Journal of Big Data

RESEARCH

Open Access

Privacy preserving data publishing based on sensitivity in context of Big Data using Hive

P. Srinivasa Rao[1*] and S. Satyanarayana[2]

*Correspondence:
psr.sri@gmail.com
[1] Computer Science and Engineering, MVGR College of Engineering, Chintalavalasa, India
Full list of author information is available at the end of the article

Abstract

Privacy preserving data publication is the main concern in present days, because the data being published through internet has been increasing day by day. This huge amount of data was named as Big Data by its size. This project deals with the privacy preservation in context of big data using a data warehousing solution called hive. We implemented nearest similarity based clustering (NSB) with Bottom-up generalization to achieve (v,l)-anonymity which deals with the sensitivity vulnerabilities and ensures the individual privacy. We also calculate the sensitivity levels by simple comparison method using the index values, by classifying the different levels of sensitivity. The experiments were carried out on the hive environment to verify the efficiency of algorithms with big data. This framework also supports the execution of existing algorithms without any changes. The model in the article outperforms than existing models.

Keywords: Sensitivity, Sensitive level, Clustering, PPDP, Bottom-up generalization, Big Data

Introduction

As a part of information sharing through internet every organization publishes the personal data which they collect from different users [1]. This published data may disclose personal private information. The data provided by the corporations, government and individuals will create enormous opportunities for individual knowledge based decision making [2]. In consideration of the mutual benefits or by the rules that require to publish the data, there is a demand for exchange or publication of data among various parties. Personal data in its actual form, however, typically contains individual sensitive information and if this data published as it is then that kind of data will violate the individual privacy [3]. The present practice initially relies on guidelines and policies to deprive the types of publishable data and on agreements on the use and storage of sensitive data. The limitation of this approach is that it either manipulates data overly or requires a trust level that is practically very low in many present data sharing scenarios [4, 5]. For instance, contracts and agreements between any parties cannot ensure that sensitive data will not be carelessly misplaced and end up in the wrong hands.

The actual task of the data provider is to develop methods and tools for publishing data in more antagonistic environment, so that the data will be available to the needed people and satisfies the privacy of an individual. This undertaking is called privacy preserving data publishing (PPDP). Researches have been done to overcome the flaws in published data, as a result many attacks were defined and many algorithms were proposed to overcome these attacks [1, 6]. Not every algorithm solves all issues, but somehow confronts the attacks and reduces the chance of more data loss. On the other hand, attackers are coming up with new challenges to violate the privacy.

Removing identity information may not be enough to protect individual privacy. While publishing the data, organizations need to be mindful of other data sources. Because attacks happen on the published tables will not concentrate on the published table alone but attackers always tries to link more table together to reveal the data of an individual. This paper deals with Sensitivity vulnerabilities to obstruct linking attacks with Nearest similarity based clustering and Bottom-up generalization to achieve (v,l)-anonymity, and solves the performance issues with Big Data [7, 8] using Hive.

Related work

Many experiments were carried out in the area of PPDP to ensure the privacy of an individual. One of those models is K-anonymity, which ensures that the EC contains minimum of k records. But it doesn't pay more attention on sensitivity of attributes, so that the privacy may be compromised in most of the cases. K-Anonymity is the base for many researches till now. The key concept of this algorithm was used in different areas of privacy models. To avoid undesirable and unlawful effects of privacy on sequence data, while designing a technological framework, they introduce privacy-by-design [9], without distributing the knowledge discovery data mining. In this, authors used k-anonymity framework for sequence data, and notation of k-anonymity for sequence datasets provides protection against attacks. Fung et al. [10] extended the k-anonymization algorithm for cluster analysis. They achieve privacy by partitioning the original data into clusters and class labels encode the cluster information, then the k-anonymization will be achieved with clusters. Lefevre et al. [11] proposed an algorithm called Incognito, which is the collection of multiple bottom-up generalization algorithms. By this method authors tried to generate all possible K-Anonymous full-domain generalizations. Generalization is the process of substituting parent values with children, which is the method mainly used to provide privacy. Wang et al. [12] proposed Bottom-up generalization in to address the efficiency issue in k-anonymization. Data utility is also important after privacy [13]. Jordi et al. shows that data utility improvement of the published datasets by micro aggregation based K-Anonymity. Machanavajjhala et al. [14] proposed *l*-diversity in, which suggests that an EC should contain *l*-different "well represented" values of SAs. It doesn't consider any difference between different sensitive attributes. E-learning is the new way learn the courses at home using internet. In Mohd et al. [15] proposed a new model to ensure the trust in online e-learning activities. Authors used identity management (IM) to protect the privacy of learners. IM ensures the protection of personal information with some degree of participant anonymity or pseudonymity. Further, because participants can hold multiple identities or can adopt new pseudonymous personas, a reliable and trustworthy mechanism for reputation transfer (RT) from one person to another is required. Such a reputation transfer model must preserve privacy and at the same time prevent linkability of learners' identities and personas. In this paper, authors present a privacy-preserving reputation management (RM) system which allows secure transfer of reputation. Emiliano et al. [16] proposed Hummingbird, which protects the tweet contents, followers' interests and hash tags from attackers through a centralized server. This privacy concept was spreaded over different areas like Wireless Sensor Networks (WSN), where the privacy is the main concern. The wireless itself is untrusted, where an anonymous nodes can also get connected. In [17] authors proposed a novel secured method called Three-factor user authentication scheme for distributed WSNs. To avoid the collisions in communication with privacy preserving data mining Larr et al. [7] proposed an anonymous ID assignment where this ID number will iteratively assign to the nodes Drushina et al. [18] proposed a network coding method for privacy in networks by removes the statistical dependence between incoming and outgoing messages, so that tracing is not possible. Bayardo et al. [19] proposed an algorithm to prune the non optimal anonymous tables by set of enumeration tree, where each node represents k-anonymous solution. Valeria et al. [20] proposed an algorithm for machine learning operations called Ridge regression. This algorithm takes large number of data points and finds the best-fit linear curve through these points as input. Xiaokui et al. [21] proposed privacy preserving data-leak detection (DLD) solution to solve the issue where a special set of sensitive data digests is used in detection. The advantage of their method is that it enables the data owner to safely delegate the detection operation to a semi honest provider without revealing the sensitive data to the provider. Huang et al. [22] proposed a novel privacy model called (v,l)-anonymity, which mainly concentrate on the vulnerabilities in sensitivity. It supports the existing privacy models and provides the different way of privacy. Authors also propose a new method of assigning sensitive levels to the sensitive values. They define sensitivity classification and presented a measure which is called as levels of sensitive values (LSV) measure to calculate the sensitive levels. This model can also work efficiently with multiple sensitive attributes. Qinghai et al. [23] proposed a privacy-preserving data publishing method, namely MNSACM, to publish micro data with multiple numerical sensitive attributes which uses the ideas of clustering and Multi-Sensitive Bucketization (MSB). Sweeney [24] experimented using k-anonymity to identify the various attacks by considering multi-level databases. Bredereck et al. [25] adopted a data-driven approach towards the design of algorithms for k-Anonymity and related problems. Tsai et al. [26] reviewed studies on the data analytics from the traditional data analysis to the recent big data analysis. Zhang et al. [27] gave a survey of big data processing systems such as batch, stream, graph, and machine learning processing and also discussed some possible future work directions.

Methodology

Many researches' have been done in the area of privacy preserving data publication but many solutions failed to concentrate on the sensitive levels.

A Micro data Table 1 was generalized in Table 2. Here Zip code and Age are the Quasi Identifier Attributes (QIAs). In Table 1 each EC2 contains HIV and cancer, both are the severe diseases. If an attacker Know that a person 'A' is under the age 20–23 and Zip

Table 1 Micro data table

No	Age	Zip code	Disease
1	18	94131	Gastritis
2	19	94132	Flu
3	20	94133	HIV
4	23	94134	Cancer
5	31	94142	Asthma
6	32	94143	Fever
7	34	94144	Flu
8	36	94145	Heart disease

Table 2 Anonymized data

No	Age	Zip code	Disease
1	18–19	9413*	Gastritis
1	18–19	9413*	Flu
2	20–23	9413*	HIV
2	20–23	9413*	Cancer
3	31–32	9414*	Asthma
3	31–32	9414*	Fever
4	34–36	9414*	Flu
4	34–30	9414*	Heart disease

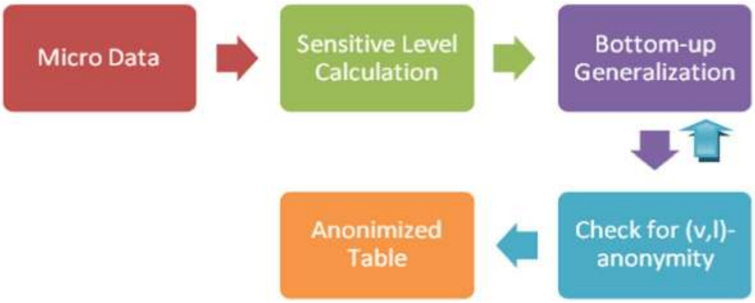


Fig. 1 Overall process of privacy preservation

code starts with 9413 then he can come to a conclusion that 'A' is suffering from very serious disease with 50% probability.

