Chapter 10

Data Privacy for Big Data Publishing Using Newly Enhanced PASS Data Mining Mechanism

Priyank Jain, Manasi Gyanchandani and Nilay Khare

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.77033

Abstract

Anonymization is one of the main techniques that is being used in recent times to prevent privacy breaches on the published data; one such anonymization technique is k-anonymization technique. The anonymization is a parametric anonymization technique used for data anonymization. The aim of the k-anonymization is to generalize the tuples in a way that it cannot be identified using quasi-identifiers. In the past few years, we saw a tremendous growth in data that ultimately led to the concept of the big data. The growth in data made anonymization using conventional processing methods inefficient. To make the anonymization more efficient, we used the proposed PASS mechanism in Hadoop framework to reduce the processing time of anonymization. In this work, we have divided the whole program into the map and reduce part. Moreover, the data types used in Hadoop provide better serialization and transport of data. We performed our experiments on the large dataset. The results proved the best efficiency of our implementation.

Keywords: big data, big data privacy and security, data mining, attribute disclosure, PASS, information loss, membership disclosure

1. Introduction

1.1. What is big data?

"Big data" is data whose scale, diversity, and complexity require new architecture, techniques, algorithms, and analytics to manage it and extract value and hidden knowledge from it.

1.2. Big data significance in industry and challenges

While understanding the estimation of big information keeps on residual a test, other down to practical challenges including financing and rate of return and aptitudes keep on remaining at the front line for various distinctive ventures that are embracing huge information. All things considered, a Gartner survey for 2015 demonstrates that over 75% of organizations are putting or are intending to put resources into enormous information in the following 2 years. These discoveries speak to a critical increment from a comparable study done in 2012, which showed that 58% of organizations put or were wanting to put resources into enormous information inside the following 2 years [1].

By and large, most associations have a few objectives for receiving enormous information ventures. While the essential objective of most associations is to upgrade client encounter, different objectives incorporate cost diminishment, better focused on promoting and making existing procedures more effective. Lately, information ruptures have additionally made upgraded security a critical objective that has huge information.

1.3. Data stream

Big data associated with the time stamp is called big data stream [2].

Examples of data streams:

1. Sensor data
2. Call center records
3. Clickstreams
4. Healthcare data
5. Constraints associated with data streams

Privacy protection: i.e., the data streams are extracted from various sources which consist of many individuals' information; hence, the sensitive data of any individuals must not be leaked.

Computer security: Access control and verification guarantee that opportune individual has a right expert to the correct protest at the perfect time and the ideal place. That is not what we need here. A general precept of information security is to discharge all the data as much as the personalities of the subjects (individuals) are ensured.

Real-time processing: Since the data is not static in nature, real-time processing is required, and at present, not many algorithms are there to process the dynamic data.

1.4. What is MapReduce?

MapReduce, as shown in **Figure 1**, is a preparing method and a program that demonstrates for circulated figuring in light of java. The structure deals with every one of the points of interest

