calculated on the basis of the distances of each cluster from the centroid, adding the *KN* number of instances of each cluster (Nayahi and Kavitha 2017). The Euclidean distance metric (Han and Kamber 2006) as in Eq. (1) is used to find the nearest neighbor distances.

$$\sqrt{}(x_1 - y_1)^2 + \ (x_2 - y_2)^2 + \ \ldots \ . \ + (x_n - y_n)^2 \tag{1}$$

Algorithm 4: *ClusterFormation1(DS_{min}, DS_{rem}, k, S)*

```
Input: $DS_{min}$, $DS_{rem}$, k, S
Output: C, Cluster
(1) Let NOC be the total number of clusters formed.
(2) if $k \leq S$
    (a) $NOC \leftarrow DS_{min}$.
    (b) Construct $C_i$ ($i = 1,2, \ldots, NOC$) clusters with a single element from $DS_{min}$.
    (c) $\forall_{\text{clusters}}$ $C_i$ in C
        i. Add the nearest neighbor cluster from $DS_{rem}$ to each cluster Ci.
    (d) $\forall_{\text{clusters}}$ Ci in C
        i. Calculate centroid Gi.
        ii. Calculate KN nearest neighbor $KN = |DS_i|/NOC$.
    (e) $\forall_{\text{clusters}}$ Ci in C
        i. $\forall_{\text{group}}$ $DS_i$ in $DS_{rem}$
            a Determine the KN nearest neighbors based on centroid $C_i$ from group $DS_i$.
            b $C_i \leftarrow$ KN number of nearest neighbors.
        iii.Eliminate instances to be added from $DS_i$.
(3) Return C.
```

Cluster Formation Case 2: ($K > S$)

Based on *k*, *S*, and DS_{min} values, calculate the clusters to be formed. It has two cases: in the first case, we consider $NOC>0$, and it equally assigns the instances of DS_{min} to all *NOC* number of clusters using split point. One nearest neighbor instance from DS_{rem} is included in every cluster. The calculation of the *KN* number of nearest neighbors and its inclusion can be done as in Case 1, and in the second case, we consider $NOC = 0$; here, the *AddlInst* number of additional instances is randomly chosen and added to DS_{min} to form at least one cluster (Nayahi and Kavitha 2017).

```
Algorithm 5: ClusterFormation2($DS_{min}$, $DS_{rem}$, $k$, $S$)
Input: $DS_{min}$, $DS_{rem}$, $k$, $S$
Output: Cluster C

(1) $NOC \leftarrow$ card{$DS_{min}$}/($k/S$) = card{$DS_{min}$ *$S$}/$k$.
(2) if $NOC$ ! = 0

  (i) $Split \leftarrow round(k/S)$
  (ii) Divide $DS_{min}$ into $Split$ number of subgroups.
  (iii) $\forall_{\text{subgroups}}$ in $DS_{min}$
      (a) $\forall_{\text{cluster}}$ $C_i$ in $C$
          i. $C_i \leftarrow$ one instance from $DS_{min}$.
          ii. Add the nearest neighbor cluster from $DS_{rem}$ to each cluster $Ci$.
      (b) $\forall_{\text{clusters}}$ $Ci$ in $C$
          i. Calculate centroid Gi.
          ii. Calculate $KN$ nearest neighbor $KN = |DS_i|/NOC$.
      (c) $\forall_{\text{clusters}}$ $Ci$ in $C$
          i. $\forall_{\text{group}}$ $DS_i$ in $DS_{rem}$
              a Determine the $KN$ nearest neighbors based on centroid $C_i$ from group $DS_i$.
              b $C_i \leftarrow KN$ number of nearest neighbors.
              c Eliminate instances to be added from $DS_i$.
3. else
  (a) $AddlInst \leftarrow (k/S)$-card{$DS_{min}$}.
  (b) $DS_{rem}$ $\leftarrow$add the $AddlInst$ duplicate instances.
  (c) Repeat the steps in 2.
```

Once the nearest neighbor clusters are formed, anonymization is performed. Each of the resultant clusters formed is treated as one equivalence class and is assigned to the mapper in Hadoop MapReduce program for parallelizing anonymization of clusters. During the anonymization step, *QI* values are substituted with the centroid of the corresponding clusters. Later, the reducer will combine all of the mapper output and produce a privacy preserved data set (*DSp*).

Results and Discussion

In order to show the efficiency of the proposed algorithm in terms of degree of privacy, data usefulness, scalability and execution time, experiments are performed to test metrics such as Kullback–Leibler divergence, F-measure, classification accuracy and discernibility cost.

Experimental Setup

All the experiments for testing the privacy and scalability of the big data set were performed in an Intel(R) i7-4790 computer with a 4-core CPU (3.60 GHz), 16 GB RAM, and 1 TB hard disk space. It operates on Ubuntu 14.04 platform where jdk1.8 and Hadoop-2.8.0 with HDFS, YARN and MapRedue frameworks are installed. The HDFS block size is 128MB. The proposed algorithm is executed on the Spyder IDE by using the Python programming language. The free integrated development environment (IDE) called Spyder, Scientific Python Development Environment, is pre-installed in Anaconda Navigator, which is included in Anaconda. The Weka 3.7.12 and Anonymization toolbox (UT Dallas Data Security and Privacy Lab 2010) are also installed for evaluating the performance of the proposed algorithm with various classifiers on different privacy preserved data sets.

Data Sets

Five data sets with varying sizes were considered for evaluating the performance of the proposed parallel clustering based anonymization algorithm. Those data sets can be collected from the UCI machine learning repository (Dua and Graff 2019) and the OpenML data repository(Joaquin 2013). The description of the data set is given in Table 5:

Adult Data Set

To compare the proposed parallel big data clustering algorithm with the existing algorithms, the benchmark adult data set in the UCI machine learning repository (Lichman 2013) is used. The adult data set contains a total of 32,561 records and has 30,162 records without missing values. There are totally 15 attributes, including six numeric and nine categorical attributes. There are 7508 instances for class ">50 K" and 22,654 instances for class "≤50 K". It is described in Table 6.

Synthetic Data Set

The scalability of the proposed algorithm on big data is ascertained by conducting experiments on eight synthetic data sets having the sizes of 10000, 30000, 50000, 60000, 120000, 240000, 480000, and 980000. The synthetic data

Table 5. Description of data sets.

Data set Name	Number of Records	Number of Attributes
Heart Disease Data(David 1988)	303	75
Kasandr data(Sumit 2017)	10000	21
Statlog(Jason 1995)	58000	9
HEPMASS(Daniel 2016)	300000	28
Adult data set(Ronny and Barry 2019)	30162	14

Table 6. Description of the adult data set.