The overall process consists of three main modules:

- Sensitive level calculation.
- Nearest similarity based clustering.
- Bottom-up generalization.

As shown in Fig. 1 the overall process consists of 5 steps. In the first stage micro data will be taken as input. In second step sensitivity levels will be calculated and updated in micro data table. Bottom-up generalization will be performed on this updated table in

Table 3 Level of sensitivity

Sensitivity level	State	Level
L_1	Top severe	1
L_2	Severe	2
L_3	Less severe	3
L_4	Not severe	4

Table 4 Calculation of index value

	H1	H2	H3	H4
I1	0.64	0.76	0.83	0.95

step 3. Then (v, l)-anonymity condition will be checked in each equivalence class in the next step. If the condition satisfied then anonymized data will be published otherwise the table again goes under generalization process for the next level and again the anonymity condition will be checked and anonymized data will be published.

Sensitivity level calculation

In this section we calculate the level of sensitivity of the sensitive values and classified the sensitive values according to the sensitivity. We also proposed a sensitivity measure (LSV), to calculate the sensitivity. But the question is how to determine the level of sensitivity, and how to choose the factors to calculate the sensitivity level.

We have taken the patient data set, which contains the information about the individual patient details including the diseases they suffered from. Here, 'disease' is the sensitive attribute, so that the classification is on different diseases based on the severity. For example, HIV and Fever are the two diseases, the severity of HIV is higher. Through this measure which is based on severity, we can classify and order HIV and Fever. For diseases, there are some other categorical attributes, such as location, infectivity, etiology, as well as numeric attributes, such as mortality, cure rate and morbidity.

For example, an attribute disease contains different sub-attributes, which are categorical attributes. With the values of sub-attributes, if the sensitivity of the sensitive values remains unchanged these attributes will be remained as sub-attributes, otherwise we call them as sensitive index or simple an index. In the disease example mortality rate of HIV reflects the severity of that disease, that is if the mortality rate is high then we can consider that disease the severity of the disease is more, likewise the cure rate also determines the severity of the disease, if the cure rate is high, the severity of the disease is less. In Table 3, neglecting the sub-attributes of the sensitive values leads to the sensitivity vulnerability in the second equivalence class (EC), where HIV and cancer comes under the same EC.

Here we proposed a sensitivity classification measure, based on the values of indexes. Each index has map to the sensitivity. The index must be chosen according to the type of the data set. Not every index is suitable to all kinds of data, so that the index value changes from one data set to the other data set. We worked on the patient data set so we have chosen the mortality rate as an index to calculate the sensitivity level, which comes under the category of monotonic index which is shown in the Table 4. The increase in the mortality rate decides the decreasing in the sensitivity level, indicating that the decreasing order of their severity.

Classification phase

Let us consider F = (L_1, L_2, L_3,…L_K) are the sensitive levels, where $F = \bigcup_{i=1}^{K} Li$ and $L_i \cap L_j = \emptyset$, for i ≠ j. In our methodology, we have chosen four levels, so that 'F' will be divided into four levels as shown in Table 4.

Here $L_1 > L_2 > L_3 > L_4$, higher the sensitivity, stronger the level, where we must pay more attention on it. Let I_1, I_2, I_3,… I_h are the indexes and S_1, S_2, S_3,… S_n are the sensitive values till 'n' in the table 'T'. S_{ij} denotes the index I_j of the sensitive value S_i. For i = 1,2,…n, j = 1,2,…h, and k = 1,2,…K, Let $\mu_{ijk} = \mu(s_{ij} \in L_k)$ denotes that the S_i belongs to L_k of jth index I_j, which is called as index level measure. Let 'm' be the number of distinct diseases in the table. Let us consider the individual indexes of the diseases are d_1,d_2,d_3…d_m. For example d_1 indicates the index of fever and the value d_1 is 0.94, then index value of fever becomes 0.94. All the other indexes will be assigned in the same way. To avoid the confusion between the indexes I_1, I_2, I_3,… I_h and d_1, d_2, d_3…d_m we consider I indexes as threshold indexes, where these threshold values decides the range of the sensitivity level distribution.

The index level measure

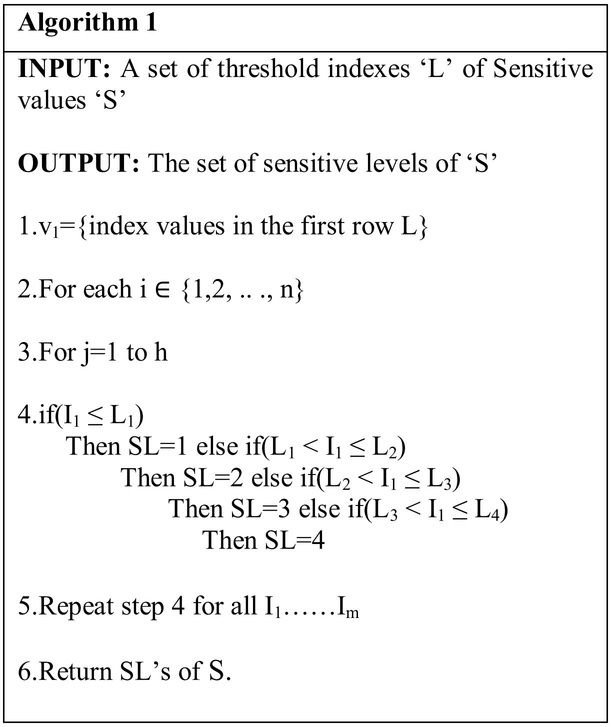
Each index has some levels. A set of index values form the basis of the corresponding level on each level of an index. The set of values on the level defines the membership degree of an index, which belongs to that level. A set of fixed index values based on the sensitivity of all values on the index gives a standard of a given level on that index. The overall process depends on applications. According to the level order, index level standard (ILS) will be formed, which is a vector. Classification of sensitivity of an index is based on index level standard. ILS of all indexes are expressed as follows:

$$L = \begin{array}{c|cccc|} & L_1 & L_2 & \cdots & L_k \\ \hline I_1 & v_{11} & v_{12} & \cdots & v_{1k} \\ I_2 & v_{21} & v_{22} & \cdots & v_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I_h & v_{h1} & v_{h2} & \cdots & v_{hk} \end{array}$$

where 'K' is the number of levels and v_{jk} is the standard of L_k, where L_k is the set of index values of I_j. L_k must satisfies $v_{ji} \cap v_{jk} = \emptyset$, for all i ≠ k. Here $\nu j1 < \nu j1 < \nu j1 \cdots < \nu jk$. The threshold index values on our experiment have shown in Table 4. These threshold values are used to compare with the individual index values of the respective diseases in the calculation of the sensitive levels and their classification.

Computation of sensitive levels

The sensitivity calculation can be done based on ILS and LSV measure as shown in Algorithm 1. Let K_j denotes the size of categories of I_j. We use only one index measure, which is mortality rate, so the value of K_j is 1. The process starts from taking the index values as input. These values are the thresholds to calculate the severity of the diseases. The index values of each disease can be static or dynamic based on the nature of the dataset. Our experiment considered the index values as static. Each individual index value will be compared with every threshold value, until and unless it satisfies the condition. Once the condition satisfied at a particular threshold, then the position of the threshold index will be the level of that particular disease. For example if the index value of fever is 0.94, this is going to be satisfied at L_4, which was in the fourth position in the Table 4. So the sensitive level of fever becomes 4. While see the level of fever we can say that it is not severe disease based on the classification done in Table 1. The same process will be continued to all diseases till the end of the dataset. Finally all the records will be allocated to the sensitive levels. Then the process will be stopped. The overall table will be updated by the newly calculated sensitive values as shown in the Table 5. The number of the sensitive levels can be determined by the publisher, who is responsible for the calculation of these levels, based on the data set. This sensitivity classification and calculation will be useful not only in information search, but also in the generalization process, which will be determined in further sections. By considering the sensitivity levels in the generalization process we can limit the sensitivity vulnerabilities in data publication.