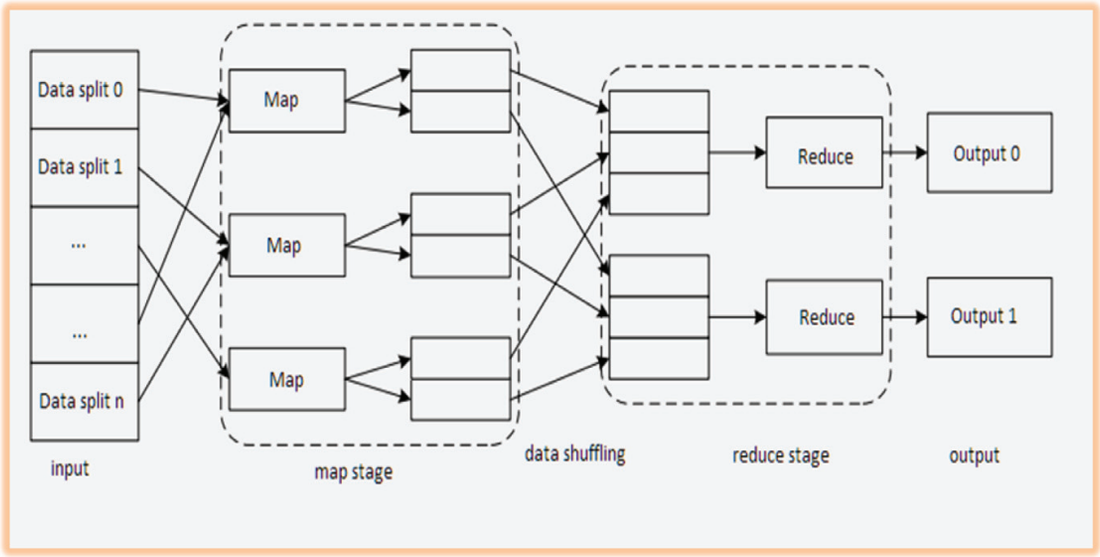

Figure 1. Internal working of Map Reduce.

of information passing, for example, issuing errands, confirming assignment culmination, and duplicating information around the group between the hubs [1]:

- Most of the registering happens on hubs with information on neighborhood circles that lessens the system activity.
- After consummation of the given errands, the bunch gathers and diminishes the information to shape a suitable outcome and sends it back to the Hadoop server.

2. Anonymization

2.1. Anonymization

Generally, the fundamental theme data anonymization [2–5] is the use of one or more techniques designed to make it impossible or at least more difficult to identify a particular individual from stored data related to them. Purposes of data anonymization are:

1. To prevent the privacy of individuals who shared data for various surveys
2. To implement effective techniques to prevent a security breach

It is privacy preservation techniques used for static data. Techniques implemented in anonymization are:

1. Encryption
2. Hashing
3. Generalization

4. Suppression of data
5. Destroy data quality
6. Adding mathematical noise

2.2. K-anonymity

A release of data is said to have the *k*-anonymity property if the information for each person contained in the release cannot be distinguished from a least k-1 individuals whose information also appear in the release. For example, if you try to identify a person from a release dataset but you only have information of his/her birth date and gender. There are k people that meet the requirement. This is k-anonymity [6, 7].

2.2.1. Classification of attributes

Key attribute is name, address, and cell phone, which can uniquely identify an individual directly. It is always removed before release.

Quasi-identifier is a zip code, birth date, and gender, a set of attributes that can be potentially linked with external information to re-identify entities. Eighty-seven percent of the population in the USA can be uniquely identified based on these attributes, according to the census summary data in 1991. There are two tables shown below: **Table 1** is hospital dataset and **Table 2** is voter data.

DOB	Sex	Zip code	Disease
1/21/1976	M	65715	Heart disease
4/13/1986	F	65715	Hepatitis
2/28/1976	M	65703	Bronchitis
1/21/1976	M	65703	Broken arm
4/13/1986	F	65706	Flu
2/28/1976	F	65706	Hang nail

Table 1. Medical dataset.

Name	DOB	Sex	Zip code
Andre	1/21/1976	Male	53715
Beth	1/10/1981	Female	55410
carol	10/1/1944	Female	90210
Dan	2/21/1984	Male	02174
Ellen	4/19/1972	Female	02237

Table 2. Voter dataset.

From above tables, we can conclude that Andre has heart disease; here the heart disease is the sensitive attribute. It is known as linking attack by combining two different tables. The solution is to consider all of the released tables before releasing the new one and trying to avoid linking. And k-anonymity does not provide privacy if sensitive values in an equivalence class lack diversity [8, 9].

3. Related work

3.1. FANNST algorithm

3.1.1. Algorithm

When the numbers of tuples in the processing window reach μ, one round of the clustering algorithm is started to slide again in order to accumulate more tuples in each round [10].

Parameters used in the algorithm are k, u, d:

k defines the parameter for cluster anonymization.

d defines the number of clusters which can be used later.

u defines the processing window size.