Attribute	Attribute Type	Domain Description
Age	Numeric	[17–90]
Workclass	Categorical	State-gov, self-emp-not-inc, private, Federal-gov, local-gov, self-emp-inc, without-pay, never-worked
Fnwgt	Numeric	[19214–1226583]
Education	Categorical	Assoc-acdm, assoc-voc, doctorate, masters, bachelors, some-college, HS-grad, prof-school, pre-school, 1st-4th, 5th-6th, 7th-8th, 9th, 10th, 11th, 12th
Education_num	Numeric	[1–16]
Marital_status	Categorical	Never-married, married-civ-spouse, divorced, married-spouse-absent, separated, married-AF-spouse, widowed
Occupation	Categorical	Adm-clerical, exec-managerial, handlers-cleaners, prof-specialty, other-service, sales, transport-moving, farming-fishing, machine-op-inspct, tech-support, craft-repair, protective-serv, armed-Forces, priv-house-serv
Relationship	Categorical	Wife, not-in-family, husband, own-child, other-relative, unmarried
Race	Categorical	Black, Amer-Indian-Eskimo, White, Black, Asian-Pac-Islander, Other
Sex	Categorical	Male, female
Capital_gain	Numeric	[0–99999]
Capital_loss	Numeric	[0–4356]
Hours_per_week	Numeric	[1–99]
Native_country	Categorical	Cambodia, Canada, China, Columbia, Cuba, Dominican-Republic, Ecuador, El-Salvador, England, France, Germany, Greece, Guatemala, Haiti, Holand-Netherlands, Honduras, Hong, Hungary, India, Iran, Ireland, Italy, Jamaica, Japan, Laos, Mexico, Nicaragua, Outlying-US (Guam-USVI, etc.), Peru, Philippines, Poland, Portugal, Puerto-Rico, Scotland, South, Taiwan, Thailand, Trinadad & Tobago, United-States, Vietnam, Yugoslavia
Income (class attribute)	Categorical	>50 K, ≤50 K

set can be generated using the *sklearn.dataset*(Scikit Learn Tutorial 2006) module present in *Scikit-learn library* of Python 3.9. The number of attributes in the synthetic data set is similar to original adult data set with equal distribution of class attributes. Similar to the adult data set, the synthetic data set has 15 attributes including six numeric {*Age, Fnwgt, Education_num, Capital_gain, Capital_loss, Hours_per_week*} and nine categorical {*Workclass, Education, Marital_status, Occupation, Relationship, Race, Sex, Native_country, Income}* attributes, among which one of the attributes '*income*' is the class attribute having two distinct values. Experiments were also conducted by changing k values on these eight synthetic data sets. The different k values used are k = 2, 5, 10, 25, and 50. To compare the proposed parallel clustering algorithm based on different metrics such as F-measure and classification accuracy, the attributes such as '*age*', '*sex*', and '*race*' are taken as a quasi-identifiers and '*income*' and '*occupation*' are chosen as a sensitive attributes.

Evaluation Metrics

To show the efficiency of the proposed parallel big data clustering algorithm, metrics such as Kullback–Leibler divergence metric, average equivalence class size metric, discernibility metric, classification accuracy and F-measure are used. The experiments are conducted based on the two cases of the adult data set, as shown in Table 7.

Metrics on Data Utility and Privacy

A high degree of anonymization would be worthwhile to achieve data privacy. On the other hand, the utility of the data may also be affected, meaning that fewer values can be extracted from the data. In big data applications, it is important to balance the trade-off between privacy and utility. Information loss is reduction in data utility: Higher loss of information suggests less utility of anonymized information. To achieve high utility of anonymized data, we must reduce the loss of information as much as possible. The metrics such as Average Equivalence Class Size (C_{Avg}) (Li et al. 2007; Nayahi and Kavitha 2015; Xiaoxun, Min, and Hua 2011), Discernibility Metric (DM) cost (Bayardo and Agrawal 2005; Li et al. 2007; Nayahi and Kavitha 2015; Wong, Li, and Fu et al. 2009), and Kullback–Leibler(KL) divergence metric (Machanavajjhala et al. 2006; Nayahi and Kavitha 2015; Xiaoxun, Min, and Hua 2011) are used to measure the usefulness of anonymized information. Lower values of these metric indicates less information loss and leads to higher utility of data.

Average Equivalence Class Size (C_{avg})

A metric to denote the loss of utility is the average equivalence class size, C_{Avg} (Nayahi and Kavitha 2015). The C_{Avg} value would be low if the number of equivalence classes created was high. A lower value for C_{Avg} is preferred to denote the less information loss. Figure 3 displays the average size of the equivalence class of the proposed parallel clustering based anonymization algorithm based on various k values. As seen from the figure, the C_{Avg} for the proposed parallel clustering algorithm, particularly in Case I, is very small and it is comparatively higher for lower k values in Case II and decreases gradually as the k value increases.

Table 7. Case I & case II of the adult data set.

	SensAttr, S		QI	
Case	Attribute	Unique Values	Attribute	Unique Values
I	Income	2	Race	5
II	Occupation	14	Age	74
			Sex	2

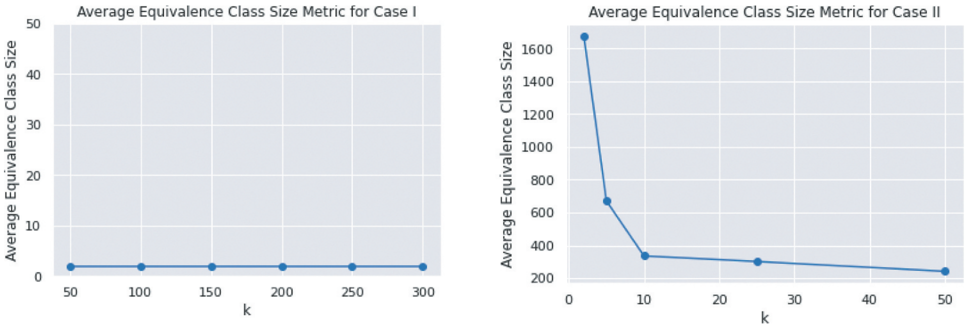


Figure 3. Average size of equivalence class for cases I & II.

The comparison of the average equivalence class size of the proposed parallel big data clustering algorithm using an original adult data set based on various privacy-preserving algorithms such as Datafly (Sweeney 1998), Mondrian (LeFevre, DeWitt, and Ramakrishnan 2006), (G,S) (Nayahi and Kavitha 2015), KNN-(G,S) (Eyupoglu et al. 2018; Nayahi and Kavitha 2017), and Incognito (LeFevre, DeWitt, and Ramakrishnan 2006) is shown in Figure 4. The average equivalence class size of the proposed algorithm is low when compared to the existing algorithms, and it leads to minimum information loss.

Discernibility Metric (DM)

The loss of utility is also measured by the Discernibility Metric (DM) cost (Nayahi and Kavitha 2015), which is a measure of equivalence class size. A lower DM cost value is favored, as lower values result in low utility loss and refer to the equivalence class of small size. We need to minimize the amount of tuples that are indistinguishable in an equivalence class to satisfy the k-anonymity criterion. Figure 5 displays the DM cost of the proposed

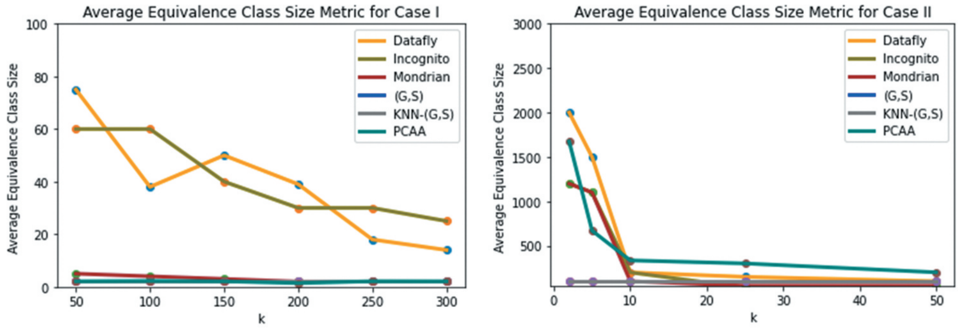


Figure 4. Comparison of Average Equivalence Class Size (*CAvg*) on other privacy-preserving algorithms.

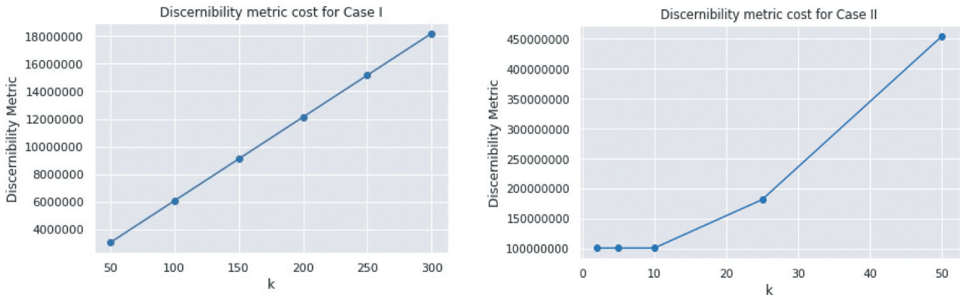


Figure 5. DM cost for cases I & II.

parallel clustering based anonymization algorithm based on Cases I and II. For Case I, it displays the least discernibility metric value than in Case II of the experiments.