Algorithm 1

INPUT: A set of threshold indexes 'L' of Sensitive values 'S'

OUTPUT: The set of sensitive levels of 'S'

```
1.$v_1$={index values in the first row L}

2.For each i ∈ {1,2, .. ., n}

3.For j=1 to h

4.if($I_1 \leq L_1$)
      Then SL=1 else if($L_1 < I_1 \leq L_2$)
             Then SL=2 else if($L_2 < I_1 \leq L_3$)
                   Then SL=3 else if($L_3 < I_1 \leq L_4$)
                        Then SL=4

5.Repeat step 4 for all $I_1 \ldots\ldots I_m$

6.Return SL's of S.
```

Table 5 Table after assigning SLs'

No	Age	Zip code	Disease	SL
1	18	94131	Gastritis	3
2	19	94132	Flu	4
3	20	94133	HIV	1
4	23	94134	Cancer	1
5	31	94142	Asthma	3
6	32	94143	Fever	4
7	34	94144	Flu	4
8	36	94145	Heart disease	1

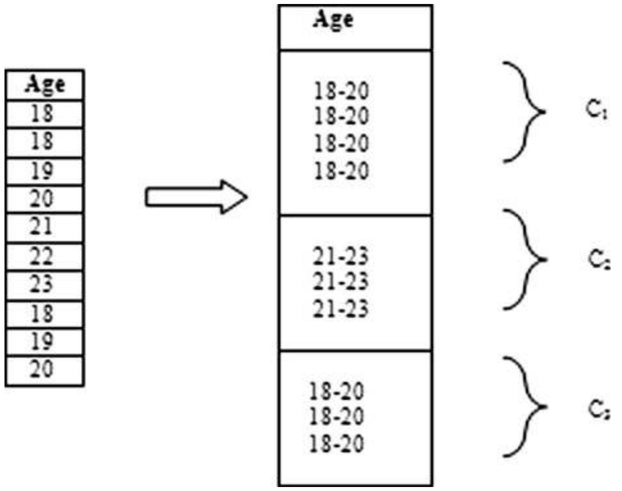


Fig. 2 Nearest similarity based clustering

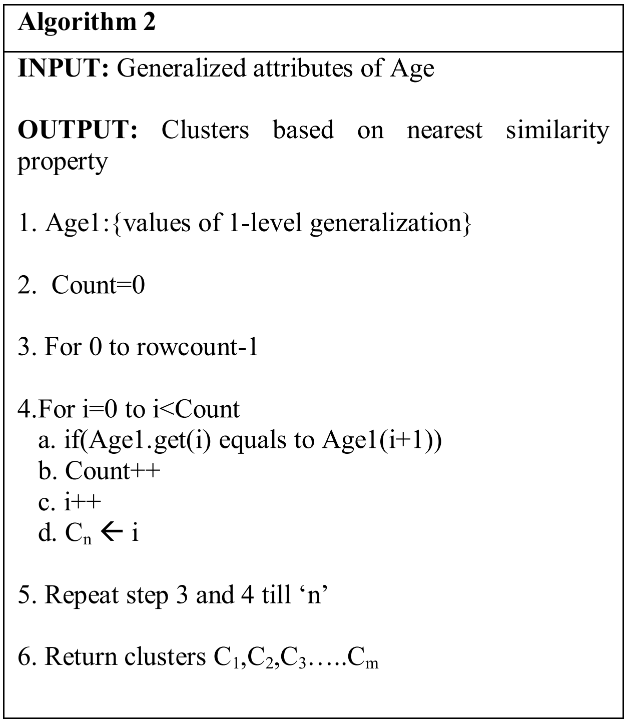
Nearest similarity based clustering

Clustering is a process of partitioning a set of data (or objects) into a set of meaningful sub-classes, called clusters. A cluster of data objects can be treated as one group. While doing cluster analysis, we first partition the set of data into groups based on data similarity and then assign the labels to the groups. The main advantage of clustering over classification is that, it is adaptable to changes and helps single out useful features that distinguish different groups.

The clustering process starts after 1st level of generalization. Every generalization process has some goals to satisfy to achieve desired level of privacy. Nearest Similarity Based Clustering as shown in Fig. 2 is also a part in the generalization goals. The decision of next level of the generalization is based on the result occurred in NSB clustering. The clustering has to be done in any of the quasi identifier set. The selection of the attribute must be taken care because the overall generalization depends on these clusters. If in case we choose the wrong attribute, the overall process leads to inconsistent result and also raises the performance issues.

All our experiments have been performed by considering 'Age' as the clustering attribute. We can also take more than one attribute for clustering. In Fig. 2 all records were divided into 3 clusters. Though it is similarity based clustering "18–20" was divided into 2 clusters c_1 and c_3, based on the nearest similarity property.

Let A_1, A_2, A_3,...A_n are the values of age after 1-level generalization till 'n'. Here 'n' represents the number of records. C_1, C_2, C_3...C_m are the clusters formed as a result of NSB clustering. 'm' represents the number of clusters. 'i' represents the current value. "Age1" represents the attribute "Age" after 1st level generalization. The algorithm is as follows:



Algorithm 2

INPUT: Generalized attributes of Age

OUTPUT: Clusters based on nearest similarity property

```
1. Age1:{values of 1-level generalization}

2.  Count=0

3. For 0 to rowcount-1

4.For i=0 to i<Count
   a. if(Age1.get(i) equals to Age1(i+1))
   b. Count++
   c. i++
   d. $C_n$ ← i

5. Repeat step 3 and 4 till 'n'

6. Return clusters $C_1$,$C_2$,$C_3$…..$C_m$
```

Bottom-up generalization

Generalization replaces the most specific value with the most generalized value. Generalization is the most common method to make the data anonymous to provide the privacy. The way of generalizing the data depends on the nature of the data and the applications. It is a hierarchical process, which can be represented in a form of three. The node at the top level will be called as parent. All the remaining nodes are child nodes. In general, generalization replaces child node values with their parent node values in a taxonomy tree. The reverse process of generalization is called as specialization. We follow the bottom-up generalization method, where the generalization process starts from the bottom and continued to top of the taxonomy tree. It means the leaf nodes will be replaced with the parent nodes, based on the level of generalization. The generalization process may stopped after first level, in case the level satisfies the anonymous conditions to get the required level of privacy. The result is shown in the Table 6. Generalization process will be done on the quasi identifier set, which will causes the re-identification or linking attacks. Not even a generalization but every anonymization techniques apply on these quasi identifier attributes. In our data set Age, Zip Code, Race, Gender are the quasi attributes.

Table 6 after 1st level of generalization

No	Age	Zipcode	Disease	Sensitive level
1	18–19	9413*	Gastritis	3
1	18–19	9413*	Flu	4
2	20–23	9413*	HIV	1
2	20–23	9413*	Cancer	1
3	31–32	9414*	Asthma	3
3	31–32	9414*	Fever	4
4	34–36	9414*	Flu	4
4	34–30	9414*	Heart disease	2

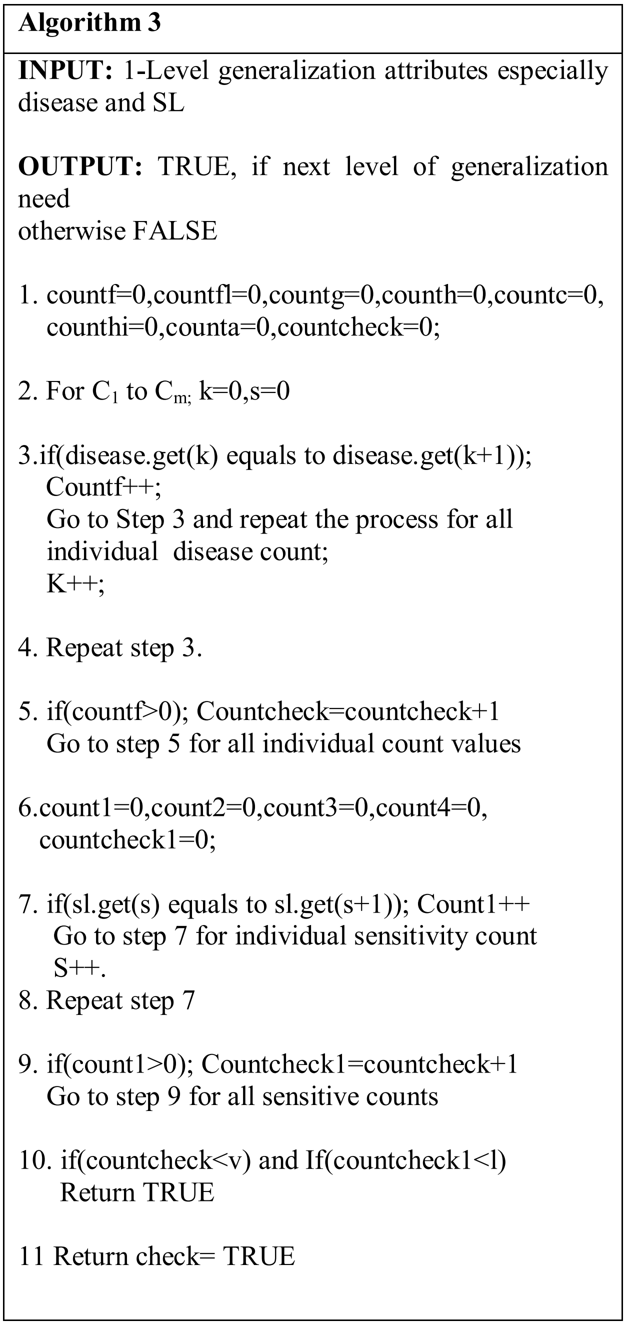
In our experiments we took 2 levels of bottom up generalization. 1st level of generalization will be done in any case, once the key value has given. The next level of generalization depends on the clusters, which were formed by NSB clustering. NSB clusters are formed after 1st level of generalization. These clusters are nothing but the equivalence classes. We cannot find any difference between the rows which were present in one equivalence class. This is the minimum requirement or the condition in any privacy preservation technique. Each cluster or equivalence class must satisfy the (v, l)-anonymity rule, which will be discussed further. If the clusters satisfy the anonymity rule then the process stopped after 1st level of generalization, otherwise 2nd level of generalization takes place. If at least one cluster do not satisfy the anonymity principle then also the next level of generalization takes place. In most of the cases the data set achieves the (v,l)-anonymity after 2nd level of generalization, or at least the data becomes more sophisticated to the attackers or the unknown users to reveal the data.

