Data Anonymization for Privacy Preservation in Big Data

Cinzun Basheer
M.Tech in Computer Scienc and Information Systems
Dept. of CSE, Rajagiri School of Engineering and Technology
Kakkanad, India
cinzun@gmail.com

Tritpi C
Asst. Professor, Dept of CSE
Rajagiri School of Engineering and Technology
Kakkanad, India
tripti84_05@rediffmail.com

***Abstract*—**Cloud computing provides capable ascendable IT edifice to provision numerous processing of a various big data applications in sectors such as healthcare and business. Mainly electronic health records data sets and in such applications generally contain privacy-sensitive data. The most popular technique for data privacy preservation is anonymizing the data through generalization.

Proposal is to examine the issue against proximity privacy breaches for big data anonymization and try to recognize a scalable solution to this issue. Scalable clustering approach with two phase consisting of clustering algorithm and K-Anonymity scheme with Generalisation and suppression is intended to work on this problem. Design of the algorithms is done with MapReduce to increase high scalability by carrying out dataparallel execution in cloud. Wide-ranging researches on actual data sets substantiate that the method deliberately advances the competence of defensive proximity privacy breaks, the scalability and the efficiency of anonymization over existing methods.

Anonymizing data sets through generalization to gratify some of the privacy attributes like k- Anonymity is a popularly-used type of privacy preserving methods. Currently, the gauge of data in numerous cloud surges extremely in agreement with the Big Data, making it a dare for frequently used tools to actually get, manage, and process large-scale data for a particular accepted time scale. Hence, it is a trial for prevailing anonymization approaches to attain privacy conservation for big data private information due to scalabilty issues.

Keywords-*Data Anonymization, Privacy Preservation, Big Data*

*****

I. INTRODUCTION

Currently worldwide interacted society places great demand on the gathering and division of person specific data for many new uses. Interestingly it happens at a time when more public information is available electronically. They bring an electric copy of a person that is as recognizing and individual when the info contains no obvious identifiers, such as designation and contact number.

In todays technically-driven data rich setting, how does a data holder, such as a therapeutic establishment, civic health agency, or monetary association, share personspecific data that the released evidence remain well-nigh useful yet the folks identity who are the subjects of the data cannot be resolute?

Big data and Cloud Computing, being the troublesome aspects at present, has an important influence on the Research domain and the IT world. Today, the big data applications and services have been organized or moved over to cloud for various aspects for mining, sharing, analysing or processing. Noticeable features of cloud computing like dynamic mode and high scalability make big data cheaply and effortlessly accessible to various organizations through public cloud infrastructure. The privacy-sensitive data can be revealed with less exertion by an adversary as the coupling of big data with public cloud surroundings disables some of the old privacy defence trials in cloud.

II. SCOPE AND OBJECTIVE

The congregation of digital data by governments, corporations, and individuals has created tremendous chances for knowledge- and data-based decision making. Determined by mutual welfares, or by regulations that require data to be published, there is a demand for the discussion and publication of data among various parties. Data that comprises of sensitive data on individuals, violates the privacy norm if its published in public media. Existing exercise in data printing relies mainly on strategies and plans to what can be printed and on contracts on published data usage. This tactic alone may lead to extreme data distortion or inadequate protection. This tactic alone may lead to extreme data distortion or inadequate protection.

Many cloud services require operators to part private data like electric health archives for data study or mining, carrying privacy concerns. Now, the cloud applications upsurges extremely with Big Data drift, thereby posing it as a trial for popular software tools to use such big vast data to a bearable time scale. Hence, due to scalabilty issue it is quite a challenge for the exisiting approaches to work on the private sensitive large scale data.

Data anonymization are extensively studied and broadly accepted for privacy preservation in non-interactive data sharing and discharging situations. Data anonymization aims at hiding uniqueness and/or sensitive data so that the confidentiality of an distinct is successfully preserved while some collective data can be still be visible for users for varied analysis and mining tasks. Numerous privacy models and data anonymization methods have been proposed and widely studied recently. Though, applying the old approaches on big data anonymization stances scalability and efficiency encounters because of the 3 Vs, i.e., Volume, Velocity and Variety. Following this line, an investigation on the scheme and an attempt is made to detect a solution for anonymizing big data. Lately, differential privacy has engrossed abundant consideration due to its healthy privacy promise regardless of an opponents prior information.

Though, besides the shortcomings pointed, differential privacy also mislays precision guarantees because it harvests lurid results to hide the influence of any single individual. Hence, syntactic anonymity privacy models still have hands-on effects in general data publishing and can be useful in many practical applications.

The local-recoding arrangement, also known as cell generalization, collections data sets into a set of cells at the statistics record level and anonymizes each cell separately. Current approaches for local recoding can only withstand utmost connection attacks by retaining k-anonymity privacy model, thus dropping short of defensive proximity privacy breaches. In fact, merging local recoding and proximity privacy models composed is thought-provoking and necessary when one wants anonymous data set with both low data alteration and the capability to combat closeness privacy attacks.

In this project, the objective is to analyse on the issue of privacy breaches in big data and recommend a scalable two-phase clustering approach accordingly for privacy preservation. As the satisfiability problematic of the proximity privacy model is proved to be NP-hard, it is thought-provoking and hands-on to model the problem as a clustering problem of curtailing both data alteration and proximity among sensitive values in a bunch, rather than to find an answer filling the privacy model thoroughly. Technically, a proximity-aware distance is presented over both quasi-identifier and sensitive features to enable clustering algorithms. To address the scalability issues, proposes a two-phase clustering approach consisting of the clustering and proximity-aware agglomerative clustering algorithms. The first phase ruptures the data set into t partitions that comprise analogous data with respect to quasiidentifiers. During the next or second phase, data partitions are dealt with agglomerative clustering algorithm in parallel. Design the algorithms using MapReduce in command to increase scalability by data-parallel computation over multiple computing nodes in cloud. Investigational results prove that the method can reserve the proximity privacy substantially, and can meaningfully improve the scalability and the time-efficiency of local-recoding anonymization over existing methods.

III. MOTIVATION

In this section, scrutiny on the glitches of prevailing approaches for anonymization from the viewpoints of proximity privacy and scalability is portrayed. Further, encounters of designing mountable MapReduce algorithms for proximity-aware local recoding are also recognized.