The comparison of Discernibility Metric of proposed parallel big data clustering algorithm using original adult data set based on various privacy-preserving algorithms such as Datafly (Sweeney 1998), Mondrian(LeFevre, DeWitt, and Ramakrishnan 2006), (G,S) (Nayahi and Kavitha 2015), KNN-(G,S) (Eyupoglu et al. 2018; Nayahi and Kavitha 2017), and Incognito (LeFevre, DeWitt, and Ramakrishnan 2006) is shown in Figure 6. As seen from the figure, the proposed algorithm gives better results in terms of DM cost than existing algorithms and leads to high utility of information.

Kullback Leibler Divergence Metric (KL)

To compute the variance between the distribution before and after the anonymization process, the Kullback Leibler Divergence Metric (KL) (Nayahi and Kavitha 2015) is used. In the distribution, lower values denote the lower distortions. If both distortions are similar, the value of KL is zero. Figure 7 shows the KL divergence of the proposed parallel big data clustering algorithm based on different values of *k*. It shows that divergence is very low in both cases of the experiments.

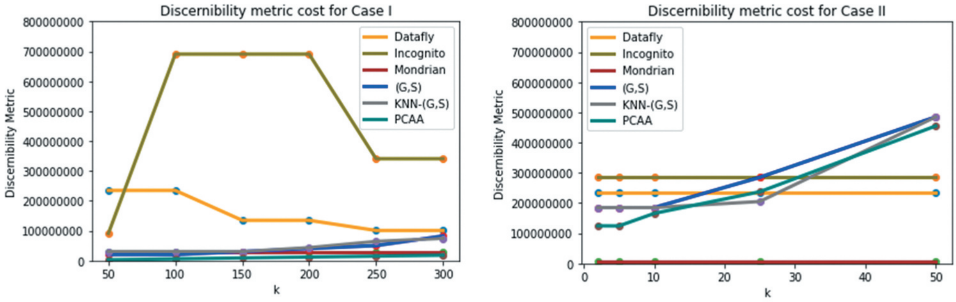


Figure 6. Comparison of discernibility metric on other privacy-preserving algorithms.

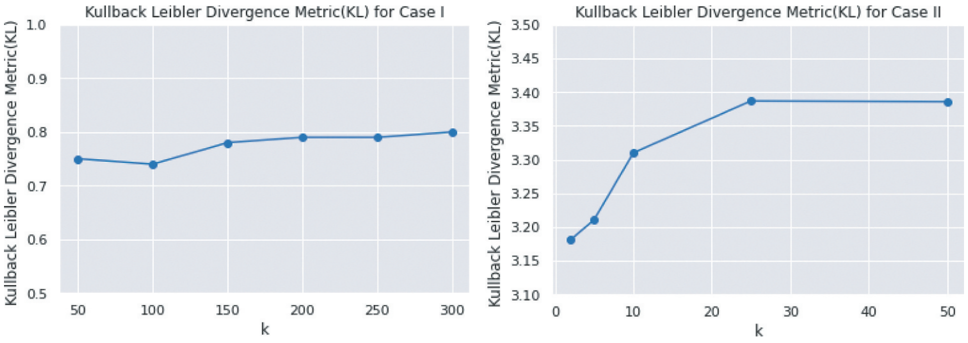


Figure 7. Kullback Leibler divergence Metric for case I and case II.

The comparison of KL divergence of proposed parallel big data clustering algorithm using original adult data set based on various privacy-preserving algorithms such as Datafly (Sweeney 1998), Mondrian(LeFevre, DeWitt, and Ramakrishnan 2006), (G,S) (Nayahi and Kavitha 2015), KNN-(G,S) (Eyupoglu et al. 2018; Nayahi and Kavitha 2017), and Incognito (LeFevre, DeWitt, and Ramakrishnan 2006) is shown in Figure 8. The KL divergence of proposed algorithm is comparatively low in Case I of the experiments than in Case II. The value of KL divergence is better when compared to other existing approaches.

Cluster Output

Another important metric for illustrating data loss and privacy is the number of instances in each equivalence class. Using the adult data set consisting of 30162 records, 15 attributes and 14 sensitive values, Figure 9 shows the number of clusters formed and the size of each cluster for various *k* values such as 50, 100, 150, 200, 250, and 300. The results show that, the number of clusters formed decreases with the increase in *k* values. This decrease in the

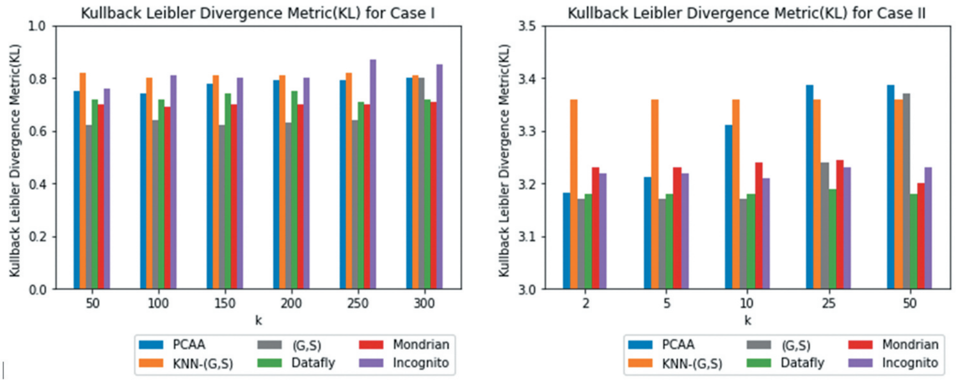


Figure 8. Comparison of KL divergence on other privacy-preserving algorithms.

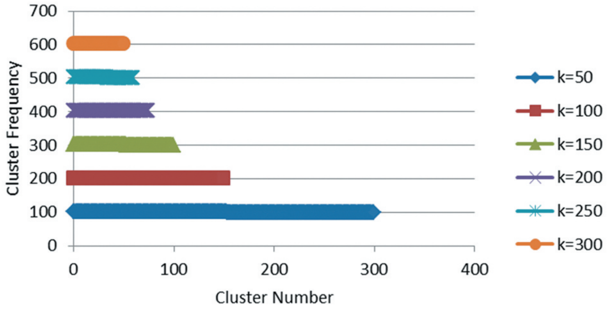


Figure 9. Cluster output of the parallel clustering based anonymization algorithm for a range of k values.

number of clusters indicates that decrease in data utility and therefore it increases the data privacy. It implies that higher loss of information and the data protection obtained is high.

Classification Accuracy

The classification accuracy (Eyupoglu et al. 2018) is another important measure to evaluate the accuracy of different classifiers. It is defined as the rate of predictions that our model got right. A higher value of classification accuracy denotes lower information loss so as to it increases the data utility. To compare the classification accuracy of proposed parallel clustering based anonymization algorithm, the classifiers such as Decision Tree(J48), Naive Bayes(NB), OneR and Voted Perceptron(VP) were chosen. The classification accuracy of parallel clustering based anonymization algorithm on five different data sets using 2-fold, 5-fold and 10-fold cross validation before and after doing privacy preservation is given in Table 8. A slight improvement in the '*k*' value increases the accuracy of the classification substantially. Both the privacy preserved and the original data sets are very close in classification accuracy.