Checking for (v,l)-anonymity

(v,l)-anonymity is the best way to deal with the sensitivity vulnerabilities while publishing the data. Here ‘v’ represents the sensitive value and ‘l’ represents the sensitive level. Every cluster contain ‘v’ number of distinct well represented sensitive values and ‘l’ distinct well represented sensitive levels. For example, a table will be called as (3,2)-anonymous, if and only if each cluster or equivalence class contain ‘3’ distinct sensitive values and ’2’ distinct sensitive levels. In our patient data set, diseases are the sensitive values and sensitive values were calculates in the previous sections. So the disease attribute must contain ‘3’ different well represented diseases and ‘2’ different well represented sensitive values under the attribute sensitive levels (SLs) in order to get the (3,2)-anonymization.

While looking at the Table 6, which is the result of 1-level of generalization was not satisfied the (3,2)-anonymity condition. First of all the cluster must contain minimum of 3 values, but this table was failed there itself, and also the second cluster does not contain 2 different sensitive values, which is the main concern. So the generalization process continues with next level of generalization.

Let C_1,C_2,C_3…C_m clusters forms after 1-level generalization using NSB clustering. ‘C_m’ is the last cluster. ‘v’ represents the sensitive values, ‘l’ represents the sensitive levels. ‘k’ represents the current value in the cluster. Countf, countfl, countg, counth, countc, counthi, counta represent the frequency of diseases fever, flu, gastritis, heart disease, cancer, HIV, asthma respectively, and will be incremented each time they encounters the respective diseases. Countcheck is used to count the overall different diseases in individual cluster, and will be incremented each time when the overall count of a disease greater than 0 in a cluster. Count1, count2, count3, count4 represents the frequencies of sensitive levels 1, 2, 3, 4 respectively. Countcheck1 is used as similar as countcheck for the sensitive levels. The step by step process of checking of (v,l)-anonymity has shown below:



Algorithm 3

INPUT: 1-Level generalization attributes especially disease and SL

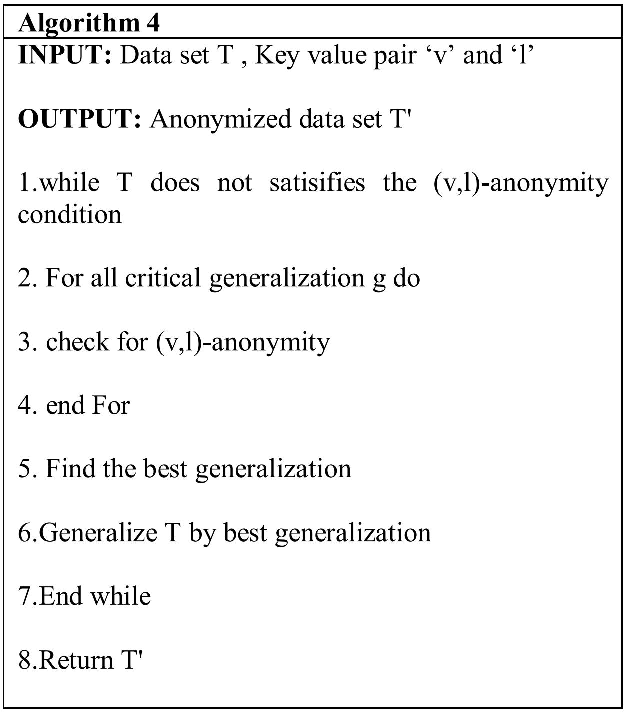
OUTPUT: TRUE, if next level of generalization need
otherwise FALSE

```
1. countf=0,countfl=0,countg=0,counth=0,countc=0,
   counthi=0,counta=0,countcheck=0;

2. For C1 to Cm; k=0,s=0

3.if(disease.get(k) equals to disease.get(k+1));
   Countf++;
   Go to Step 3 and repeat the process for all
   individual  disease count;
   K++;

4. Repeat step 3.

5. if(countf>0); Countcheck=countcheck+1
   Go to step 5 for all individual count values

6.count1=0,count2=0,count3=0,count4=0,
  countcheck1=0;

7. if(sl.get(s) equals to sl.get(s+1)); Count1++
    Go to step 7 for individual sensitivity count
    S++.
8. Repeat step 7

9. if(count1>0); Countcheck1=countcheck+1
   Go to step 9 for all sensitive counts

10. if(countcheck<v) and If(countcheck1<l)
     Return TRUE

11 Return check= TRUE
```

Table 7 Table after 2-level generalization

No	Age	Zip code	Disease	Sensitive level
1	18–23	941**	Gastritis	3
1	18–23	941**	Flu	4
1	18–23	941**	HIV	1
1	18–23	941**	Cancer	1
2	31–32	941**	Asthma	3
2	31–32	941**	Fever	4
2	31–32	941**	Flu	4
2	31–32	941**	Heart disease	2

Table 7 is the result of 2-level generalization, where the total records are divided into 2 equivalence classes and make the records more anonymize to achieve (3,2)-anonymization. We can say that each cluster contain 3 different sensitive values and 2 different sensitive levels. So we can say that the Table 7 is (3,2)-anonymous. Algorithm for Bottom-up generalization as follows:



Algorithm 4

```
INPUT: Data set T , Key value pair ‘v’ and ‘l’

OUTPUT: Anonymized data set T'

1.while T does not satisifies the (v,l)-anonymity condition

2. For all critical generalization g do

3. check for (v,l)-anonymity

4. end For

5. Find the best generalization

6.Generalize T by best generalization

7.End while

8.Return T'
```

Implementation

We have two programs which are names as vlanonymity.java and sensitivity.java. One program is to apply (v, l)-Anonymization to the table and other is to calculate the sensitivity level respectively. It is not required to update the table for vlanonymity.java because operations taken place at runtime but not saved again in the table. But it is different with sensitivity.java where it is required to update the table. A normal table creation is not sufficient to support update operations. So we have to create a table which supports bucketization that supports updations. The execution starts with calculation of sensitive levels. The execution takes place with the help of CLI. Then the query sends to the driver such as JDBC or ODBC to execute. With the help of query compiler the driver parses the query. It checks the syntax and query plan or the requirement of query. The compiler sends metadata request to Metastore. Metastore sends metadata as a response to the compiler. The compiler checks the requirement and resends the plan to the driver. Up to here, the parsing and compiling of a query is complete. The driver sends the execute plan to the execution engine. Internally, the process of execution job is a MapReduce job in case of sensitivity level calculation but not to the anonymization process. The execution engine sends the sensitivity calculation job to JobTracker, which is in Name node and it assigns this job to TaskTracker, which is in Data node. Here, the query executes MapReduce job. Meanwhile in execution, the execution engine can execute metadata operations with Metastore. The execution engine receives the results from Data nodes. The execution engine sends those resultant values to the driver. The driver sends the results to Hive Interfaces. With the help of the interface we can see the result which contains all the anonymized records in case of vlanonymity.java execution and the records with sensitive values in case of sensitivity.java execution.

The experiments were carried out in Hive environment with ubuntu 14.04LTS as operating system. Outcomes of the experiments are as follows:

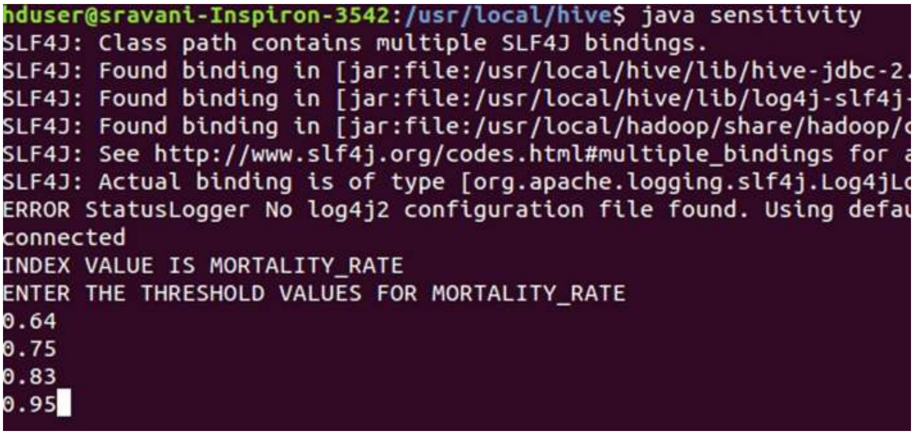
Figure 3 shows the sensitivity values with which the sensitive levels can be set.

Figure 4 shows the screenshot in which sensitivity levels are assigned to different diseases based on the sensitivity level.

Figure 5 shows the input data set considered before sensitivity calculation and the Fig. 6 shows the data set with calculated sensitivity values.