3.1.2. Drawback

The main drawback of FANNST is that some tuples may remain in the system for more than allowable time constraint. In addition, the time and space complexity of the algorithm is O(S*S) and not efficient for a data streaming algorithm. Another weakness of FANNST is that it does not support categorical data.

3.2. FADS algorithm

The algorithm considers a set as a buffer and saves at most δ tuples in it [11, 12]. Also, another set (setkc) is considered to hold the newly created cluster for later reuse. Each k-anonymized cluster will be remained in setkc up to the reuse constraint Tkc, and after that, the cluster is removed.

3.2.1. Drawbacks

The main drawback of the FADS is that the algorithm does not check the remaining time of tuples that hold in the buffer in each round and give their result when they might be considered to have expired. The other important weakness of FADS is that it is not parallel and cannot handle a large number of data streams in tolerable time.

4. Terminology of proposed algorithm

4.1. Data stream

A sequence of tuples is defined as <sn>n∈N where N is the natural number set. The kth term of <sn> is order pair (t, tk) where k is a number and tk is a tuple.

A data stream S is a potentially infinite sequence of tuples, depicted by <t_i>, where all tuples t_i follow the schema t_i = <ID,a_1,a_m,q_1,q_n,TS>. ID is an identifier attribute; q_1 to q_n are quasi-identifiers, and TS is the time stamp.

4.2. Cluster

The cluster is a set of tuples in a stream [12]. Suppose that PS is a set of tuples in stream cluster C which can be defined as follow:

C = {t | t belongsPs}

4.2.1. K-anonymized cluster

If a cluster C is built from the data stream and the number of the unique tuple in the cluster is greater than k, the cluster is called a k-anonymized cluster.

4.2.2. Generalization

Generalization is a function that maps a cluster into a tuple. More formally, generalization function G is defined as G: PowerSet(TUPLE) → TUPLE where TUPLE is the set of all possible tuples.

4.2.3. Numerical value generalization

Numerical values are generalized in between maximum and minimum value, i.e., they are generalized in their domain.

4.2.4. Categorical value generalization

Categorical values are generalized to their lowest common ancestors as shown in **Figure 2**.

4.2.5. Example of above two types of generalization

Considering a cluster of three tuples which contains both numerical and categorical values, the tuples contain the name, profession, and age of employees.

C = <“prof.young”, Academic, 43>,

<“Mr.Zhou”, non-Academic, 39>,

<“Prof.Chung”, Academic, 46>.

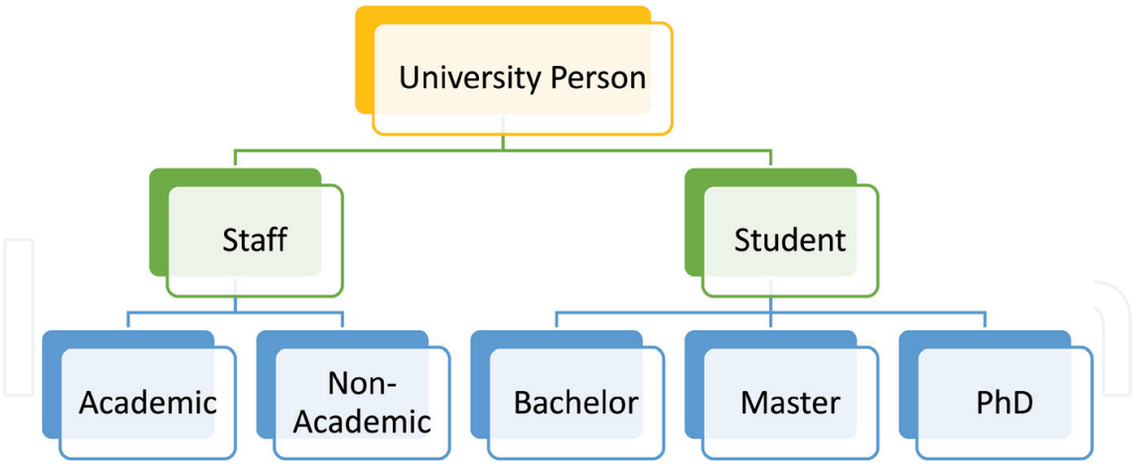

Figure 2. University taxonomy tree.