Most prevailing local-recoding methods focus on battling record linkage attacks by paying kanonymity privacy model. Though, k-anonymity miscarries to battle attribute attacks like similarity and proximity attacks. For example, if the sensitive values for the records in a QI-group of size k are identical, challengers can still link an discrete with approximately sensitive values of high poise although the QI-group satisfies k-anonymity, ensuing in privacy violation. This process mainly results from two explanations analyzed as follows. The first one is that, unlike global-recoding schemes, k-anonymity based methods for record linkage occurrences cannot be just extended for attribute linkage outbreaks. Meanwhile global-recoding outlines partition data sets according to spheres, they can be satisfied effectively in a top-down fashion. The property of global-recoding schemes ensures that k-anonymity based approaches can be extended to fight attribute linkage attacks though checking extra privacy satisfiability during each round of the top-down anonymization process. Though, the local recoding scheme fails to share the same merits because it partitions data sets in a bunching fashion where the top-down anonymization property is inappropriate. Even though top-down approach is planned for local recoding, the approach can only achieve partially local recoding because global recoding is exploited to partition data sets as the first step and local recoding is only conducted inside each partition. Thus, the approach will incur more data distortion compared with the full potential of the local-recoding system. The second reason is that most models have the stuff of non-monotonicity, which makes such models solid to achieve in a topdown way, even for global-recoding schemes. Officially, monotonicity denotes if two disjoint data subsets G1 and G2 data set satisfy a privacy model, their union G1 and G2 satisfies the model as well. Monotonicity is a requirement for top-down anonymization approaches because it ensures to find minimally anonymized data sets. Specifically, if the data set dont satisfy a privacy model, we can infer that any of its subsets will fail to satisfy the model. Thus, anonymizing data sets in a top-down fashion, one can dismiss the process if further partitioning a subset violates the privacy model. Consequently, most existing anonymization approaches become inapplicable with such privacy models. However, this approach targets the multidimensional scheme, rather than local recoding investigated herein. Also, proximity is not integrated into the search metric that guides data partitioning in the twostep approach, potentially incurring high data distortion. Current clustering approaches on anonymization are inherently sequential and assume that the data sets processed can fit into memory. Unfortunately, the statement frequently fail in many of the big data applications in cloud nowadays. Accordingly, the approaches often suffer from the scalability problem when it comes to big data applications. Even if a single machine with huge memory could be offered, the I/O cost of reading/writing big data sets in a serial manner will be quite high. Thus, parallelism is by far the best choice for big data based applications. Utilizing a bunch of small and cheap computation nodes rather a large expensive one is more cost effective, which also coheres to the spirits of cloud computing where computation is used in various forms in the virtual machines. The attempt is to influence MapReduce in addressing the scalability issue of clustering approaches for anonymization. However, designing appropriate MapReduce jobs on complex applications is st a challenge still, as MapReduce is a constrained programming paradigm. Usually, it is essential to consider the problems like which part of an application can be parallelized by MapReduce, how to design Map as well as Reduce functions to make them scalable, and how to reduce network traffics among worker nodes. The answers to these questions often vary for different applications. Hence, extensive research is still required to design MapReduce jobs for a specific application.

IV. BIG DATA AND MAP REDUCE BASICS

Big data is a concept of working with data sets so big or complex that traditional data handling application software is insufficient to pact with. Big data trials include data storage, capturing data, data analysis, sharing, search, visualization, transferring, updating, querying and data privacy. Lately, the term ”big data” tends to refer to the use of prognostic analytics, user behavior analytics, or some other progressive data analytics methods that excerpt value from data, and rarely to a specific size of data set. There is little reluctance of data quantity accessible are certainly huge, but thats not the most pertinent specification of this new data system. Investigation of data sets could be found on new associations to business inclinations, prevent diseases, fight crime and so on.” Experts, business executives, physicians of medicine, publicity and governments similar regularly meet hitches with big data-sets in areas as well as business informatics. Experts meeting boundaries in e-Science, that includes genomics, climatology, connectomes, biology, complex physics simulations and ecological research. Big data is mainly identified by the below features:

1. Volume - The measure of completed and stored data. Scope of data governs whether it can essentially be considered big data.

2. Velocity - Speediness at which the data is produced and processed to meet the strains and trials that lie in the path of progress and development.

3. Variety - The nature of data. It helps users to efficiently use the resulting insight.

4. Veracity - The data quality of captured data can differ significantly, distressing the exact analysis

5. Variability - Discrepancy of the data set can hinder processes to hold and manage it.

MapReduce, a similar and dispersed significant data processing model, is the lengthily investigated and extensively accepted for big data applications recently. MapReduce becomes much more powerful, elastic and lucrative due to the noticeable features of cloud computing. A typical instance is the Amazon Elastic Map- Reduce service. Essentially, a MapReduce job contains of two innovative jobs - Map and Reduce, which is defined over the data structure key-value pair depicted as (key, value). Map job can be formalized i.e., Map get the pair (k1, v1) as input and subsequently output the intermediate key-value pair (k2, v2). These are expended by the Reduce job as input. Officially, the Reduce job can be characterized as: i.e., Reduce get the intermediate k2 and its values list as input which provide another pair (k3; v3) as output. Typically, (k3, v3) list is the outcomes which MapReduce users want to obtain. Map and Reduce are specified by users according to their specific applications. An occurrence running a Map function is called Mapper, and that running a Reduce function is called Reducer, respectively.

V. PROBLEM DEFINITION

Considerable number of big data products and services are installed or migrated onto the cloud for data analysis, processing or sharing. Data sets on numerous big data applications frequently comprise personal privacy-sensitive data such as financial transaction records and electronic health records that has to be further worked upon for Privacy Preservation. Traditional approaches do not hold efficient in Big data on Cloud on the Scalability and Time Efficiency parameter. When treating huge data sets the problem of scalability problem exists for all the traditional approaches. Centralized TDS methods uses data structure to increase scalability and effectiveness through indexing anonymous records of data and retaining data. Data structure thus speed up the specialism progression since indexing structure evades regularly skimming full data sets and storing numerical outcomes evades recompilation over-heads. On extra side, the quantity of metadata taken to preserve the numerical data and linkage data of record dividers is comparatively huge compared to the data sets themselves, thus overriding necessary memory. Moreover, overheads experienced by preserving the linkage structure there by updating the statistic data and will be enormous once date sets is large. Due to this, centralize methods perhaps suffer after low effectiveness and scalability on treating huge data sets.

VI. Implementation Methodology

The scope of the work includes coming up with an optimized approach that would reduce the scalability issue in big data while giving in maximum efficiency in the Data consistency with Privacy Preservation while minimal distortion. Design of the system is done as per the norms of the large-scale data involved. The dataset is obtained from the reputed source to further implement the anonymization. Algorithm is designed for the same and is implemented using Map Reducer. A hybrid approach based on K-Anonymity Generalization and Two- Phase Top Down Scheme implemented through Map Reduce for Data Anonymization to address the problem, timeefficiently. Taxonomy Tree is generated for the data set and the generalization is done through Map Reduce. A series of MapReduce jobs is developed and coordinated to conduct data-parallel computation.