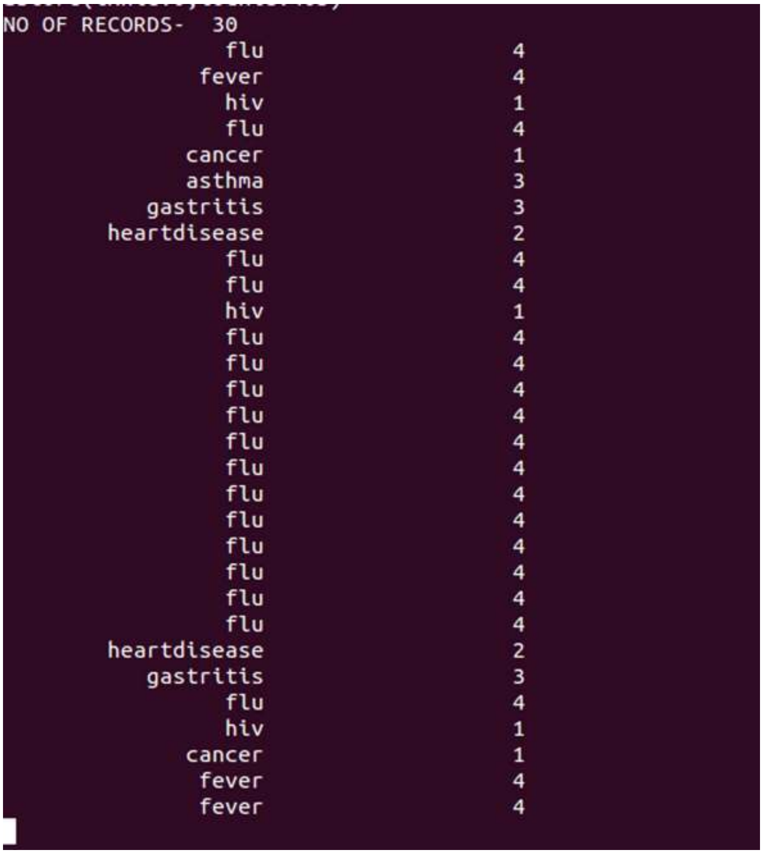
Figure 7 shows the accepting values of (v,l) values where v indicates values with which the attributes generated up to given level as l.

Figure 8 shows the screenshot of the experimentation which gives a view to know how one level hierarchy can be achieved.



```
hduser@sravani-Inspiron-3542:/usr/local/hive$ java sensitivity
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/hive-jdbc-2.
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/log4j-slf4j-
SLF4J: Found binding in [jar:file:/usr/local/hadoop/share/hadoop/c
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for a
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLo
ERROR StatusLogger No log4j2 configuration file found. Using defau
connected
INDEX VALUE IS MORTALITY_RATE
ENTER THE THRESHOLD VALUES FOR MORTALITY_RATE
0.64
0.75
0.83
0.95
```

Fig. 3 Accepting index values to calculate SL through sensitivity.java



```
NO OF RECORDS-  30
             flu                      4
           fever                      4
             hiv                      1
             flu                      4
          cancer                      1
          asthma                      3
       gastritis                      3
    heartdisease                      2
             flu                      4
             flu                      4
             hiv                      1
             flu                      4
             flu                      4
             flu                      4
             flu                      4
             flu                      4
             flu                      4
             flu                      4
             flu                      4
             flu                      4
             flu                      4
             flu                      4
             flu                      4
    heartdisease                      2
       gastritis                      3
             flu                      4
             hiv                      1
          cancer                      1
           fever                      4
           fever                      4
```

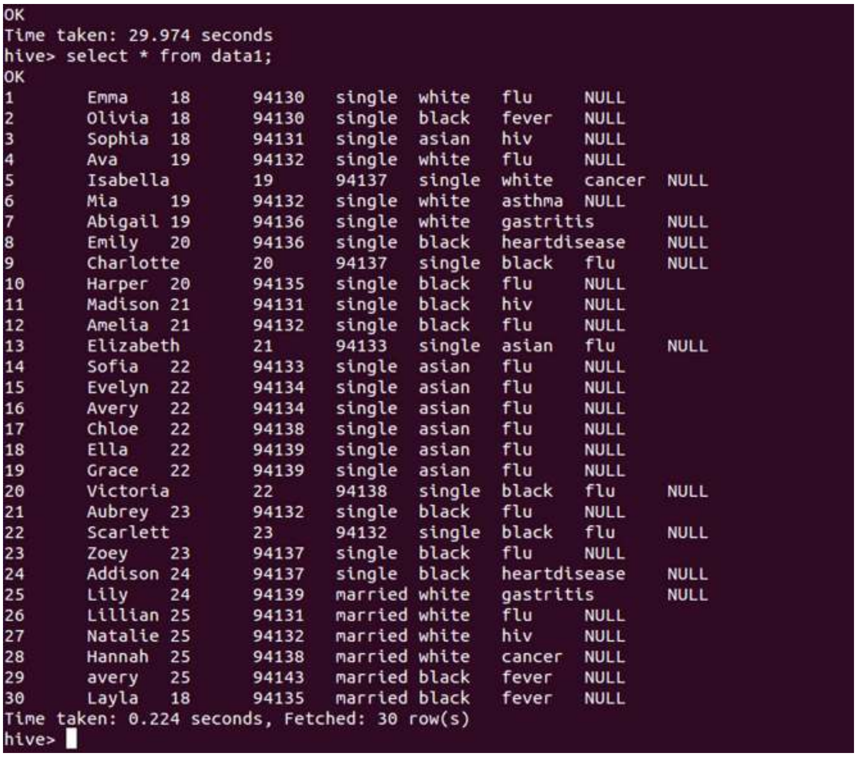
Fig. 4 After sensitivity.java execution (SV's with SL's)

Figure 9 shows the screenshot of the experimentation which gives a view to know how two level hierarchy can be achieved.

Figure 10 gives a screenshot view of how anonymization is achieved with a key value (3,2).

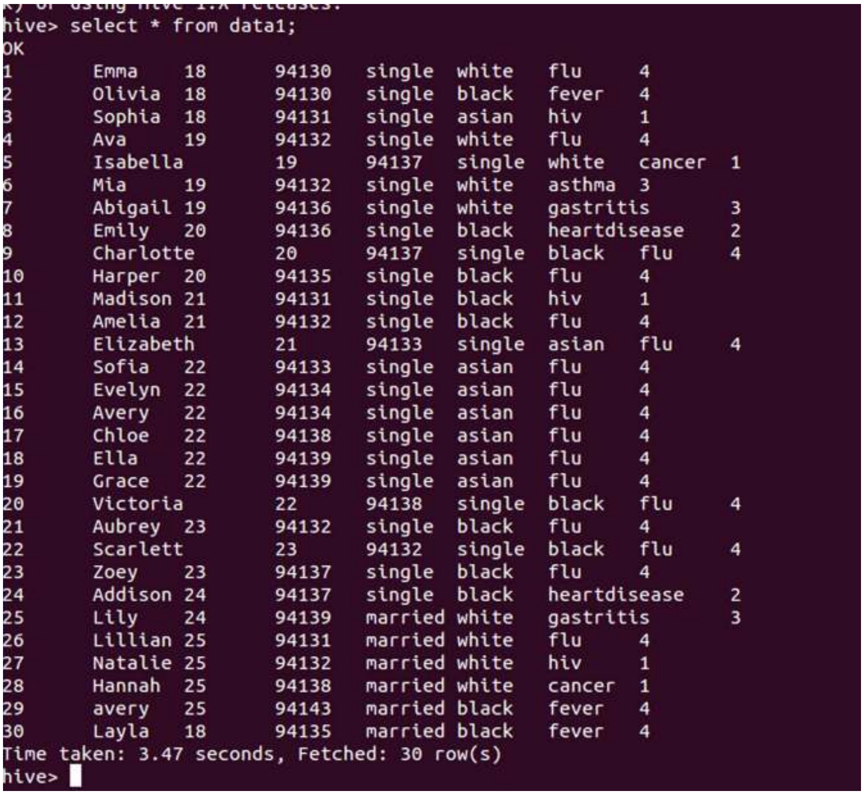
Experimental results

As shown in Figs. 11 and 12, we compare information loss with number of records and increasing key value respectively. Figure 13 shows the variation of information loss with increase in cluster size. Generally in Big Data these are more chances to have large clusters, but with our approach we can get less information loss with large clusters. Figure 14