The above tuple can be generalized as follows: gc = <∗,staff,[39–46]>. Since we do not want to disclose the name, we kept * in the first column; here profession is categorical value, and age is numerical value; age is generalized as [max, min], and profession is generalized to lowest common ancestor of academic and nonacademic.

4.2.6. Distance

Distance is used to calculate the similarity or dissimilarity between two tuples. This function is the heart of the clustering. Generally, clustering is done based on distance calculation; the tuples with the closest distance are placed the same cluster.

4.2.6.1. Types of distances

4.2.6.1.1. The distance between the numerical values

Let v_1,v_2 be 2 numerical values.

The distance between v_1,v_2 = d(v_1,v_2) = $|v_1 - v_2|/|D|$

where D is the domain of the values.

4.2.6.1.2. The distance between two categorical values

If all the categorical values are arranged in the form of a tree where the root is the most generalized value of all the values and lowest most level containing more specialized values of the categorical values, e.g., of a categorical tree as shown in **Figure 3** Country taxonomy tree and **Figure 4** Occupation taxonomy tree.

Distance between two categorical values v_1,v_2 = d(v_1,v_2) = (height of the subtree roots at lowest common ancestor of (v_1,v_2))/(height of tree):

For example, distance between India and Egypt (considering the tree from the above picture).

=Height of subrooted tree of a lowest common ancestor of India and Egypt/height of the tree.

=Height of the tree with east as root/height of tree = 1/3 = 0.33.

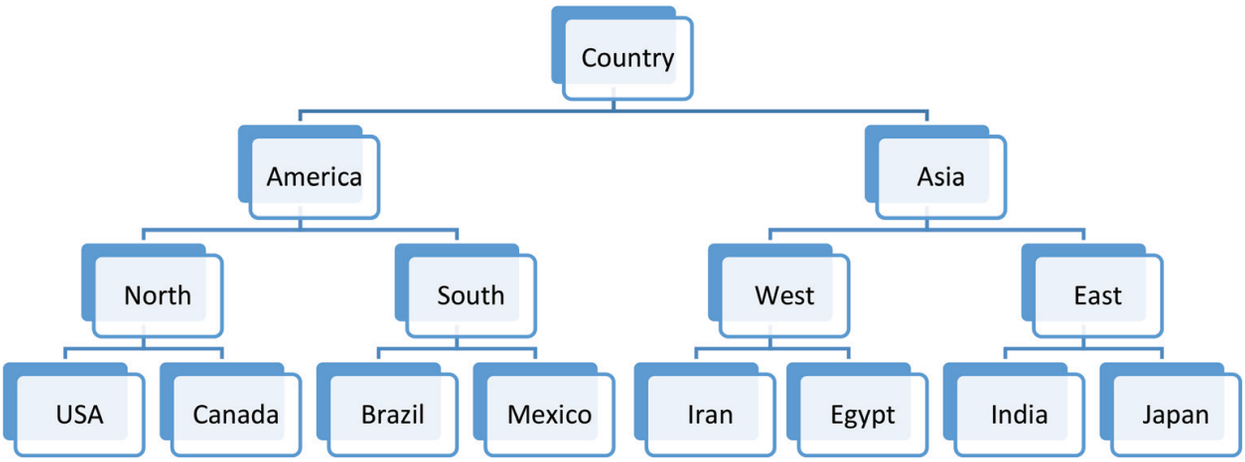

Figure 3. Country taxonomy tree.