TPTDS Using Map Reduce has the below listed advantages and limitations:

Efficient
Data Consistency
Two Phase TDS –Map Reduce
K -Anonymity Generalization
Scalability
Quality
Reduce Network traffic
Optimal Specializ ation

Figure 3.1: Advantages adapted through the hybrid approach

Advantages

1. Most advanced and appropriate for Big data in cloud.
2. No Scalability and Efficiency issues
3. Parallelization - Reduce Communication traffic Jobs and tasks run parallel.
4. Anonymity calculation results in the tight traffic. Produces m key-value pairs on each initial record, noticeably reducing the traffic.
5. Two phase anonymization process Better anonymization quality.

Limitations

1. Data splitting cause transmission overhead.
2. Though Map Reduce paradigm is simple, designing it for Big data with multiple Map Reducer functions is quite challenging.

By and large, TDS is an iterative procedure beginning from the highest area esteems in the scientific categorization trees of traits. Each round of emphasis comprises of three principle steps, to be specific, finding the best specialization, performing specialization and refreshing estimations of the look metric for the following round. Such a procedure is rehashed until the point when k-Anonymity is disregarded, to uncover the most extreme information utility. The decency of a specialization is estimated by a Search metric. TPTDS way to deal with lead the calculation required in TDS in an exceedingly versatile and proficient mold. The two periods of our approach depend on the two levels of parallelization provisioned by MapReduce on cloud. Essentially, MapReduce on cloud has two levels of parallelization, i.e., 40 work level and assignment level. Employment level parallelization implies that numerous MapReduce occupations can be executed all the while to make full utilization of cloud framework assets. Joined with cloud, MapReduce turns out to be all the more effective and flexible as cloud can offer framework assets on request, for instance, Amazon Elastic MapReduce benefit. Errand level parallelization alludes to that various mapper/reducer assignments in a MapReduce work are executed all the while over information parts. To accomplish high adaptability, parallelize different occupations on information segments in the principal stage, however the resultant anonymization levels are not indistinguishable. To acquire at long last steady mysterious informational indexes, the second stage is important to coordinate the intermediate results and facilitate anonymize whole data collections.

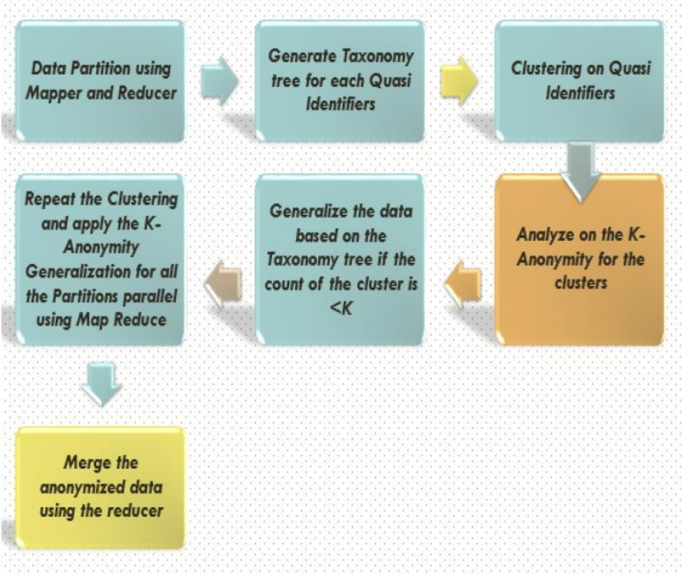

Data Partition using Mapper and Reducer
Generate Taxonomy tree for each Quasi Identifiers
Clustering on Quasi Identifiers
Analyze on the K-Anonymity for the clusters
Generalize the data based on the Taxonomy tree if the count of the cluster is <K
Repeat the Clustering and apply the K-Anonymity Generalization for all the Partitions parallel using Map Reduce
Merge the anonymized data using the reducer

Figure 3.2: Execution Methodology

In the primary stage, a unique informational index D is partitioned into smaller ones.

Run a subroutine over each of the apportioned informational indexes in parallel to make full utilization of the activity level parallelization of MapReduce. The subroutine is a MapReduce adaptation of incorporated TDS (MRTDS) which solidly directs the calculation required in TPTDS. MRTDS anonymizes information allotments to create intermediate anonymization levels.

A transitional anonymization level implies that further specialization can be performed without abusing k-Anonymity. MRTDS just use the assignment level parallelization of MapReduce. In the second stage, all intermediate anonymization levels are converged into one. The blended anonymization level is indicated as ALI

ALGORITHM 1: Two Phase TDS Scheme

Input: Data set D, number of partitions p and anonymity parameters k,k_1. **Output:** Anonymous data set D*.

1. Partition D onto D_ii,
2. Execute MRTDS (D_i; k_1;AL_0) to give AL(i), parallel multiple MapReduce jobs.
3. Merge the whole intermediate anonymization levels onto one, 4. Execute MRTDS (D_i; k_1;AL_1) to give AL* to achieve k-anonymity.

5. Specialize D according to AL*, Output D*.

In fundamental, TPTDS isolates specialization activities required for anonymization into the two stages. Give SP_1, a chance to sequence the specialization arrangement on Di in the first stage. The first subsequence of SP_1 is shown as SP_1 . Let SP_2 be the specialization succession in the second stage. SP_2 is dictated by AL_1 instead of k_1 . In particular, more AL_1 suggests smaller SP_2. Throughout TPTDS, specializations on set SP_1, SP_2 come into effect for anonymization. The influence of p and k_1 on the efficiency is analyzed as follows. Greater p and low k_1 can improve the efficiency. However, greater p and low k_1 probably lead to larger SP Extra, thereby degrading the overall efficiency. Usually, greater p causes smaller SP_1 and larger and less k1 result in larger SP_1 . The main idea of TPTDS is to get good scalability by doing a balance between scalability and the data utility. It expects that slight decrease of data utility can lead to high scalability. The influence of p and k_1 on the data utility is analyzed as follows. The information utility delivered by means of TPTDS is generally controlled by SP_1 and SP_2. High p implies that the specializations in SP_1 are chosen by IGPL values from smaller informational indexes, bringing about uncovering less information utility. Notwithstanding, more noteworthy p additionally suggests smaller SP_1 however bigger SP_2, which implies more information utility can be created in light of the fact that specializations in SP_2 are chosen agreeing a whole informational collection. Bigger k_1 shows bigger SP_2, creating more information utility. As far as the above examination, the advancement of the exchange off amongst versatility and information utility can be satisfied by tuning p and k_1. It is difficult to quantitatively plan the connections between TPTDS execution and the two parameters because it is informational index content. In any case, clients can use the subjective connections examined above to tune execution heuristically.

A. Module 1 - Data Partition

When D is partitioned into D_i, it is required that the distribution of data records in D_i is similar to D. A data record here can be treated as a point in an m-dimension space, where m is the number of attributes. Therefore, the intermediate anonymization levels resultant from the D_i, are further alike as it get a healthier merged anonymization levels.