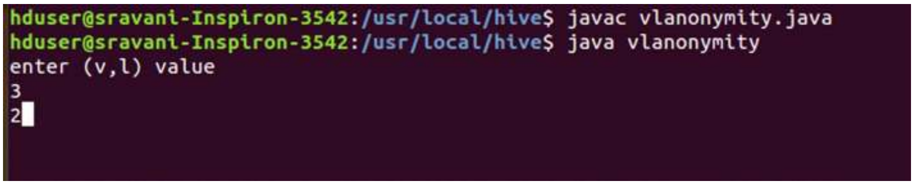
```
OK
Time taken: 29.974 seconds
hive> select * from data1;
OK
1       Emma    18      94130   single  white   flu     NULL
2       Olivia  18      94130   single  black   fever   NULL
3       Sophia  18      94131   single  asian   hiv     NULL
4       Ava     19      94132   single  white   flu     NULL
5       Isabella        19      94137   single  white   cancer  NULL
6       Mia     19      94132   single  white   asthma  NULL
7       Abigail 19      94136   single  white   gastritis       NULL
8       Emily   20      94136   single  black   heartdisease    NULL
9       Charlotte       20      94137   single  black   flu     NULL
10      Harper  20      94135   single  black   flu     NULL
11      Madison 21      94131   single  black   hiv     NULL
12      Amelia  21      94132   single  black   flu     NULL
13      Elizabeth       21      94133   single  asian   flu     NULL
14      Sofia   22      94133   single  asian   flu     NULL
15      Evelyn  22      94134   single  asian   flu     NULL
16      Avery   22      94134   single  asian   flu     NULL
17      Chloe   22      94138   single  asian   flu     NULL
18      Ella    22      94139   single  asian   flu     NULL
19      Grace   22      94139   single  asian   flu     NULL
20      Victoria        22      94138   single  black   flu     NULL
21      Aubrey  23      94132   single  black   flu     NULL
22      Scarlett        23      94132   single  black   flu     NULL
23      Zoey    23      94137   single  black   flu     NULL
24      Addison 24      94137   single  black   heartdisease    NULL
25      Lily    24      94139   married white   gastritis       NULL
26      Lillian 25      94131   married white   flu     NULL
27      Natalie 25      94132   married white   hiv     NULL
28      Hannah  25      94138   married white   cancer  NULL
29      avery   25      94143   married black   fever   NULL
30      Layla   18      94135   married black   fever   NULL
Time taken: 0.224 seconds, Fetched: 30 row(s)
hive> █
```

Fig. 5 Dataset before sensitivity calculation



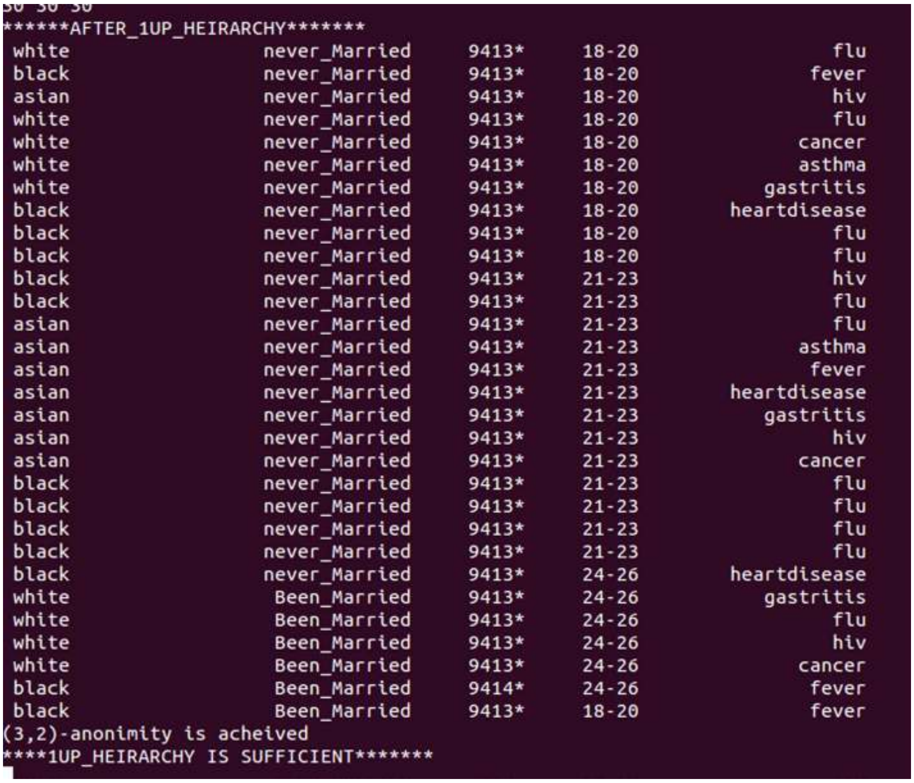
```
k) or using hive 1.X releases.
hive> select * from data1;
OK
1       Emma    18      94130   single  white   flu     4
2       Olivia  18      94130   single  black   fever   4
3       Sophia  18      94131   single  asian   hiv     1
4       Ava     19      94132   single  white   flu     4
5       Isabella        19      94137   single  white   cancer  1
6       Mia     19      94132   single  white   asthma  3
7       Abigail 19      94136   single  white   gastritis       3
8       Emily   20      94136   single  black   heartdisease    2
9       Charlotte       20      94137   single  black   flu     4
10      Harper  20      94135   single  black   flu     4
11      Madison 21      94131   single  black   hiv     1
12      Amelia  21      94132   single  black   flu     4
13      Elizabeth       21      94133   single  asian   flu     4
14      Sofia   22      94133   single  asian   flu     4
15      Evelyn  22      94134   single  asian   flu     4
16      Avery   22      94134   single  asian   flu     4
17      Chloe   22      94138   single  asian   flu     4
18      Ella    22      94139   single  asian   flu     4
19      Grace   22      94139   single  asian   flu     4
20      Victoria        22      94138   single  black   flu     4
21      Aubrey  23      94132   single  black   flu     4
22      Scarlett        23      94132   single  black   flu     4
23      Zoey    23      94137   single  black   flu     4
24      Addison 24      94137   single  black   heartdisease    2
25      Lily    24      94139   married white   gastritis       3
26      Lillian 25      94131   married white   flu     4
27      Natalie 25      94132   married white   hiv     1
28      Hannah  25      94138   married white   cancer  1
29      avery   25      94143   married black   fever   4
30      Layla   18      94135   married black   fever   4
Time taken: 3.47 seconds, Fetched: 30 row(s)
hive> █
```

Fig. 6 Dataset after sensitivity calculation



```
hduser@sravani-Inspiron-3542:/usr/local/hive$ javac vlanonymity.java
hduser@sravani-Inspiron-3542:/usr/local/hive$ java vlanonymity
enter (v,l) value
3
2
```

Fig. 7 Accepting key value through vlanonymity.java



```
30 30 30
******AFTER_1UP_HEIRARCHY*******
 white                never_Married     9413*     18-20              flu
 black                never_Married     9413*     18-20            fever
 asian                never_Married     9413*     18-20              hiv
 white                never_Married     9413*     18-20              flu
 white                never_Married     9413*     18-20           cancer
 white                never_Married     9413*     18-20           asthma
 white                never_Married     9413*     18-20        gastritis
 black                never_Married     9413*     18-20     heartdisease
 black                never_Married     9413*     18-20              flu
 black                never_Married     9413*     18-20              flu
 black                never_Married     9413*     21-23              hiv
 black                never_Married     9413*     21-23              flu
 asian                never_Married     9413*     21-23              flu
 asian                never_Married     9413*     21-23           asthma
 asian                never_Married     9413*     21-23            fever
 asian                never_Married     9413*     21-23     heartdisease
 asian                never_Married     9413*     21-23        gastritis
 asian                never_Married     9413*     21-23              hiv
 asian                never_Married     9413*     21-23           cancer
 black                never_Married     9413*     21-23              flu
 black                never_Married     9413*     21-23              flu
 black                never_Married     9413*     21-23              flu
 black                never_Married     9413*     21-23              flu
 black                never_Married     9413*     24-26     heartdisease
 white                 Been_Married     9413*     24-26        gastritis
 white                 Been_Married     9413*     24-26              flu
 white                 Been_Married     9413*     24-26              hiv
 white                 Been_Married     9413*     24-26           cancer
 black                 Been_Married     9414*     24-26            fever
 black                 Been_Married     9413*     18-20            fever
(3,2)-anonimity is acheived
****1UP_HEIRARCHY IS SUFFICIENT*******
```

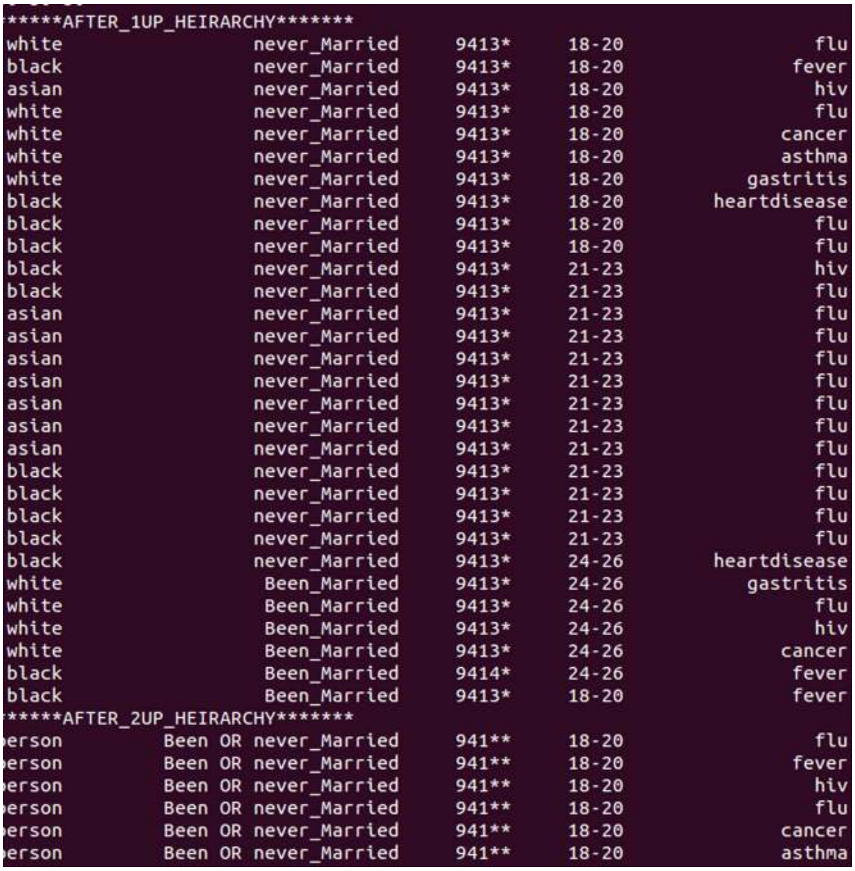
Fig. 8 Anonymization with 1-up hierarchy

shows the performance comparison with Hive versus traditional RDBMS, where Hive can work efficiently with large amount of records. Our experimental results as follows.