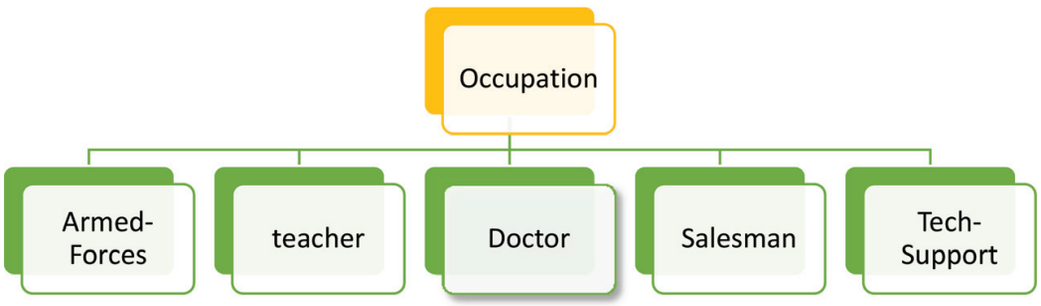

Figure 4. Occupation taxonomy tree.

4.2.6.1.3. The distance between two tuples

Distance between two tuples t = {N1,…,Nm, C1,…,Cn} is the quasi-identifier of table T, where N_i (i = 1,…,m) is an attribute with a numeric domain and Cj(j = 1,…,n) is an attribute with a categorical domain.

The distance d(r1,r2) (i.e., the distance between two tuples r1, r2) is defined as:

d(r1,r2) = **sum of distances between numerical attributes of two tuples + sum of distances between categorical attributes of two tuples**.

Information loss: generalization leads to information loss, but we have to group clusters in such a way that the information loss is minimum.

Information loss of a single cluster is calculated as:

Total information loss = sum of information loss of all the clusters.

Information loss of the cluster = info loss of all the tuples in the cluster.

Information loss of the tuple = information loss of all the attributes (categorical attributes and numerical attributes).

Information loss of numerical attribute = (value of attribute)/(domain of the attribute).

Information loss of categorical attribute = (height of the tree rooted with categorical attribute)/(height of categorical attribute tree) where h is the height of the tree and k is the height of the tree rooted at the required categorical attribute.

5. Proposed PASS algorithm

5.1. Details of the PASS algorithm

S = total number of tuples in the dataset.

K = anonymization parameter.

$ = number of tuples to be read before processing.

SetTp = set of $ tuples.

SetKc = set of all unique generalized sets.

Snew = set of K tuples.

Gs = generalized set of Snew.

The algorithm reads $ tuples continuously and inserts them into the SetTp. At First, for each tuple in SetTp procedure finds t's K-1 nearest tuples in SetTp, with the help of tuple t and its K-1 nearest tuples, generate a new set called as Snew and generalize it into Gs. Then a set with minimum information loss (Sk-best) that covers tuple t is chosen from SetKc if Sk-best exists and has smaller information loss than Gs; then tuple t is published Sk-best generalization.

If tuple t does not match with any set of SetKc which has less information loss compared to Gs, then tuple t is published with Snew generalization, i.e., Gs. Then Gs is inserted in SetKc.

In the following, a simple example is illustrated for better understanding. **Table 3** is a portion of a university person data stream, in which quasi-identifiers are age and job. Also $ and K are assumed as $ = 3 and K = 2. Suppose that in thread n, the value of variables is as follows:

Pid	Age	University person
Id1	22	Bachelor
Id2	24	Master
Id3	37	Nonacademic
.	.	.
.	.	.
.	.	.
Idn	45	Academic
Idn + 1	26	Nonacademic
Idn + 2	39	PhD

Table 3. University person.

Pid	Age	University person
Id1	[22–24]	Student
Id2	[22–24]	Student
Id3	[15–95]	University person
.	.	.
.	.	.
.	.	.
Idn	[44–46]	Staff
Idn + 1	[26–39]	University person
Idn + 2	[26–39]	University person

Table 4. Two anonymized university persons.

In this stage, information loss of Sk-best is compared with Gs information loss. As the information loss of Sk-best is less than Gs, a tuple with idn is published with Sk-best generalization. **Table 4** represents Two anonymized university persons.

- SetTp = {(<idn,45,academic>, <idn + 1,26,Non − academic>,<idn + 2,39,PhD>)}
- SetKc = {(([22–24],university), ([31–39],staff),([44–46],staff))}
- Snew = (<idn,45,academic>,<idn + 2,26,non-academic>)
- Gs = ([26–45],staff)
- Sk-best = ([44–46],staff)

5.2. Proposed PASS algorithm

Big data Anonymization (S,K,$)

{

while S!=0 **do**

Read $ tuples and insert them into

SetTp.