F-Measure

Another measure to evaluate the test accuracy of different classifiers is F-measure (Eyupoglu et al. 2018). It is calculated based on the measure of accuracy and completeness calculation called precision and recall. To compare the F-measure of proposed parallel big data clustering algorithm, the classifiers such as OneR, Voted Perceptron(VP), Decision Tree(J48), and Naive Bayes(NB) were chosen. The F-measure of parallel clustering based anonymization algorithm on five different data sets using 2-fold, 5-fold and 10-fold cross validation before and after doing privacy preservation is given in Table 9. Higher F-measure values which are closer to originals are preferred.

Table 8. Analysis of classification accuracy of the parallel big data clustering algorithm on different data sets.

	k-Fold Cross Validation		2-Fold				5-Fold				10-Fold			
Sl.No	Data Sets		J48	Voted Percept	One R	Naive Bayes	J48	Voted Percept	One R	Naive Bayes	J48	Voted Percept	One R	Naive Bayes
1.	Input Data set	Before	88.73	78.44	80.22	81.88	88.67	78.84	80.06	80.01	88.74	79.72	80.42	81.15
		After	90.22	78.43	80.25	80.23	90.09	78.81	80.15	80.36	90.14	79.76	80.26	80.95
2.	60000	Before	89.75	79.42	80.25	81.15	89.96	78.47	79.56	81.92	88.65	78.42	78.83	81.75
		After	90.07	79.42	80.12	81.78	90.14	78.43	79.56	80.25	90.23	78.77	78.23	81.03
3.	120000	Before	90.27	79.42	78.35	80.36	91.24	78.45	80.15	81.86	92.89	78.45	80.15	81.25
		After	90.56	79.43	78.72	80.78	91.14	78.43	79.63	81.25	92.23	78.77	80.23	81.03
4.	240000	Before	91.56	78.47	79.56	81.63	92.02	78.89	80.25	82.90	92.15	78.74	82.54	82.90
		After	92.36	78.54	79.42	81.41	93.02	78.17	81.12	82.65	93.45	78.94	87.74	82.63
5.	480000	Before	93.56	78.40	88.69	82.90	94.52	78.42	89.30	82.90	94.25	78.44	88.73	82.90
		After	92.16	78.52	87.56	82.41	93.45	77.52	88.56	83.75	93.25	78.58	87.56	82.36

Table 9. Analysis of F-Measure of parallel big data clustering algorithm on different data sets.

	k-Fold Cross Validation		2-Fold				5-Fold				10-Fold			
Sl.No	Data Sets		Voted Percept	One R	Naïve Bayes	J48	Voted Percept	One R	Naïve Bayes	J48	Voted Percept	One R	Naïve Bayes	J48
1.	Input	Before	0.708	0.748	0.808	0.849	0.720	0.749	0.812	0.854	0.723	0.751	0.818	0.851
	Data set	After	0.708	0.742	0.813	0.850	0.712	0.747	0.817	0.845	0.723	0.751	0.818	0.849
2.	60000	Before	0.722	0.749	0.817	0.858	0.723	0.732	0.818	0.852	0.722	0.748	0.818	0.853
		After	0.722	0.752	0.818	0.849	0.721	0.731	0.817	0.849	0.722	0.752	0.817	0.862
3.	120000	Before	0.723	0.749	0.829	0.853	0.726	0.758	0.820	0.853	0.727	0.762	0.817	0.877
		After	0.722	0.756	0.818	0.854	0.723	0.762	0.815	0.856	0.722	0.758	0.818	0.884
4.	240000	Before	0.723	0.826	0.818	0.863	0.721	0.767	0.818	0.865	0.722	0.749	0.816	0.885
		After	0.723	0.827	0.818	0.867	0.722	0.767	0.819	0.868	0.722	0.745	0.815	0.885
5.	480000	Before	0.723	0.876	0.819	0.854	0.722	0.789	0.818	0.862	0.724	0.747	0.818	0.892
		After	0.723	0.856	0.827	0.865	0.720	0.784	0.819	0.856	0.721	0.748	0.819	0.895

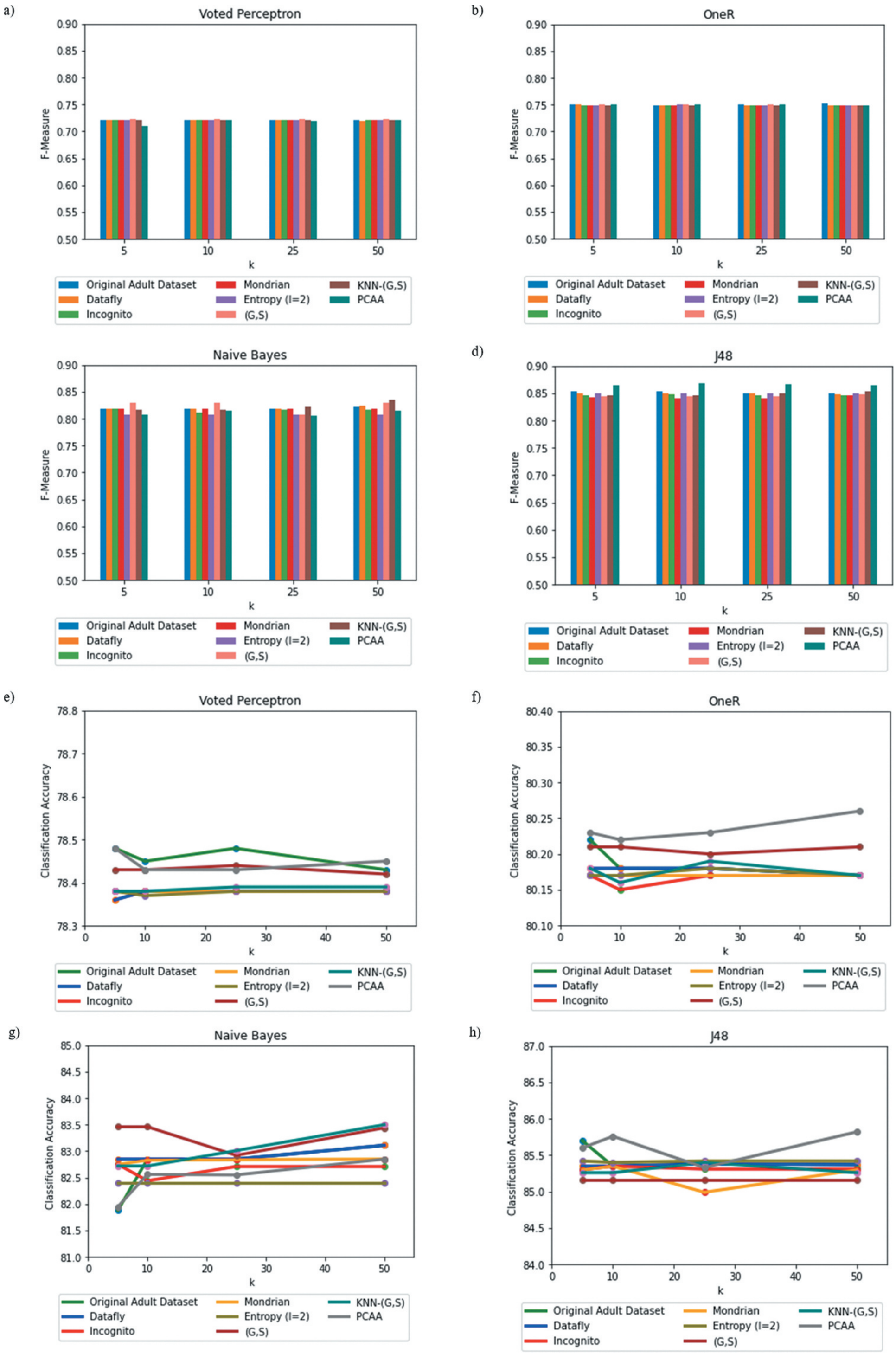


Figure 10. (a)–(h) F-measure and classification accuracy comparison of J48, Naive Bayes, Voted perceptron, and OneR classifiers.