Random sampling technique has been accepted to partition D that can gratify the mentioned requirement. Precisely, random number rand, is produced for each of the data record. A record is allocated to the partition D_r*and*. The Reducers count must be equal to P, such that each of the Reducer delay with one value of rand, resulting in producing p files.

Each file comprises of a sample random of the D

B. Module 2 - Data Specialization

A unique data collection D is solidly specialised for anonymization in a one-pass MapReduce job. In the wake of getting the consolidated moderate anonymization level AL_1, it runs MRTDS(D, k, AL) on the whole Data index D, and get the last anonymization level AL*. At that point, the data index D is valuesarea in AL*. Points of interest of Map and Reduce elements of the data specialization MapReduce work are depicted in Algorithm 2. The Map job produces anonymise records and its count. The Reduce job basically totals these anonymised records and checks their number. An anonymised record and its count represent the QI-gathering. The QI-group comprises the last anonymised data sets.

ALGORITHM 2 - MAP & REDUCE for Data Specialization

. **Input:** Data record (ID, r), r is an element of D ; Anonymization level AL*.

Output: Anonymous record (r*, count).

1. Map: Build anonymous record r*=P1, (P2,P3.. Pm,Sv), Pi, m is parent of a specialism in the current AL and is an ancestor of vi in r; emit (r*; count).
2. Reduce: For each of r*, sum ¡= count; emit (r*, sum).

C. Module 3 - MapReduce version of the Centralized TDS

It elaborates the MRTDS in this section. MRTDS assumes the center part in two-stage TDS approach, as it is summoned in the two stages to solidly direct calculation. Essentially, MapReduce program comprises of Map and Reduce capacities, and a Driver that organizes the full scale execution.

MRTDS Driver

Usually, a solitary MapReduce work is lacking to achieve a mind boggling assignment in numerous applications. Along these lines, a gathering of MapReduce employments are coordinated in a driver program to accomplish such a target. MRTDS comprises of MRTDS Driver and two sorts of jobs i.e., IGPL Initialization and IGPL Update. The driver masterminds the execution of jobs. Calculation outlines MRTDS Driver where an data index is anonymized by TDS. It is algorithmic outline of the jobs. It use anonymization level to deal with the procedure of anonymization. Stage 1 instates the estimations of information pick up and protection misfortune for all specializations, which should be possible by the activity IGPL Initialization. To start with, the best specialization is chosen from legitimate specializations in current anonymization level as depicted in Step 2.1. A specialization spec is a legitimate one in the event that it fulfill two conditions. One is that its parent value isn’t a leaf, and the other is anonymity, i.e., the data collection is still k-anonymity if spec is performed. At that point, the present anonymization level is changed by means of playing out the best specialization, i.e., evacuating the old specialization and embeddings new ones that are gotten from the old one. Data gain of the recently included specializations and security loss of all specializations should be recomputed, which are refined by work IGPL Update. The cycle proceeds until the point when all specializations wind up invalid, accomplishing the most extreme information utility. MRTDS produces an indistinguishable mysterious information from the unified TDS, in light of the fact that they take after similar advances. MTRDS primarily contrasts from Centralised TDS on figuring IGPL values. IGPL value rules the versatility of TDS approaches, as it requires TDS calculations to check the measurable data of the data indexes iteratively. MRTDS uses MapReduce on cloud to make the calculation of IGPL parallel and also scalable. It show IGPL Initialization and IGPL Update along these lines. IGPL is calculated as follows:

$IGPL_{(}spec) = IG_{(}spec)/(PL_{(}spec) + 1)$ (3.1)

$IG_{(}spec)$ is Information Gain post spec and $PL_{(}spec)$ is privacy loss both of which is computed through the statistics from datasets

ALGORITHM 3 - TWO PHASE TDS with Clustering and K- Anonymity

Input : Dataset D, Anonymity Parameter

Ouput : The anonymous Dataset D*

1. Partition the data D into b-Files using Random approach. These Files are assumed to be in separate systems. Partition Parameter to be decided based on the System availability. Partition done through a single Mapper Reducer. Output files are produced on the Partition parameter. Partition Parameter =3, implies 3 output files with the Data divided into 3 randomly. Random Generator function is hence used.
2. Generate the Taxonomy tree for the Quasi attributes. Import the class Taxonomy Manager to analyze and the draw the Taxonomy tree
3. For each Quasi Attributes get the clusters to analyze on the K- factor w.r.t KAnonymity. Decide on the K value. Current value for K = 10
4. If the count of the cluster is less than K, generalize by replacing the value with a general one from the Taxonomy tree until the K-Anonymity is met.
5. The above steps are repeated for all the b-Files parallel using the Map Reducer.
6. Combine the entire generalized data using a reducer to generalized data D*.

Dataset

The Dataset opted is popularly known as Adult Dataset. This datset was taken out by Ronny Kohavi and Barry Becker from the database for 1994 census. Set of clean data was taken to accommodate the requirements. Around 50K records were extracted with around 14 attributes. The project currently look upon 10 attributes of which 7 attributes (except numerals) are selected as quasi identifiers.

VII. Software/Hardware Specifications

Software/Hardware specfication for the undertaking is as follows:

1. Hadoop . v1.2.1
2. Eclipse . vNeon
3. JAVA .1.7
4. Ubuntu (OS)
5. The implementation is done on Ubuntu Operating System with Hadoop (1.2.1).

VIII. System Architecture and Execution Process

To expand how data collections are handled in MRTDS, the execution structure in light of standard MapReduce is outlined in the Figure. The strong bolt lines depict the data streams in the standard MapReduce structure.

It can be seen that the cycle of MapReduce jobs is controlled by anonymization level AL in Driver. The data streams for taking care of iterations are signified by dashed bolt lines. AL is dispatched from Driver to all specialists including Mappers and Reducers through the circulated reserve system.

The estimation of AL is altered in Driver as indicated by the yield of the IGPL Initialization or IGPL Update jobs. As the measure of such data is to a great degree little contrasted and dataal collections that will be anonymized, they can be proficiently transmitted amongst Driver and workers. It embraces Hadoop an open-source execution of MapReduce, to actualize MRTDS.

Since the vast majority of Map and Reduce functions need to get to current anonymization level AL, It utilize the disseminated store component to pass the substance of AL to every Mapper or Reducer hub as appeared in Figure. Additionally, Hadoop gives the component to set straightforward worldwide factors for Mappers and Reducers. The best specialization is passed into the Map function of IGPL Update job along these lines.

The parcel hash work in the rearrange stage is adjusted on the grounds that the two jobs require that the key-value sets with a similar key: p field instead of whole key ought to go to a similar Reducer. The Reduce job in the Algorithm can in any case accurately register anonymity without monitoring the substance of qid.