Discussion

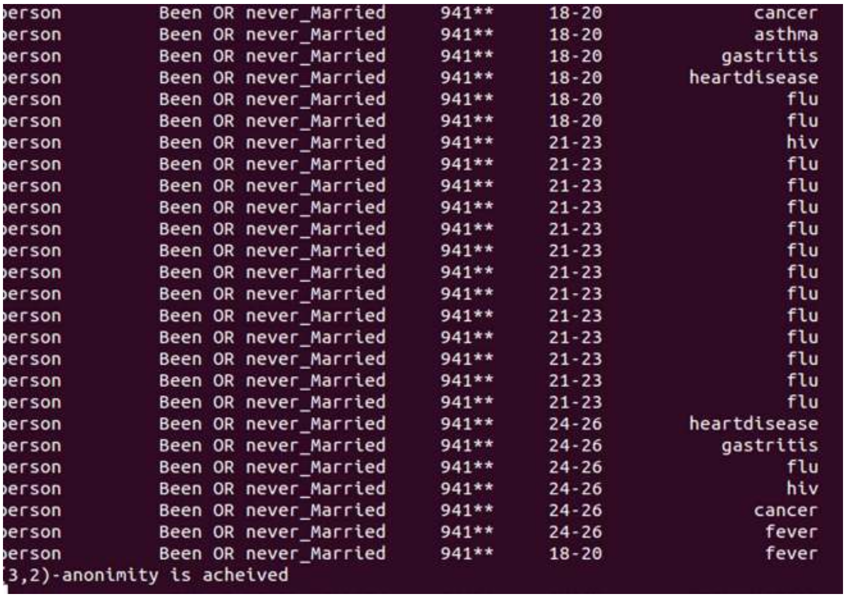
In this paper Privacy preserving data publication was developed by using (v,l)-anonymity with the flavours of clustering and bottom-up generalization.

For experimentation we used Hadoop 2.7.1 with Hive installed on top of it. Traditional systems may not efficiently deals with this huge amount of data and may leads to the systems run very slow. But hive is an environment where we can handle Big Data very



```
*****AFTER_1UP_HEIRARCHY*******
white                  never_Married     9413*     18-20            flu
black                  never_Married     9413*     18-20          fever
asian                  never_Married     9413*     18-20            hiv
white                  never_Married     9413*     18-20            flu
white                  never_Married     9413*     18-20         cancer
white                  never_Married     9413*     18-20         asthma
white                  never_Married     9413*     18-20      gastritis
black                  never_Married     9413*     18-20   heartdisease
black                  never_Married     9413*     18-20            flu
black                  never_Married     9413*     18-20            flu
black                  never_Married     9413*     21-23            hiv
black                  never_Married     9413*     21-23            flu
asian                  never_Married     9413*     21-23            flu
asian                  never_Married     9413*     21-23            flu
asian                  never_Married     9413*     21-23            flu
asian                  never_Married     9413*     21-23            flu
asian                  never_Married     9413*     21-23            flu
asian                  never_Married     9413*     21-23            flu
asian                  never_Married     9413*     21-23            flu
black                  never_Married     9413*     21-23            flu
black                  never_Married     9413*     21-23            flu
black                  never_Married     9413*     21-23            flu
black                  never_Married     9413*     21-23            flu
black                  never_Married     9413*     24-26   heartdisease
white                   Been_Married     9413*     24-26      gastritis
white                   Been_Married     9413*     24-26            flu
white                   Been_Married     9413*     24-26            hiv
white                   Been_Married     9413*     24-26         cancer
black                   Been_Married     9414*     24-26          fever
black                   Been_Married     9413*     18-20          fever
*****AFTER_2UP_HEIRARCHY*******
erson         Been OR never_Married     941**     18-20            flu
erson         Been OR never_Married     941**     18-20          fever
erson         Been OR never_Married     941**     18-20            hiv
erson         Been OR never_Married     941**     18-20            flu
erson         Been OR never_Married     941**     18-20         cancer
erson         Been OR never_Married     941**     18-20         asthma
```

Fig. 9 Anonymization with 2-up hierarchy



```
erson         Been OR never_Married     941**     18-20         cancer
erson         Been OR never_Married     941**     18-20         asthma
erson         Been OR never_Married     941**     18-20      gastritis
erson         Been OR never_Married     941**     18-20   heartdisease
erson         Been OR never_Married     941**     18-20            flu
erson         Been OR never_Married     941**     18-20            flu
erson         Been OR never_Married     941**     21-23            hiv
erson         Been OR never_Married     941**     21-23            flu
erson         Been OR never_Married     941**     21-23            flu
erson         Been OR never_Married     941**     21-23            flu
erson         Been OR never_Married     941**     21-23            flu
erson         Been OR never_Married     941**     21-23            flu
erson         Been OR never_Married     941**     21-23            flu
erson         Been OR never_Married     941**     21-23            flu
erson         Been OR never_Married     941**     21-23            flu
erson         Been OR never_Married     941**     21-23            flu
erson         Been OR never_Married     941**     21-23            flu
erson         Been OR never_Married     941**     21-23            flu
erson         Been OR never_Married     941**     21-23            flu
erson         Been OR never_Married     941**     24-26   heartdisease
erson         Been OR never_Married     941**     24-26      gastritis
erson         Been OR never_Married     941**     24-26            flu
erson         Been OR never_Married     941**     24-26            hiv
erson         Been OR never_Married     941**     24-26         cancer
erson         Been OR never_Married     941**     24-26          fever
erson         Been OR never_Married     941**     18-20          fever
(3,2)-anonimity is acheived
```

Fig. 10 Two level of anonymization with key value (3,2)