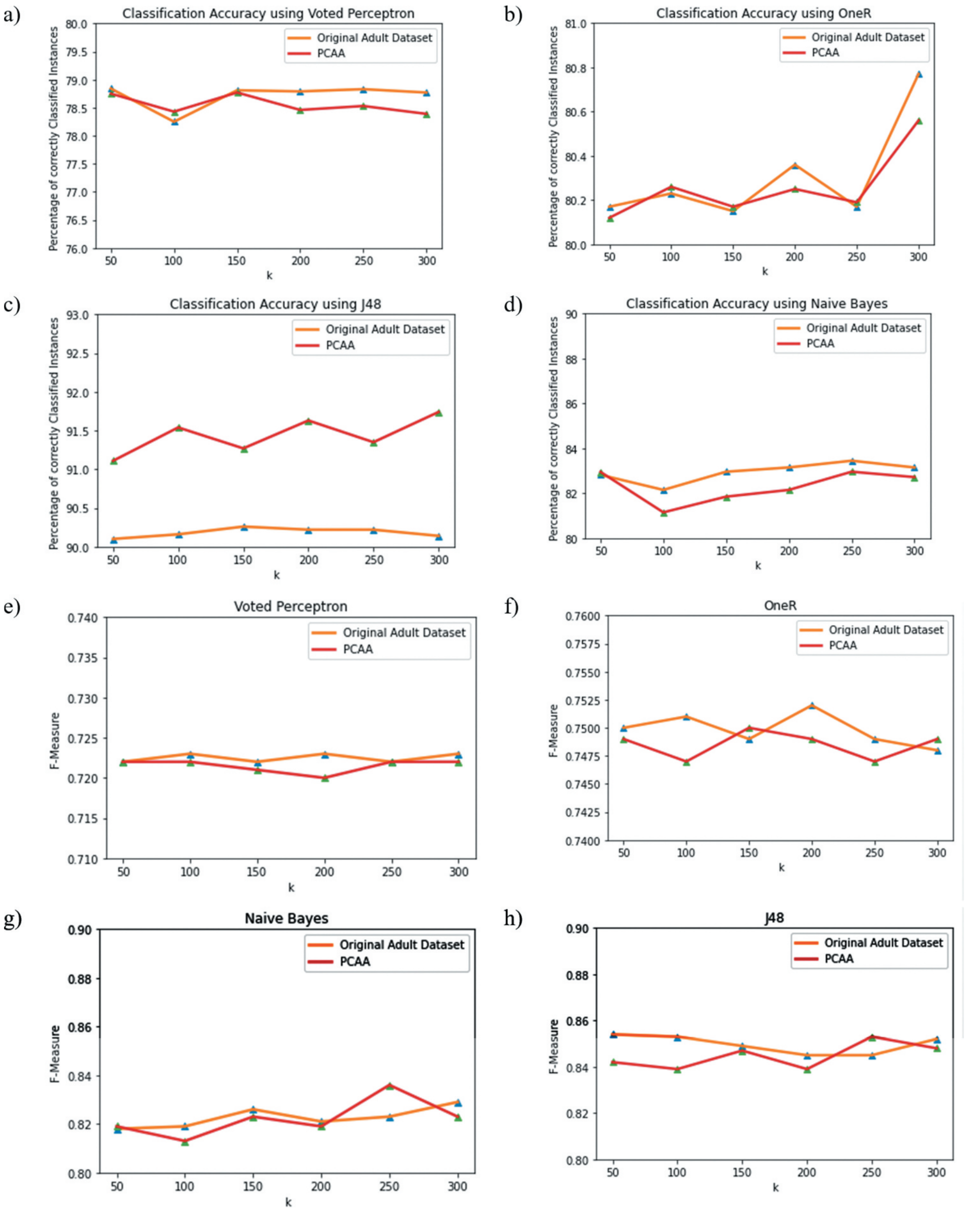
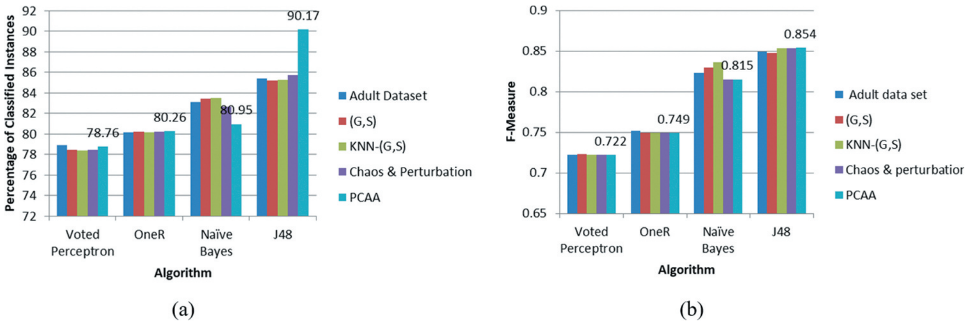


Figure 11. (a)–(h) Percentage of the correctly classified instances and F-measure of the four classifiers: J48, naive Bayes, voted perceptron, and OneR.

Figure 10(a)-(d) as well as Figure 10(e)-(h) shows the F-measure and classification accuracy comparison of four classifiers, Voted Perceptron, OneR, J48, and Naive Bayes using original adult data set by 10-fold cross-validation scheme based on various privacy-preserving algorithms such as Datafly (Sweeney 1998), Entropy l-diversity(Wong, Li, and Fu et al. 2009),



(a)

(b)

Figure 12. Comparison of the proposed algorithm: (a) percentage of correctly classified instances and (b) F-measure.

(G,S) (Nayahi and Kavitha 2015), Mondrian(LeFevre, DeWitt, and Ramakrishnan 2006), KNN-(G,S) (Eyupoglu et al. 2018; Nayahi and Kavitha 2017), and Incognito (LeFevre, DeWitt, and Ramakrishnan 2006).

When utilizing J48 and OneR classifiers, the parallel clustering-based anonymization algorithm outperforms (G,S) and KNN-(G,S) in terms of F-Measure and percentage of correctly identified instances. It also gives better results in terms of F-Measure and percentage of correctly classified instances when utilizing the J48 classifier.

Classification Result Analysis

Different classifiers, namely Voted Perceptron(VP), OneR, Naive Bayes(NB), and Decision Tree(J48) are executed to compare the percentage of correctly classified instances and F-Measure of the proposed parallel big data clustering algorithm on the privacy-preserved data set and for the original adult data set. The comparison of these classifiers based on Case I is illustrated in Figure 11 (a)-(d) and Figure 11(e)-(h). The proposed parallel clustering based anonymization algorithm shows better results when classified using J48 and Naive Bayes classifiers regarding the percentage of correctly classified instances and F-Measure.

Figure 12(a) and 12(b) show the percentage of correctly categorized instances and the F-measure of the four classifiers on the privacy-preserved data sets, namely Voted Perceptron, Naive Bayes, OneR and J48 and the original adult data set in Case II with a k value of 50. As seen from the figure, the proposed parallel clustering based anonymization algorithm performs better in terms of percentage of correctly classified instances and F-Measure, when utilizing the classifiers OneR and J48.

Table 10. Degree of privacy.

Experimental Case	*K*	Parallel Clustering based Anonymization
I	2	4
	5	4
	10	4
	15	4
	25	4
II	50	1675
	100	3351
	150	3351
	200	3348
	250	3348
	300	3348

Degree of Privacy

The degree of privacy P (David et al. 2010; Ghinita, Kalnis, and Tao 2011; Nayahi and Kavitha 2015) should be at least S, which is specified as the cardinality of the sensitive attribute domain, *SensAttr*. The privacy degree (P) achieved by the algorithm is at least S (i.e. $P \geq S$). Table 10 shows the privacy degree of the proposed parallel big data clustering algorithm in the two experimental cases performed on the Adult data set.