For each tuple t **do**

1. Select K-1 unique tuples

which are closest to t among

the tuples in SetTp and insert them into

set Snew.

2. Generalize Snew into Gs.

For each set which covers t **do**

Calculate the information loss

End for

3. Select a set which includes less information loss

Call the set as Sk-best

4. **If** (Sk-best exist and Sk-best generate less information loss

Than Gs) **then**

Publish t with Sk-best generalization

Else

Publish t with Gs and insert Gs in set Kc

End if

End for

End while

}

6. Result and discussion

6.1. Experiment environment

This experiment is performed on the system having Intel i5 processor with the processing power of 2.2 GHz and main memory of 4.0 GB using Linux platform. The algorithm is implemented in Java and executed with the help of Hadoop MapReduce framework.

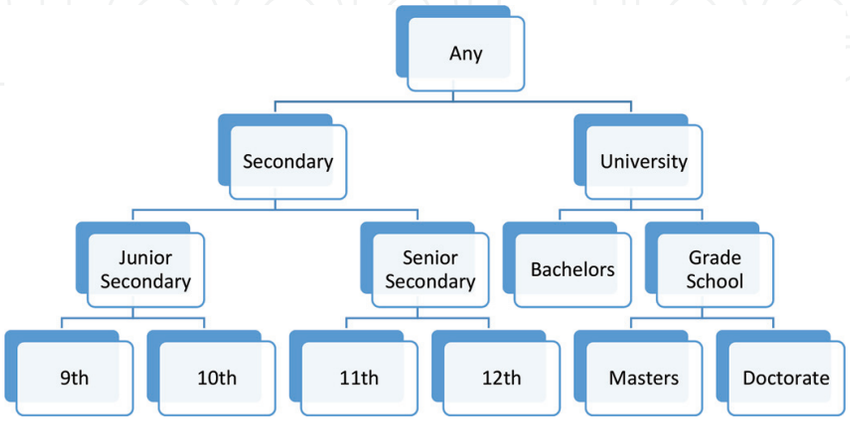

Figure 5. Taxonomy tree.

6.2. Dataset description

In this experiment, we evaluated the performance of the proposed algorithm on the adult dataset from UCI [13]. The dataset was widely used for the privacy-preserving purpose. The taxonomy tree is defined as per **Figure 5**. The sensitive attribute in the dataset is age (numerical) and profession (categorical).

6.3. Results and discussions

The total number of records in the dataset used for the experiment purpose is 32,599 tuples. The efficiency of proposed algorithm is verified by parameter information loss. The average information loss of the proposed PASS algorithm, FADS and FAST, is presented in **Figure 6**. The proposed PASS algorithm publishes data with less information loss, because the SetKc in the proposed approach as shown in **Figure 7** has more entities so that the data tuple has more

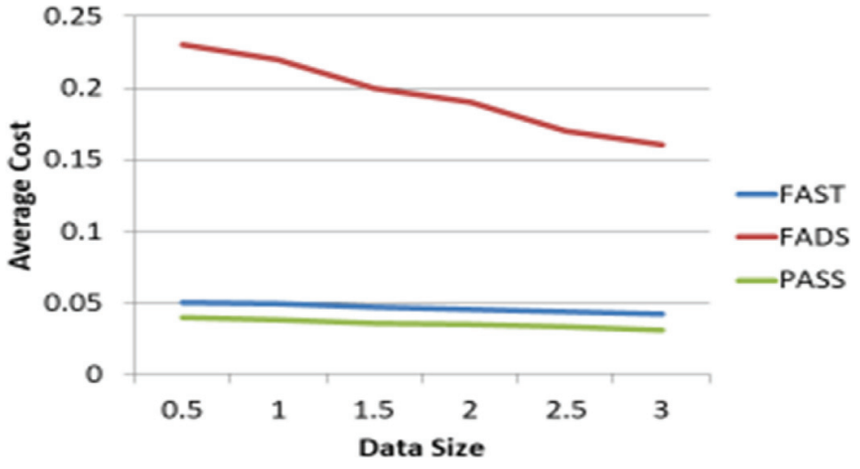

Figure 6. Information loss in FAST and FADS algorithms.