Dataset is taken from the UCI Machine Learning Repository. It is to foresee whether salary surpasses $50K/yr in light of evaluation data. Otherwise called ”Census Income” dataset. Abstraction was finished by Barry Becker from the 2005 Census database. An arrangement of sensibly clean records was removed. There are 14 attributes on which 4 are decided to be the Quasi Identifiers. There are 50K records in the Dataset.

The three MapReduce jobs are coordinated together to accomplish the local-recoding anonymization. The approach is fully parallel on the standpoint of data flow. The light solid arrow outlines in Fig. 4.3 represent data flows in the canonical MapReduce framework, while the dashed arrow lines stand for data flows of dispatching seeds to distributed caches and the data flow of updating seeds. The initial data set is read by Map functions and its splits are processed in a parallel manner. As such, the two-phase clustering.

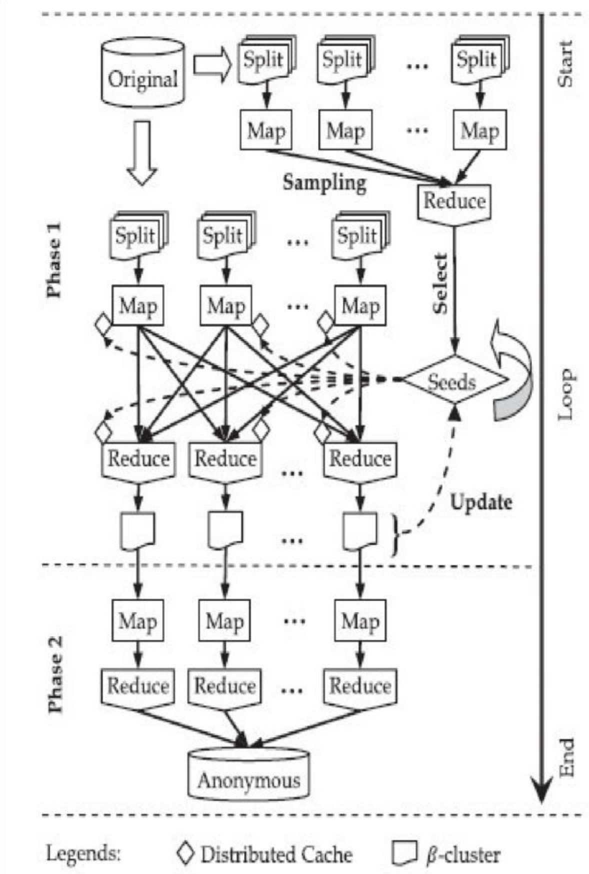


Figure 3.3: Execution framework overview of MRTDS.

approach can handle large-scale data sets. Note that the amount of seeds (or ancestors) in the Seed-Update job is relatively small with proper parameter t, so that they can be delivered to distributed caches efficiently.

IX. Output Analysis

The initial dataset has the entire data without any generalisation and highly prone to privacy breaches. The quasi identifier attributes are selected based on the causal analysis. These attributes are provided in the attributes.txt file. Partition of the dataset is based on the number of nodes available and the partition parameter is selected accordingly. Partition is done based on random function.

For each quasi attributes a Taxonomy tree is generated. Search on the records are done based on the attributes on an incremental basis. If the number of records is less than the k value selected, generalisation needs to be done. Anonymization level is deduced for the respective attribute based on the Taxonomy tree. The next level values are assumed for the attributes and the IG is calculated for the respective Anonymization levels. Information Gain is calculated based on the Entropy value. The Anonymization Level with the maximum IG along with K value condition acceptance is selected as the optimum Anonymization Level. Data Specialization of the data based on the Final Anonymization level is applied on the dataset to get the final Generalised data. There are four Map

Reducers to implement this at

1. Partition
2. Anonimization Level initialization
3. Anonymization Level Updation
4. Data Specialization.

The intermediate results at these output help to analyse on the output received for the dataset. During the Mapper function, the Anonymization Level, Node and the next level is passed analyse on the Anonymization level optimization. The driver facilitates the mapping pf the respective Mapper and Reducer functions. Combiner gives the summative count of the records before taken in by the reducer. At Reducer the Information Gain is calculated further to attain the optimized level value. The condition is further analysed with the K-Parameter condition to get the Final Anonymization Level. Data Specialization is finally done based on the final Anonmization Level.

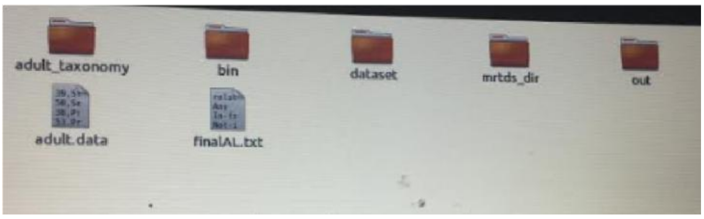
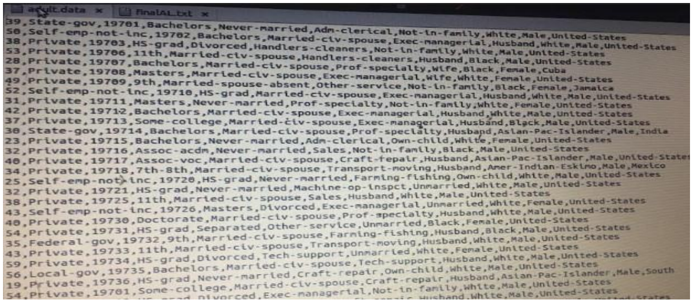