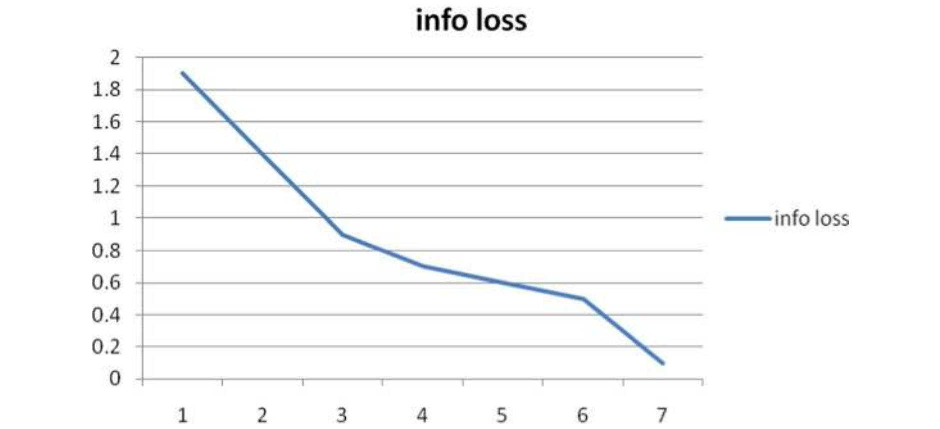


Fig. 11 Info loss with fixed key value (3,2) and variant no. of records

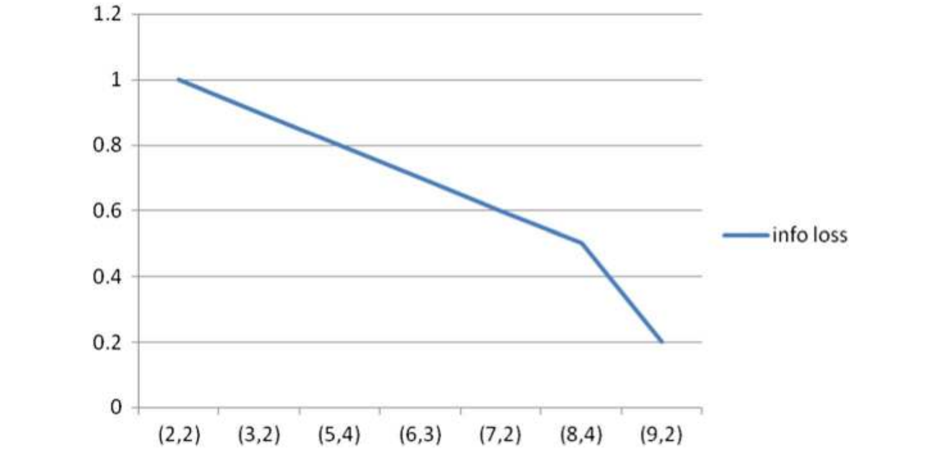


Fig. 12 Info loss with fixed no. of records and variant key values

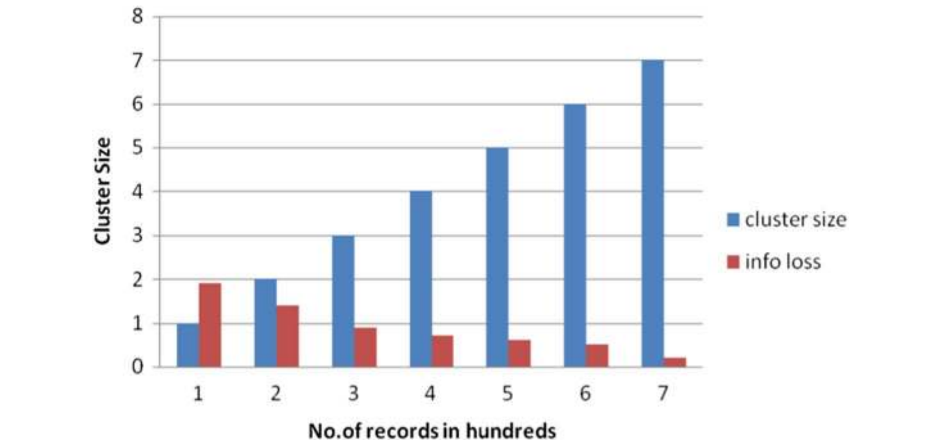


Fig. 13 Variation of informtion loss with increase in cluster size

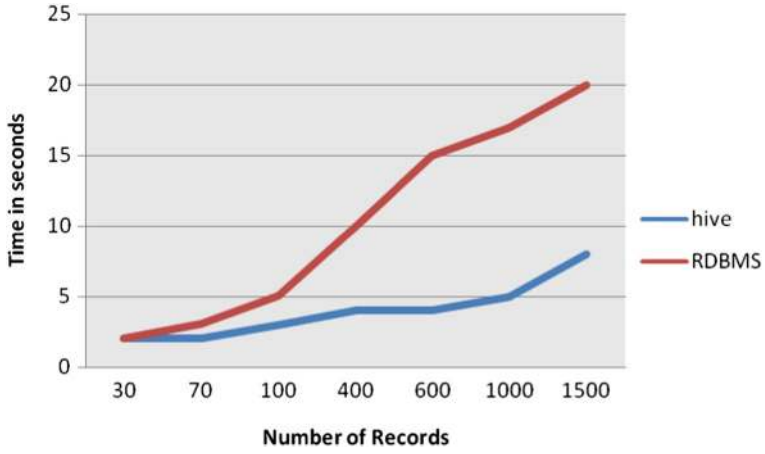


Fig. 14 Performance comparison between Hive and traditional RDBMS

efficiently without any changes to the existing procedures. In this paper we only considered the index value, which is mortality rate. We calculated the sensitivity levels with different index values.

Conclusion and future work

In this paper Privacy preserving data publication was achieved through (v,l)-anonymity by using the flavours of clustering and bottom-up generalization. The proposed privacy model deals with sensitivity vulnerabilities and overcome the disadvantages of existing privacy models. The whole process was done in the context of Big Data, which is the result of increase in the communication means and knowledge sharing. Traditional systems may not efficiently deals with this huge amount of data and may leads to the systems run very slowly. But hive is an environment where we can handle Big Data very efficiently without any changes to the existing procedures. In this paper we only considered one type of index value, which is mortality rate. We can also calculate the sensitivity levels with different index values. The more researches have to be done on the tools to handle Big Data mining, so that the experiments can be carried out in the real world context.

Authors' contributions

PSR performed analysis on all samples, interpreted data, wrote manuscript. SS helped to evaluate and edit the manuscript and acted as corresponding author. Both authors read and approved the final manuscript.

Author details

[1] Computer Science and Engineering, MVGR College of Engineering, Chintalavalasa, India. [2] CSE Department, Adama Science and Technology University, Adama, Ethiopia.

Acknowledgements

I have great pleasure in expressing my gratitude to Dr. S. Satyanarayana presently working as professor in Adama Science and Technology University, ETHIOPIA. Dr. K. V. L. Raju Principal, MVGR College of Engineering, for his kind support in providing me an opportunity to do research in this college.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Not applicable.

Consent for publication

Not applicable.

Ethics approval and consent to participate

Not applicable.

Funding

Not applicable.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 13 December 2017 Accepted: 18 June 2018
Published online: 06 July 2018

References

1. Zhang Z, Gupta BB. Social media security and trustworthiness: overview and new direction. Future Gener Comput Syst. 2016. https://doi.org/10.1016/j.future.2016.10.007.
2. Chen Feng, et al. Data mining for the internet of things: literature review and challenges. Int J Distrib Sens Netw. 2015;11(8):431047.
3. Hu J, Vasilakos AV. Energy big data analytics and security: challenges and opportunities. IEEE Trans Smart Grid. 2016;7:2423–36.
4. Zhang Z, et al. A situational analytic method for user behavior pattern in multimedia social networks. IEEE Trans Big Data. 2017. https://doi.org/10.1109/TBDATA.2017.2657623.
5. Zhang Z, Wang K. A trust model for multimedia social networks. Soc Netw Anal Min. 2013;3:969–79.
6. Qin Y, et al. When things matter: a survey on data-centric internet of things. J Netw Comput Appl. 2016;64:137–53.
7. Dunning LA, Kresman R. Privacy preserving data sharing with anonymous ID assignment. IEEE Trans Inf Forensics Secur. 2013;8(2):402–13.
8. Fong S, Raymond W, Vasilakos AV. Accelerated PSO swarm search feature selection for data stream mining big data. IEEE Trans Serv Comput. 2016;9:33–45.
9. Monreale A, et al. Anonymity preserving sequential pattern mining. Artif Intell Law. 2014;22:141–73.
10. Fung BCM, et al. Privacy-preserving data publishing for cluster analysis. Data Knowl Eng. 2008;68(6):552–75.
11. Lefevre K, Dewitt DJ, Ramakrishnan R. Incognito: efficient full-domain k-anonymity. Int J Uncertain Fuzziness Knowl Based Syst. 2002;10:557–70.
12. Wang K, Yu PS, Chakraborty S. Bottom-up generalization: a data mining solution to privacy protection. Int J Uncertain Fuzziness Knowl Based Syst. 2004;10:557–70.
13. Soria-Comas J, et al. Enhancing data utility in differential privacy via micro aggregation-based k-anonymity. VLDB J. 2014;23:771–94.
14. Machanavajjhala A, Kifer D, Gehrke J, Venkitasubramaniam M. l-diversity: Privacy beyond k-anonymity. In: Proceedings of 2013 IEEE 29th international conference on data engineering, Atlanta; 2013.
15. Anwar M, Greer J. Facilitating trust in privacy-preserving e-learning environments. IEEE Trans Learn Technol. 2015;5(1):62–73.
16. De Cristofaro E, et al. Hummingbird: privacy at the time of Twitter. In: IEEE symposium on security and privacy; 2012. p. 285–99.
17. Das AK. A secure and efficient user anonymity preserving three-factor authentication protocol for large scale distributed wireless sensor network. Wirel Pers Commun. 2015;82:1377–404.
18. Trushina OV, Gabidulin EM. A new method for ensuring anonymity and security in network coding. Probl Inf Transm. 2015;51:75–81.
19. Bayardo RJ, Agrawal R. Data privacy through optimal K-anonymization. Int J Uncertain Fuzziness Knowl Based Syst. 2002;10:557–70.
20. Nikolaenko V, et al. Privacy-preserving ridge regression on hundreds of millions of records. In: Proceedings of the 2013 IEEE symposium on security and privacy; 2013. p. 334–48.
21. Shu X, Yao D, Bertino E. Privacy-preserving detection of sensitive data exposure. IEEE Trans Inf Forensics Secur. 2015;10(5):1092–103.
22. Huang X, et al. Privacy beyond sensitive values. Sci China Inf Sci. 2015;58(7):1–5.
23. Liu Q, Shen H, Sang Y. Privacy-preserving data publishing for multiple numerical sensitive attributes. IEEE Comput. 2015;20(3):246–54.
24. Sweeney L. K-anonymity a model for protecting privacy. Int J Uncertain Fuzziness Knowl Based Syst. 2002;10:557–70.
25. Bredereck R, et al. The effect of homogeneity on the computational complexity of combinatorial data anonymization. Data Min Knowl Discov. 2014;28:65–91.
26. Tsai CW, et al. Big data analytics: a survey. J Big Data. 2015;02(21):21.
27. Zhang Y, et al. Parallel processing systems for big data: a survey. Proc IEEE. 2016;104:2114–36.