1/P is an attacker's confidence level for linking the QI attribute of a person with its corresponding sensitive attribute value. The algorithm proposed in Case I of the experiments show only a 0.25% or 25% chance of connecting a record to the sensitive value of the attribute. In addition, the clustered data set in Case II of the experiments decreases the probability of connecting the data set even more. For a given data set, increase in number of clusters decreases the number of instances in each cluster. This results in a lower loss of information in order to achieve a low degree of privacy.

Execution Time

Using the synthetic data set of size 10000, 30000, 50000, 60000, 120000, 240000, 480000, and 960000, the execution time and scalability of the proposed algorithm is estimated. Figure 13(a) and (b) displays the execution time of the proposed Parallel Clustering based Anonymization Algorithm(PCAA) based on the data set size and number of clusters formed. With the increase in the number of clusters, the execution time of the algorithm increases. This is due to the increase in the number of iterations as the data set size increases. In terms of execution time, the proposed parallel big data clustering algorithm performs well compared to the existing (G,S) (Nayahi and Kavitha 2015) and KNN-(G, S) (Nayahi and Kavitha 2017), as shown in Figure 13(c)–(d). It also demonstrates that in terms of scalability and feasibility for handling large data sets, the proposed algorithm is optimal enough.

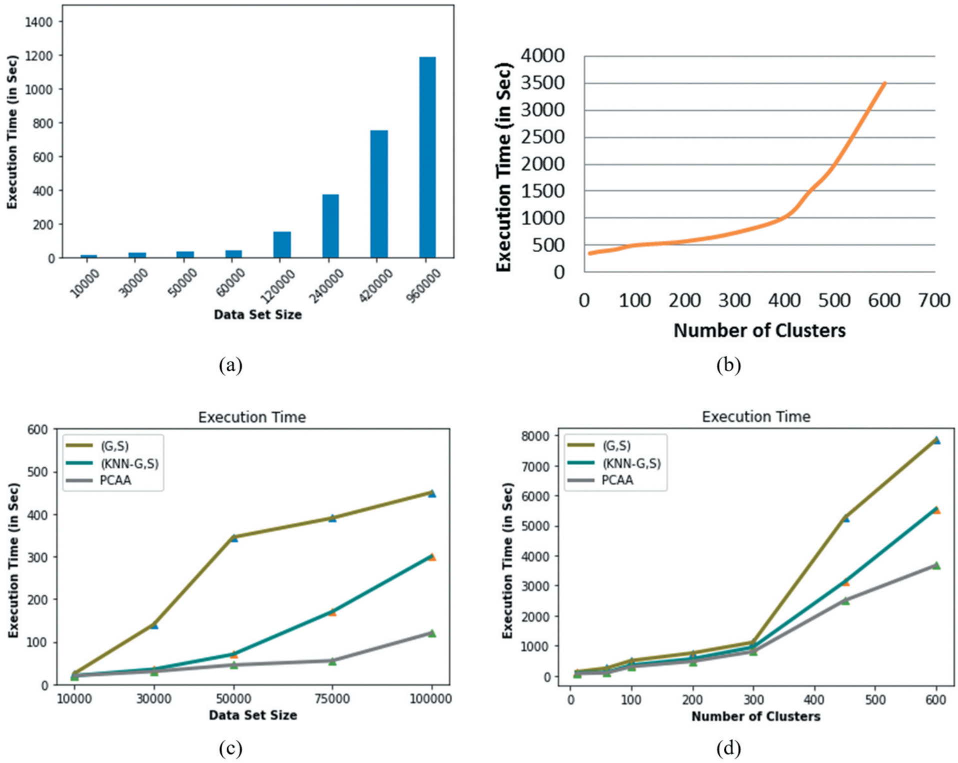


Figure 13. Execution time vs (a) data set size and (b) number of clusters. (c) and (d) Comparison with (G,S) and KNN-(G,S).

Conclusion

In this paper, a Parallel Clustering based Anonymization Algorithm(PCAA) has been introduced to ensure the preservation of privacy and utility in big data. Hadoop MapReduce framework is used to parallelize the anonymization process for handling huge volume of data. Using the proposed big data clustering algorithms, sensitive information in big data can be protected against various attacks, such as linking attack, homogeneity attack, similarity attack and probabilistic inference attack. Based on several data sets of different sizes, the execution time efficiency and scalability of the proposed algorithm was investigated. A Parallel Clustering based Anonymization Algorithm(PCAA) performs well in terms of F-measure, classification accuracy and Kullback–Leibler divergence metrics. The experimental results show that the proposed parallel clustering based anonymization algorithm performs better in terms of execution time when compared to the existing (G,S) and KNN-(G,S) approaches. This can be further improved by parallelizing the complete clustering algorithm that produces better results in terms of scalability and speed as a future work. The proposed algorithm ensures its suitability that the big data generated from heterogeneous data sources are efficiently protected to satisfy the ever-growing requirements of the application and ensure the privacy of the individual before publishing and sharing data.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

Aggarwal G. et al. 2005. Anonymizing Tables. In: Eiter T., Libkin L. (Eds) Database Theory - ICDT 2005, pages 246–258.

Aggarwal, C. C., and P. S. Yu. 2008. *Privacy-preserving data mining: Models and algorithms.* Berlin: Springer Publication. doi:10.1007/978-0-387-70992-5.

Al-Zobbi, M., S. Shahrestani, and C. Ruan. 2017. Improving MapReduce privacy by implementing multi-dimensional sensitivity-based anonymization. *Journal of Big Data* 4 (1):45. doi:10.1186/s40537-017-0104-5.

Amit, K., and S. Neeraj. 2016. Privacy preservation in big data using k-anonymity algorithm with privacy key. *International Journal Of Computer Applications* 153 (5):0975–8887.

Bayardo, R., and R. Agrawal. 2005. Data privacy through Optimal k-anonymization, in: Proceedings of 21st International Conference on Data Engineering (ICDE), Tokyo, 5-8 April 2005, pp.217–28. 10.1109/ICDE.2005.42.

Chamikara, M. A. P., P. Bertok, D. Liu, S. Camtepe, and I. Khalil. 2019. Efficient privacy preservation of big data for accurate data mining, Information Sciences. doi:10.1016/j.ins.2019.05.053.

Chamikara, M. A. P., P. Bertok, D. Liu, S. Camtepe, and I. Khalil. 2020. Efficient privacy preservation of big data for accurate data mining. *Information Sciences* 527:420–43. 0020-0255doi:10.1016/j.ins.2019.05.053.

Chen, K., G. Sun, and L. Liu. 2007. Towards attack-resilient geometric data perturbation, in: Proceedings of the Seventh SIAM International Conference on Data Mining, April 26-28, 2007, Minneapolis, Minnesota, USA, pp. 78–89. doi: 10.1137/1.9781611972771.8.

Chen, K., and L. Liu. 2009. Privacy-preserving multiparty collaborative mining with geometric data perturbation. *IEEE Transactions on Parallel and Distributed Systems* 20 (12):1764–76. doi:10.1109/TPDS.2009.26.

Chen, K., and L. Liu. 2011. Geometric data perturbation for privacy preserving outsourceddata mining, Springer-Knowl. *Inf. Syst* 29:657–95. doi:10.1007/s10115-010-0362-4.

Chi-Wing Wong, R., J. Li, A. Wai-Chee Fu, and K. Wang. (2006). (α, K)-Anonymity: An enhanced k-anonymity model for privacy preserving data publishing. ACM Digital Library, Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, USA, p.754–59

Daniel Whiteson daniel '@' uci.edu. 2016. HEPMASS Data Set. https://archive.ics.uci.edu/ml/datasets/HEPMASS#

David, R. M., J. Forne, and J. Domingo-Ferrer. 2010. From t-closeness-like privacy to post randomization via information theory. *IEEE Transactions on Knowledge and Data Engineering* 22 (11):1623–36. doi:10.1109/TKDE.2009.190.