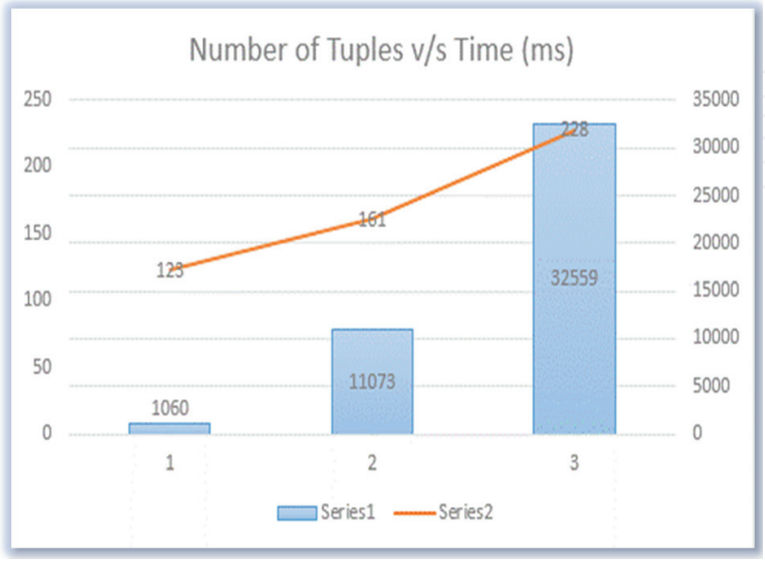

Figure 7. Number of tuples vs. running time.

http://dx.doi.org/10.5772/intechopen.77033

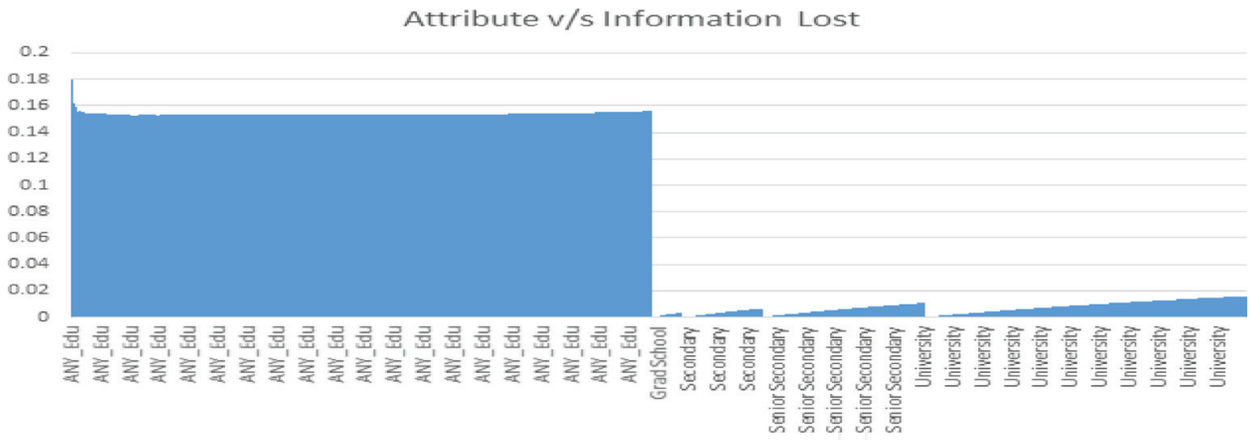

Figure 8. Attribute vs. information loss.

options to select, and this decreases the information loss as shown in **Figure 8**, and hence the results of an algorithm show improvement. The average execution time drastically decreases as MapReduce-based newly enhanced PASS mechanism is used.