Figure 4.1: Output Folder Structure



```
39,State-gov,19701,Bachelors,Never-married,Adm-clerical,Not-in-family,White,Male,United-States
50,Self-emp-not-inc,19702,Bachelors,Married-civ-spouse,Exec-managerial,Husband,White,Male,United-States
38,Private,19703,HS-grad,Divorced,Handlers-cleaners,Not-in-family,White,Male,United-States
53,Private,19706,11th,Married-civ-spouse,Handlers-cleaners,Husband,Black,Male,United-States
28,Private,19707,Bachelors,Married-civ-spouse,Prof-specialty,Wife,Black,Female,Cuba
37,Private,19708,Masters,Married-civ-spouse,Exec-managerial,Wife,White,Female,United-States
49,Private,19709,9th,Married-spouse-absent,Other-service,Not-in-family,Black,Female,Jamaica
52,Self-emp-not-inc,19710,HS-grad,Married-civ-spouse,Exec-managerial,Husband,White,Male,United-States
31,Private,19711,Masters,Never-married,Prof-specialty,Not-in-family,White,Female,United-States
42,Private,19712,Bachelors,Married-civ-spouse,Exec-managerial,Husband,White,Male,United-States
37,Private,19713,Some-college,Married-civ-spouse,Exec-managerial,Husband,Black,Male,United-States
30,State-gov,19714,Bachelors,Married-civ-spouse,Prof-specialty,Husband,Asian-Pac-Islander,Male,India
23,Private,19715,Bachelors,Never-married,Adm-clerical,Own-child,White,Female,United-States
32,Private,19716,Assoc-acdm,Never-married,Sales,Not-in-family,Black,Male,United-States
40,Private,19717,Assoc-voc,Married-civ-spouse,Craft-repair,Husband,Asian-Pac-Islander,Male,United-States
34,Private,19718,7th-8th,Married-civ-spouse,Transport-moving,Husband,Amer-Indian-Eskimo,Male,Mexico
25,Self-emp-not-inc,19720,HS-grad,Never-married,Farming-fishing,Own-child,White,Male,United-States
32,Private,19721,HS-grad,Never-married,Machine-op-inspct,Unmarried,White,Male,United-States
38,Private,19725,11th,Married-civ-spouse,Sales,Husband,White,Male,United-States
43,Self-emp-not-inc,19726,Masters,Divorced,Exec-managerial,Unmarried,White,Female,United-States
40,Private,19730,Doctorate,Married-civ-spouse,Prof-specialty,Husband,White,Male,United-States
54,Private,19731,HS-grad,Separated,Other-service,Unmarried,Black,Female,United-States
35,Federal-gov,19732,9th,Married-civ-spouse,Farming-fishing,Husband,Black,Male,United-States
43,Private,19733,11th,Married-civ-spouse,Transport-moving,Husband,White,Male,United-States
59,Private,19734,HS-grad,Divorced,Tech-support,Unmarried,White,Female,United-States
56,Local-gov,19735,Bachelors,Married-civ-spouse,Tech-support,Husband,White,Male,United-States
19,Private,19736,HS-grad,Never-married,Craft-repair,Own-child,White,Male,United-States
54,Private,19701,Some-college,Married-civ-spouse,Craft-repair,Husband,Asian-Pac-Islander,Male,South
                                   Divorced,Exec-managerial,Not-in-family,White,Male,United-States
```

Figure 4.2: Adult Dataset.

X. Output Evaluation

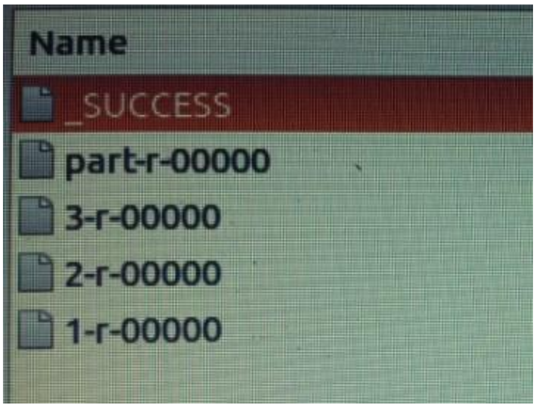
The Dataset extracted has limited number of records and attributes as depicted in the screenshots below. On executing the Hadoop code base from Eclipse, finally generalized data is derived at the output for the Reducers. Intermediate output are generated at each of the combiners and reducers to analyse on the output values. The output folder structure is as depicted in Fig: 4.1. Partitioned Dataset are navigated to the ouput folders based on the partition parameter. The partition parameter is decided to be 3. Hence have 3 files with randomly distributed files. Initial dataset is displayed as in Fig: 4.2. The partitioned data is displayed in the Partitioned Data folder as shown in

Fig : 4.3. This is implemented through a single Map-Reducer. The partition parameter can be varied based on the nodes available.

The attributes are fed into the program for identification. This can be generalized or customized. Its updated in the AttributeList.txt as shown in Fig: 4.5

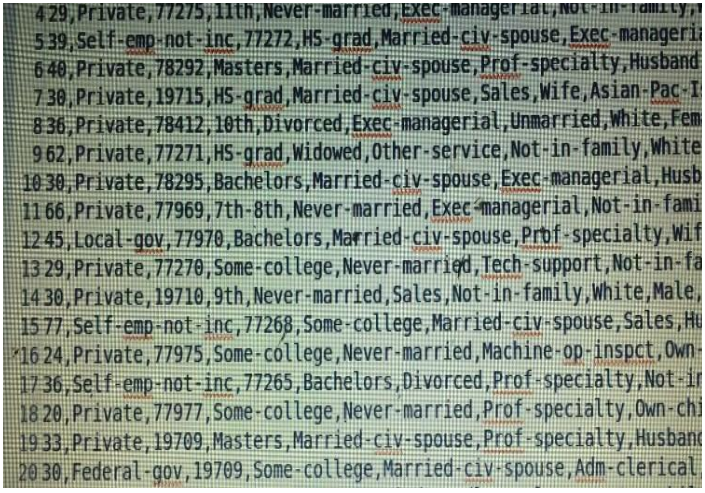
Mapper and Combiner function has the output folder where the outputs are paved to. This is depicted in Fig:4.6.

.



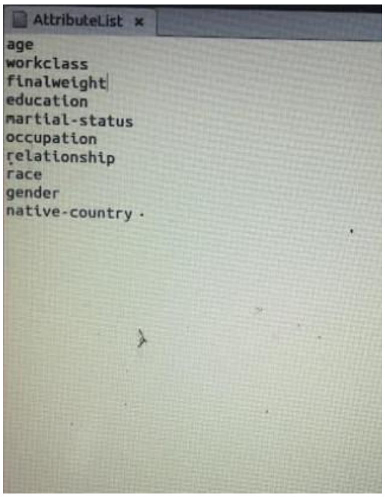
```
Name
_SUCCESS
part-r-00000
3-r-00000
2-r-00000
1-r-00000
```

Figure 4.3: Partitioned Datasets.