David W. Aha.1988. Heart Disease Data Set, (714): 856-8779. http://archive.cs.uci.edu/ml/datasets/Heart+Disease

Dean, J., and S. Ghemawat. 2004. MapReduce: Simplied data processing on large clusters. OSDI.

Dua, D. and Graff, C. 2019. UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. http://archive.ics.uci.edu/ml

Eyupoglu, C., M. A. Aydin, A. H. Zaim, and A. Sertbas. An efficient big data anonymization algorithm based on chaos and perturbation techniques. *Entropy (Basel)* 2018, May 17. 20 (5):373. doi:10.3390/e20050373. PMID: 33265463; PMCID: PMC7512893.

Fahad, A., Z. Tari, A. Almalawi, A. Goscinski, I. Khalil, and A. Mahmood. 2014. PPFSCADA: Privacy preserving framework for SCADA data publishing. *Future Generation Computer Systems* 37:496–511. doi:10.1016/j.future.2014.03.002.

Fung, B. C. M., K. Wang, R. Chen, and P. S. Yu. 2010. Privacy preserving data publishing: A survey on recent developments. *ACM Computing Surveys* 42 (4):14: 1–14:53. doi:10.1145/1749603.1749605.

Ghinita, G., P. Kalnis, and Y. Tao. 2011. Anonymous publication of sensitive transactional data. *IEEE Transactions on Knowledge and Data Engineering* 23 (2011):161–74. doi:10.1109/TKDE.2010.101.

Girka, A., V. Terziyan, M. Gavriushenko, and A. Gontarenko. 2021. Anonymization as homeomorphic data space transformation for privacy-preserving deep learning. *Procedia Computer Science* 180:867–76. 1877-0509. doi:10.1016/j.procs.2021.01.337.

Govinda, K., and E. Sathiyamoorthy. 2012. Identity anonymization and secure data storage using group signature in private cloud. *Procedia Technology* 4:495–99. doi:10.1016/j.protcy.2012.05.079.

Goyal, V., O. Pandey, A. Sahai, and B. Waters. 2006. Attribute-based encryption for fine-grained access control of encrypted data. Proceedings of the 13th ACM Conference on Computer and Communications Security - CCS '06. Alexandria, VA, USA. doi:10.1145/1180405.1180418

Gu, R., X. Yang, J. Yan, Y. Sun, B. Wang, C. Yuan, and Y. Huang. 2014. Hadoop: Improving MapReduce performance by optimizing job execution mechanism in hadoop clusters. *J. Parallel Distrib. Comput* 74 (3):2166–79. doi:10.1016/j.jpdc.2013.10.003.

Hadoop, (2009). http://hadoop.apache.org

Han, J., and M. Kamber. 2006. *Data mining concepts and techniques*. Morgan Kaufmann Publishers. Imprint of Elsevier. 225 Wyman Street, Waltham, MA 02451, USA.

Hayward, R., and C. Chiang. 2015. Parallelizing fully homomorphic encryption for a cloud environment. *Journal of Applied Research and Technology* 13 (2):245–52. doi:10.1016/j.jart.2015.06.004.

HIPAA. (1999). Health insurance portability and accountability act of 1999. http://www.hhs.gov/ocr/privacy/hipaa/administrative/privacyrule (accessed 20.06.15).

Hongwei, T., and Z. Weining. 2011. Extending l-diversity to generalize sensitive data. *Elsevier Journal of Data and Knowledge Engineering* 70 (1):101–26. doi:10.1016/j.datak.2010.09.001.

Islam, M. Z., and L. Brankovic. 2011. Privacy preserving data mining: A noise addition framework using a novel clustering technique. *Knowl.-Based Syst* 24 (8):1214–23. doi:10.1016/j.knosys.2011.05.011.

Jain, P., M. Gyanchandani, and N. Khare. 2016. Big data privacy: A technological perspective and review. *J Big Data* 3 (1):25. doi:10.1186/s40537-016-0059-y.

Jason Catlett. 1995. Statlog (Shuttle) Data Set. https://archive.ics.uci.edu/ml/datasets/Statlog+%28Shuttle%29

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. 2013. OpenML: networked science in machine learning. *SIGKDD Explorations* 15(2): 49–60.

Khan, S., K. Iqbal, S. Faizullah, M. Fahad, J. Ali, and W. Ahmed. 2019. Clustering based privacy preserving of big data using fuzzification and anonymization operation. *International Journal of Advanced Computer Science and Applications* 10 (12). doi: 10.14569/IJACSA.2019.0101239.

Lammel, R. 2008. Google's MapReduce programming model-revisited. *Sci Comput Progr* 70 (1):1–30. doi:10.1016/j.scico.2007.07.001.

Lauter, K., M. Naehrig, and V. Vaikuntanathan. 2011. *Can homomorphic encryption be practical?*, 113–24. Chicago, IL: The 3rd ACM Workshop on Cloud Computing Security.

LeFevre, K., D. J. DeWitt, and R. Ramakrishnan. 2005. Incognito: Efficient full domain k-anonymity. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Baltimore, MD, USA, 14–16 June 2005; pp.49–60.

LeFevre, K., D. J. DeWitt, and R. Ramakrishnan. 2006. Mondrian multidimensional k-anonymity. In Proceedings of the 22nd International Conference on Data Engineering (ICDE'06), Atlanta, GA, USA, IEEE.

N. Li, T. Li and S. Venkatasubramanian. 2007.t-Closeness: Privacy beyond k-anonymity and l-diversity. IEEE 23rd International Conference on Data Engineering (2007), Istanbul, Turkey, 106–15. 10.1109/ICDE.2007.367856.

Li, N., W. Qardaji, and D. Su (2012). On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy. In Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, Seoul, Korea, 2–4; pp. 32–33.

Lichman, M., (2013). UCI Machine Learning Repository, http://archive.ics.uci.edu/ml (accessed 20.06.15).

Lindell, Y., and B. Pinkas. 2009. Secure multiparty computation for privacy-preserving data mining. *J. Priv. Confidentiality* 1:59–98.

Liu, H., X. Huang, and J. K. Liu. 2015. Secure sharing of personal health records in cloud computing: Ciphertext-policy attribute-based signcryption. *Future Gener.Comput. Syst* 52:67–76. doi:10.1016/j.future.2014.10.014.

Machanavajjhala, A., D. Kifer, J. Gehrke, and M. Venkitasubramaniam. 2007. L-diversity. *ACM Transactions on Knowledge Discovery from Data* 1 (1):1. doi:10.1145/1217299.1217302.

Machanavajjhala, A., J. Gehrke, D. Kifer, and M. Venkitasubramaniam. 2006. L-diversity: Privacy beyond k-anonymity. 22nd International Conference on Data Engineering (ICDE'06), Atlanta, GA, USA. doi:10.1109/icde.2006.1;

Mehmood, A., I. Natgunanathan, Y. Xiang, G. Hua, and S. Guo. 2016. Protection of Big Data Privacy. *IEEE Access* 4. 1-1. doi:10.1109/ACCESS.2016.2558446.

Meyerson, A., and R. Williams. 2004. On the complexity of optimal K-Anonymity. In: Proc. of the ACM Symp. on Principles of Database Systems. Paris France. June 14 - 16, 2004.

Nayahi, J. J. V., and V. Kavitha. 2015. An efficient clustering for anoymizing data and protecting sensitive labels. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst* 23 (5):685–714. doi:10.1142/S0218488515500300.