7. Conclusion

All the algorithms which are present for data stream processing are not capable of processing big data, i.e., data with high capacity and volume. The data which is processed using data anonymization (nonparallel) algorithms use old languages (JAVA, SQL) and old techniques, which are not very effective means because they take a lot of time for computation and sometimes provide tuples, which are expired; this lead to loss of accuracy as well as loss of privacy which is very dangerous. Static algorithms need all the computations to be performed on a single node due to which the data and the processing requirements are very high and the computers used are prone to failure which is very expensive to recover.

In this paper, we have proposed PASS algorithm, which uses Hadoop framework to process the data. Using Hadoop, the computer's resources are used to the maximum extent by which time required for computation is reduced which in turn prevents the publishing of expired tuples. Other advantages of this algorithm are that computations can be performed on nodes which have less computation and less storage capacity than that of computers which perform nonparallel data processing. The proposed PASS algorithm publishes data with less information loss. Using Hadoop, the failures in both data and processors can be recovered. These features drastically reduce the maintenance cost and the initial setup cost.

Author details

Priyank Jain*, Manasi Gyanchandani and Nilay Khare

*Address all correspondence to: priyankjain1984@gmail.com

Department of Computer Science and Engineering, MANIT, Bhopal, India

References

[1] Aggarwal CC, Han J, Wang J, Yu PS. A framework for clustering evolving data streams. In: Proceedings of the 29th International Conference on Very Large Data Bases-Vol. 29. VLDB Endowment; 2003. pp. 81-92

[2] Cao J, Carminati B, Ferrari E, Tan K-L. Castle: Continuously anonymizing data streams. IEEE Transactions on Dependable and Secure Computing. 2011;**8**(3):337-352

[3] Dwork C. Differential privacy. In: Bugliesi M, Preneel B, Sassone V, Wegener I, editors. ICALP Lecture Notes in Computer Science. Springer, 2006;**4052**(2):1-12

[4] Li F, Sun J, Papadimitriou S, Mihaila GA, Stanoi I. Hiding in the Crowd: Privacy Preservation on Evolving Streams through Correlation Tracking. In: ICDE Istanbul Turkey, 2007. p. 2

[5] Li N, Li T, Venkatasubramanian S. T-Closeness: Privacy beyond K-Anonymity and L-Diversity. In: ICDE Istanbul Turkey, 2007, p. 106-115

[6] Fung B, Wang K, Chen R, Yu PS. Privacy-preserving data publishing: A survey of recent developments. ACM Computing Surveys (CSUR). 2010;**42**(4):14

[7] Kim S, Sung MK, Chung YD. A framework to preserve the privacy of electronic health data streams. Journal of Biomedical Informatics. 2014;**50**:95-110

[8] Fung BC, Wang K, Yu PS. Top-down specialization for information and privacy preservation. In: Proceedings of the 21st International Conference on Data Engineering (ICDE 2005). IEEE; 2005. pp. 205-216

[9] Machanavajjhala A, Kifer D, Gehrke J, Venkitasubramaniam M. L-diversity: Privacy beyond k-anonymity. ACM Transactions on Knowledge Discovery from Data (TKDD). March 2007;**1**(1):3. DOI: 10.1145/1217299.1217302

[10] Zakerzadeh H, Osborn SL. FAST: Fast Anonymizing Algorithm for Numerical Streaming Data. Proceedings of the 5th International Workshop on Data Privacy Management, and 3rd International Conference on Autonomous Spontaneous Security. Athens, Greece; September 23, 2010

[11] Guo K, Zhang Q. Fast clustering-based anonymization approaches with time constraints for data streams. Knowledge-Based Systems. 2013;**46**:95-108

[12] Guo K, Zhang Q. Fast clustering-based anonymization approaches with time constraints for data streams. Knowledge-Based Systems. July 2013;**46**:95-108. DOI: 10.1016/j.knosys.2013.03.007

[13] Newman CBD, Merz C. UCI Repository of machine learning databases. Technical report, University of California, Irvine, Department of Information and Computer Sciences. 1998