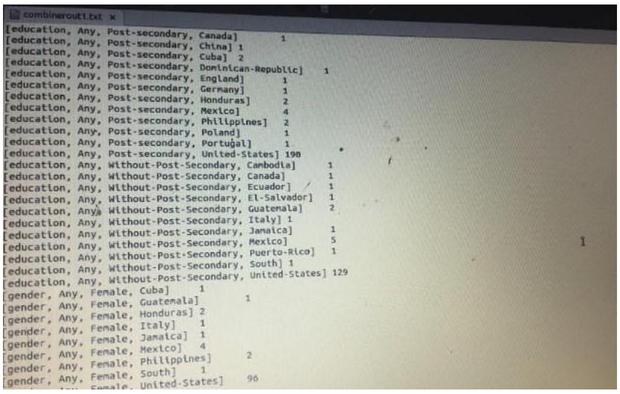
```
29,Private,77275,11th,Never-married,Exec-managerial,Not-in-family,
39,Self-emp-not-inc,77272,HS-grad,Married-civ-spouse,Exec-manageri
40,Private,78292,Masters,Married-civ-spouse,Prof-specialty,Husband
30,Private,19715,HS-grad,Married-civ-spouse,Sales,Wife,Asian-Pac-I
36,Private,78412,10th,Divorced,Exec-managerial,Unmarried,White,Fem
62,Private,77271,HS-grad,Widowed,Other-service,Not-in-family,White
30,Private,78295,Bachelors,Married-civ-spouse,Exec-managerial,Husb
66,Private,77969,7th-8th,Never-married,Exec-managerial,Not-in-fami
45,Local-gov,77970,Bachelors,Married-civ-spouse,Prof-specialty,Wif
29,Private,77270,Some-college,Never-married,Tech-support,Not-in-fa
30,Private,19710,9th,Never-married,Sales,Not-in-family,White,Male,
77,Self-emp-not-inc,77268,Some-college,Married-civ-spouse,Sales,Hu
24,Private,77975,Some-college,Never-married,Machine-op-inspct,Own-
36,Self-emp-not-inc,77265,Bachelors,Divorced,Prof-specialty,Not-ir
20,Private,77977,Some-college,Never-married,Prof-specialty,Own-chi
33,Private,19709,Masters,Married-civ-spouse,Prof-specialty,Husband
30,Federal-gov,19709,Some-college,Married-civ-spouse,Adm-clerical
```

Figure 4.4: Partitioned Dataset file

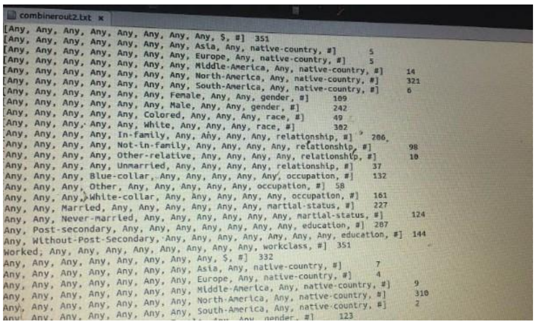


```
age
workclass
finalweight
education
martial-status
occupation
relationship
race
gender
native-country
```

Figure 4.5: Attribute List

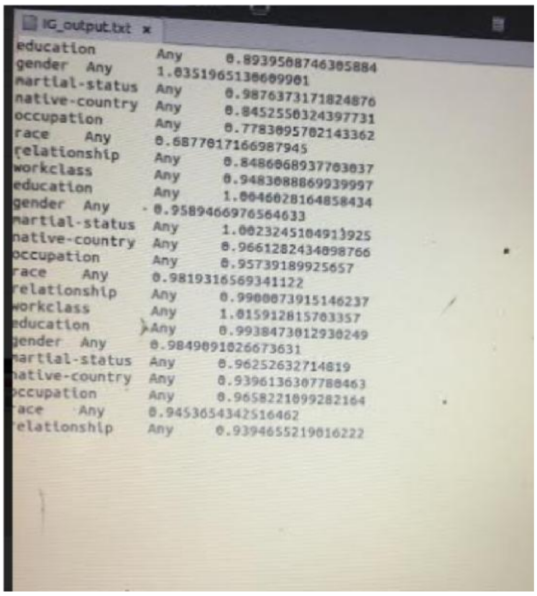


```
[education, Any, Post-secondary, Canada]        1
[education, Any, Post-secondary, China] 1
[education, Any, Post-secondary, Cuba]  2
[education, Any, Post-secondary, Dominican-Republic]    1
[education, Any, Post-secondary, England]       1
[education, Any, Post-secondary, Germany]       1
[education, Any, Post-secondary, Honduras]      2
[education, Any, Post-secondary, Mexico]        4
[education, Any, Post-secondary, Philippines]   2
[education, Any, Post-secondary, Poland]        1
[education, Any, Post-secondary, Portugal]      1
[education, Any, Post-secondary, United-States] 190
[education, Any, Without-Post-Secondary, Cambodia]      1
[education, Any, Without-Post-Secondary, Canada]        1
[education, Any, Without-Post-Secondary, Ecuador]       1
[education, Any, Without-Post-Secondary, El-Salvador]   1
[education, Any, Without-Post-Secondary, Guatemala]     2
[education, Any, Without-Post-Secondary, Italy] 1
[education, Any, Without-Post-Secondary, Jamaica]       1
[education, Any, Without-Post-Secondary, Mexico]        5
[education, Any, Without-Post-Secondary, Puerto-Rico]   1
[education, Any, Without-Post-Secondary, South] 1
[education, Any, Without-Post-Secondary, United-States] 129
[gender, Any, Female, Cuba]     1
[gender, Any, Female, Guatemala]        1
[gender, Any, Female, Honduras] 2
[gender, Any, Female, Italy]    1
[gender, Any, Female, Jamaica]  1
[gender, Any, Female, Mexico]   4
[gender, Any, Female, Philippines]      2
[gender, Any, Female, South]    1
[gender, Any, Female, United-States]    96
```



```
[Any, Any, Any, Any, Any, Any, Any, Any, $, #]  351
[Any, Any, Any, Any, Any, Any, Any, Asia, Any, native-country, #]       5
[Any, Any, Any, Any, Any, Any, Any, Europe, Any, native-country, #]     5
[Any, Any, Any, Any, Any, Any, Any, Middle-America, Any, native-country, #]     14
[Any, Any, Any, Any, Any, Any, Any, North-America, Any, native-country, #]      321
[Any, Any, Any, Any, Any, Any, Any, South-America, Any, native-country, #]      6
[Any, Any, Any, Any, Any, Any, Female, Any, Any, gender, #]     109
[Any, Any, Any, Any, Any, Any, Male, Any, Any, gender, #]       242
[Any, Any, Any, Any, Any, Colored, Any, Any, Any, race, #]      49
[Any, Any, Any, Any, Any, White, Any, Any, Any, race, #]        302
[Any, Any, Any, Any, In-family, Any, Any, Any, Any, relationship, #]    206
[Any, Any, Any, Any, Not-in-family, Any, Any, Any, Any, relationship, #]        98
[Any, Any, Any, Any, Other-relative, Any, Any, Any, Any, relationship, #]       10
[Any, Any, Any, Any, Unmarried, Any, Any, Any, Any, relationship, #]    37
[Any, Any, Any, Blue-collar, Any, Any, Any, Any, Any, occupation, #]    132
[Any, Any, Any, Other, Any, Any, Any, Any, Any, occupation, #]  58
[Any, Any, Any, White-collar, Any, Any, Any, Any, Any, occupation, #]   161
[Any, Any, Married, Any, Any, Any, Any, Any, Any, martial-status, #]    227
[Any, Any, Never-married, Any, Any, Any, Any, Any, Any, martial-status, #]      124
[Any, Post-secondary, Any, Any, Any, Any, Any, Any, Any, education, #]  207
[Any, Without-Post-Secondary, Any, Any, Any, Any, Any, Any, Any, education, #]  144
[Worked, Any, Any, Any, Any, Any, Any, Any, Any, workclass, #]  351
[Any, Any, Any, Any, Any, Any, Any, Any, $, #]  332
[Any, Any, Any, Any, Any, Any, Any, Asia, Any, native-country, #]       7
[Any, Any, Any, Any, Any, Any, Any, Europe, Any, native-country, #]     4
[Any, Any, Any, Any, Any, Any, Any, Middle-America, Any, native-country, #]     9
[Any, Any, Any, Any, Any, Any, Any, North-America, Any, native-country, #]      310
[Any, Any, Any, Any, Any, Any, Any, South-America, Any, native-country, #]      2
```