Nayahi, J. J. V., and V. Kavitha. 2017. Privacy and utility preserving data clustering for data anonymization and distribution on Hadoop. *Future Gener. Comput. Syst* 74:393–408. doi:10.1016/j.future.2016.10.022.

Ogburn, M., C. Turner, and P. Dahal. 2013. Homomorphic encryption. *Procedia Computer Science* 20:502–09. doi:10.1016/j.procs.2013.09.310.

Orsini, M., M. Pacchioni, A. Malagoli, and G. Guaraldi. 2017. My smart age with HIV: An innovative mobile and IoMT framework for patient's empowerment, in: Proc. IEEE International Forum on Research and Technologies for Society and Industry(RTSI), Modena, Italy, pp. 1–6.

Moutafis, Panagiotis & Mavrommatis, George & Vassilakopoulos, Michael, and Sioutas, Spyros. 2019. Efficient processing of all-k-nearest-neighbor queries in the MapReduce programming framework. *Data & Knowledge Engineering* 121:42–70. doi:10.1016/j.datak.2019.04.003.

Pinkas, B. 2002. Cryptographic techniques for privacy-preserving data mining. *ACM SIGKDD Explor. Newsl* 4 (2):12–19. doi:10.1145/772862.772865.

Potey, M. M., C. A. Dhote, and D. H. Sharma. 2016. Homomorphic encryption for security of cloud data. *Procedia Computer Science* 79:175–81. doi:10.1016/j.procs.2016.03.023.

Qian, J., Xia, M., and Yue, X. 2018. Parallel knowledge acquisition algorithms for big data using MapReduce. *International Journal of Machine Learning and Cybernetics*. 9(6):1007–21. doi:10.1007/s13042-016-0624-x.

Rahul, M., H. A. Alhumyani, and M. Muntjir. 2017. An improved homomorphic encryption for secure cloud data storage. *International Journal of Advanced Computer Science and Application* 8 (12):12. doi:10.14569/IJACSA.2017.081258.

Reddy, C. K., and C. C. Aggarwal. (Eds.). 2015. Healthcare data analytics (1st ed.), In *Data mining and knowledge discovery series* Chapman & Hall/CRC. https://doi.org/10.1201/b18588

Ronny, K. and Barry, B. 2019. Adult Data Set. http://archive.ics.uci.edu/ml/datasets/Adult

Saadoon, M., S. H. A. Hamid, H. Sofian, H. Altarturi, N. Nasuha, Z. H. Azizul, A. A. Sani, and A. Asemi. 2021. Experimental analysis in Hadoop MapReduce: A closer look at fault detection and recovery techniques. *Sensors* 21 (11):3799. doi:10.3390/s21113799.

Samarati, P. 2001. Protecting respondents’ identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering* 13 (6):1010–27. doi:10.1109/69.971193.

Samarati, P., and L. Sweeney. 1998. Protecting privacy when disclosing information: K-anonymity and its enforcement through generalization and suppression, in: Proceedings of IEEE Symposium on Research in Security and Privacy, Oakland, CA, USA. pp. 188–206.

Scikit Learn Tutorial. 2006. https://www.tutorialspoint.com/scikit_learn/index.htm

Sedayao, J., R. Bhardwaj, and N. Gorade. 2014. Making big data, privacy, and anonymization work together in the enterprise: Experiences and issues. *2014 IEEE International Congress on Big Data*. doi:10.1109/bigdata.congress.2014.92.

Shafer, J., S. Rixner, and A. L. Cox. 2010. The hadoop distributed file system: Balancing portability and performance, in: IEEE International Symposium on Performance Analysis of Systems & Software,White Plains, NY, USA(32), pp. 122–33. 10.1109/ISPASS.2010.5452045.

Shvachko, K., H. Kuang, S. Radia, and R. Chansler (2010). The hadoop distributed file system, in: 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies, Incline Village, NV, USA(33), pp. 1–10. doi: 10.1109/MSST.2010.5496972.

Soria-comas, J., J. Domingo-Ferrer, D. Sanchez, and S. Martinez. 2015. t-closeness through microaggregation: Strict privacy with enhanced utility preservation. *IEEE Transactions on Knowledge and Data Engineering* 27 (11):3098–110. doi:10.1109/TKDE.2015.2435777.

Soria-Comas, J., J. Domingo-Ferrer, D. Sánchez, and S. Martínez. 2014. Enhancing data utility in differential privacy via microaggregation-based k-anonymity. *VLDB J* 23 (5):771–94. doi:10.1007/s00778-014-0351-4.

Sowmya, Y., and M. Nagaratna. 2016. Parallelizing K-Anonymity algorithm for privacy preserving knowledge discovery from big data. *International Journal of Applied Engineering Research* 0973-4562 Volume 11, 2 pp 1314–21 © Research India Publications ():. http://www.ripublication.com .

Sumit, S., C. Laclau, M. Amini, G. Vandelle, and Andre. 2017. 'KASANDR: A Large-Scale Dataset with Implicit Feedback for Recommendation. https://archive.ics.uci.edu/ml/datasets/KASANDR

Sweeney, L. (1997). Guaranteeing anonymity when sharing medical data, the datafly system. Proceedings: a conference of the American Medical Informatics Association. AMIA Fall Symposium, Opryland Hotel, Nashville, TN, 51–55.

Sweeney, L. 1998. Datafly: A system for providing anonymity in medical data. In *Database security XI. IFIP advances in information and communication technology*, T. Y. Lin and S. Qian. ed., Boston, MA: Springer, pp 356-381. doi:10.1007/978-0-387-35285-5_22.

Sweeney, L. 2002a. Achieving k-anonymity privacy protection using generalization and suppression. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst* 10 (5):571–88. doi:10.1142/S021848850200165X.

Sweeney, L. 2002b. k-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst* 10 (5):557–70. doi:10.1142/S0218488502001648.

Tankard, C. 2012. Big data security. *Netw. Secur* 2012:5–8.

UT Dallas Data Security and Privacy Lab, UTD Anonymization Toolbox, 2010. http://cs.utdallas.edu/dspl/cgi-bin/toolbox/index.php (accessed 20. 06.15).

Venugopal, V., and S. Vigila. 2018. Implementing big data privacy with mapreduce for multi-dimensional sensitive data. *International Journal of Applied Engineering Research* 13 (15):11824–29.

Wong, R., J. Li, A. Fu, K. Wang. 2009. (α, k)-anonymous data publishing. *Journal of Intelligent Information Systems* 33 (2):209–34. doi:10.1007/s10844-008-0075-2.

Wong, R. C., J. Li, A. W. Fuand, and K. Wang (2006). (α, k) Anonymity: An enhanced k-anonymity model for privacy-preserving data publishing, in: Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia PA USA, pp. 733–44. doi: 10.1145/1150402.1150499

Xiaoxun, S., L. Min, and W. Hua. 2011. A family of enhanced (L,α) diversity models for privacy preserving data publishing. *Elsevier Journal of Future Generation Computer System* 27 (3):348–56. doi:10.1016/j.future.2010.07.007.

Xu, J., W. Wang, J. Pei, X. Wang, B. Shi, and A. W.-C. Fu (2006b). Utility-based anonymization using local recoding. Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’06. Philadelphia PA USA. 8(2),21–30, doi:10.1145/1150402.1150504.

Xu, J., W. Wang, J. Pei, X. Wang, B. Shi, and Fu. (2006a). Utility-based anonymization for privacy preservation with less information loss, UBDM’06, Philadelphia, Pennsylvania, USA. Copyright 2006 ACM 1-59593-440-5/06/0008.

Zhang, X., W. Dou, J. Pei, S. Nepal, C. Yang, C. Liu, and J. Chen. 2015. Proximity-aware local recoding anonymization with MapReduce for scalable big data privacy preservation in cloud. *IEEE Trans.*on Computers, 64(8): 2293-2307, 1 Aug. 2015, doi: 10.1109/TC.2014.2360516.