Figure 4.7: Combiner Outputs



```
education       Any     0.8939508746305884
gender  Any     1.0351965130609901
martial-status  Any     0.9876373171824876
native-country  Any     0.8452550324397731
occupation      Any     0.7783095702143362
race    Any     0.6877017166987945
relationship    Any     0.8486068937703037
workclass       Any     0.9483088869939997
education       Any     1.0046028164858434
gender  Any     0.9589466976564633
martial-status  Any     1.0023245104913925
native-country  Any     0.9661282434098766
occupation      Any     0.95739189925657
race    Any     0.9819316569341122
relationship    Any     0.9900073915146237
workclass       Any     1.015912815703357
education       Any     0.9938473012930249
gender  Any     0.9849091026673631
martial-status  Any     0.96252632714819
native-country  Any     0.9396136307780463
occupation      Any     0.9658221099282164
race    Any     0.9453654342516462
relationship    Any     0.9394655219016222
```

Figure 4.9: IG calculated for each attributes

The IG(Information Gain) is calculated and is updated in a text file as depicted in the screenshot Fig: 4.9.

Final Anonimizsed Dataset

The Final Anonymization Level that is derived from the entropy is updated in the text file. Based on the Final Anonymised Level the Final Generalised Dataset is derived as depicted in Fig:4.10. This is done through generalising the data using the Anonymised level selected from the Taxonomy tree.

Hence a complete generalised dataset is derived that satisfies the K-Anonymity as well as with optimum Information Gain.

XI. CONCLUSION\

The scalability issue of vast scale information anonymization by TDS is investigated, and proposed a profoundly versatile Two-stage TDS approach utilizing MapReduce for Big Data. Data collections are divided and anonymized in parallel in the principal stage, creating intermediate results. At that point, the moderate outcomes are combined and anonymized to create predictable k-anonymous data collections in the second stage. MapReduce on cloud to data anonymization and intentionally planned a gathering of inventive MapReduce jobs to solidly achieve the specialization calculation in an exceedingly adaptable manner. The proposed approach demonstrates the adaptability and proficiency of TDS as enhanced essentially enhanced over existing methodologies. In cloud environment, the privacy preservation for data analysis, share and mining is a challenging research issue due to increasingly larger volumes of data sets, thereby requiring intensive investigation. Improved balanced scheduling approaches are probable to be settled towards complete scalable privacy protection aware data set forecasting. The proposed approach has the below highlights

- Most advanced Most appropriate for Big data on cloud
- Comparatively no scalability and efficiency issues.
- Parallelization - Reduce Network or Communication traffic Jobs run parallel.
- Two phase anonymization process Better anonymization quality.

The contributions of this project are as follows:

- Most advanced Most appropriate for Big data on cloud
- Comparatively no scalability and efficiency issues.
- Parallelization - Reduce Network or Communication traffic Jobs run parallel.
- Two phase anonymization process Better anonymization quality.

XII. FUTURE SCOPE AND RECOMMENDATION

- An extended proximity privacy model is put forth via allowing multiple sensitive attributes and semantic proximity of categorical sensitive values.
- The proposal models the issue of big data against proximity privacy breaches.
- A scalable and efficient two-phase clustering method is well proposed to parallelize jobs on multiple data partitions
- MapReduce jobs are designed and coordinated to concretely conduct data-parallel computation for scalability making it time efficient.

The proposed method has the generalization and finally suppression been implemented using Map Reducers to get a parallel execution of jobs/tasks with maximum consideration on efficiency, scalability and data consistency. Instead of going in for suppression, randomization can be applied to anonymize the data where the K-anonymity is violated even after considering the last level of the taxonomy tree. Its a proposal on making one more step towards optimizing the anonymization there by ensuring the privacy protection of Big Data.

References

[1] Monali S Bachav, Amitkumar Manetkar A Survey on Two-Phase Top-Down Specialization for Data Anonymization using Map Reduce on Cloud , International Journal of Computer Applications (0975 8887) Innovations and Trends in Computer and Communication Engineering (ITCCE -2014)

[2] Xuyun Zhang and Jian Pei Proximity-Aware Local-Recoding Anonymization with MapReduce for Scalable Big Data Privacy Preservation in Cloud, IEEE Transactions on Computers, Vol. 64, NO. 8, August 2015

[3] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression, International Journal on Unsomety, Fuzziness and Knowledge-based Systems, 10 (5), 2002; 571588.

[4] Xuyun Zhang, Laurence T. Yang. A Scalable Two-Phase Top-Down Specialization Approach for Data Anonymization Using MapReduce on Cloud,IEEE Transactions on Parallel and Distributed Systems, Vol. 25, NO. 2, February 2014

[5] Benjamin C M Fung. Privacy Preserving in Data Mining Using Hybrid Approach,2012 Fourth International Conference on Computational Intelligence and Communication Networks

[6] N. Mohammed, B. Fung, P.C.K. Hung, and C.K. Lee Centralized and Distributed Anonymization for High-Dimensional Healthcare, ACM Trans. Knowledge Discovery from Data, vol. 4, no. 4, Article 18, 2010.

[7] H. Takabi, J.B.D. Joshi, and G. Ahn. Security and Privacy Challenges in Cloud Computing Environments. .,IEEE Security and Privacy, vol. 8, no. 6, pp. 24-31, Nov. 2010.

[8] B.C.M. Fung, K. Wang, and P.S. Yu, Anonymizing Classification Data for Privacy Preservation,, IEEE Trans. Knowledge and Data Eng., vol. 19, no. 5, pp. 711-725, May 2007.

[9] X. Xiao and Y. Tao. Anatomy: Simple and Effective Privacy Preservation. , Proc. 32nd Intl Conf. Very Large Data Bases (VLDB06), pp. 139-150, 2006.

[10] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan. Incognito: Efficient FullDomain K-Anonymity, Proc. ACM SIGMOD Intl Conf. Management of Data (SIGMOD 05), pp. 49-60, 2005.

[11] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan, Mondrian Multidimensional K-Anonymity, Proc. 22nd Intl Conf. Data Eng. (ICDE 06), 2006.