International Journal of Engineering and Techniques - Volume 2 Issue 3, May – June 2016

RESEARCH ARTICLE OPEN ACCESS

Enhancement of Privacy Preserving Technique using Slicing with Entity Resolution

S.Renuka Devi[1], A.C. Sumathi[2]

1 PG Scholar, Dept. of CSE, SNS College of Engineering, Coimbatore, Tamil Nadu, India,

2 Associate Professor, Dept. of CSE, SNS College of Engineering, Coimbatore, Tamil Nadu, India,

Abstract:

The various anonymization techniques, called generalization and bucketization, have been designed for providing data privacy and preserving micro data publishing. Recent work shows that generalization loses considerable amount of information on high dimensional data and bucketization did not prevent membership disclosure with clear separation between quasi-identifying attributes and sensitive attributes. The slicing techniques partitions the data both horizontally and vertically is proposed with entity resolution which preserves data utility and membership disclosure protection .Slicing develops an efficient algorithm with the ‘l-diversity requirement. The workload experiments with sensitive attribute ensures that slicing provides better utility than generalization and is more effective than bucketization. As an extension we proposed a technique called overlapped slicing, were the attributes are divided into more than one column and release in each column consists of more attribute correlations.

***Keywords* —**

1 INTRODUCTION

1.1 **DATA Stream Management Systems (DSMS)** is proposed to process transactional data for health monitoring system. Access control mechanisms for data streams prove that only the authorized parts of the stream are available to each user or role. The queries or views of the data stream has to be protected by access control mechanism. If the sensitive information in a data stream is not secured properly, then the isolation of a person can be negotiation even in the presence of access control. The identified privacy preservation techniques of k-anonymity and l-diversity have also been used for privacy protection of data streams. The attribute values in the data stream tuples can be indiscriminate to satisfy the given isolation necessities

1.2 **Attribute data** of generalization establish ambiguity in the uncertainty results for access control mechanism. If the publishing of stream data is delayed then imprecision may be reduced. Since, the delay initiate false negatives in the query results, the tuples convince the query predicate have not been made obtainable to the access control mechanism at the occurrence of query evaluation.

1.3 The fundamental idea of slicing is to split the association cross columns, but to preserve the association within each column. Here the data are divided by horizontally and vertically this reduces the dimensionality of the data and preserves better utility comparatively with generalization and bucketization. Slicing protects utility because it groups highly-correlated attributes together and preserves the association between those attributes. The process of slicing is it breaks the associations between uncorrelated attributes, which are uncommon and thus identifying. When the dataset contains QIs and one SA, bucketization has to split their correlation. Slicing can group some QI attributes with the SA preserving attribute correlations with the sensitive attribute. The workload experiments authenticate that slicing preserves better utility comparatively with the generalization and more successful than bucketization in workloads involving with the sensitive attribute and the row reduction concepts are implemented to improve more privacy.[1][2][3]

2. EXISTING SYSTEM

2.1 Problem Definition:

In the existing system the Anonymization techniques used are generalization and bucketization, have been designed for privacy preserving Micro data publishing. Access control mechanisms for data streams ensure that only the approved parts of the stream are accessible to each user or role. The sensitive information in a data stream is not privacy protected; The well-known privacy preservation techniques of k-anonymity and l-diversity have also been used for privacy protection of data streams. The attribute values in the data stream tuples are generalized to satisfy the given privacy requirements. Attribute data generalization introduces indistinctness in the query results for access control mechanism. This imprecision can be reduced, if the publishing of stream data is Deferred. However, the delay introduces false negatives in the query results if the tuples fulfilling the query predicate have not been made available to the access control mechanism at the instance of query evaluation. Initially, many accessible clustering algorithms (e.g., kmeans) require the calculation of the "centroids". Second, k-medoid method is very vigorous to the existence of outliers (i.e., data points that are very far away from the rest of data points). Third, the sort in whom the data points are examined does not affect the clusters computed from the k-medoid method. In both method called generalization and bucketization, attributes are detached into three categories: (1) identifiers that can uniquely identify an individual, such as Name or Social Security Number; (2) some attributes are Quasi-Identifiers (QI), which the adversary may already know (possibly from other publicly-available databases) and which, when taken together, can potentially identify an individual, e.g., Birth- date, Sex, and Zipcode; (3) some attributes are Sensitive Attributes (SAs), which are unknown to the adversary and are considered sensitive, such as Disease and Salary. [4][5][6]

2.2 Generalization:

Generalization is one of the most commonly used anonymized approaches, which replaces quasi-identifier values to more generalized value that are less- exact but Semantically constant. Then, all quasi-identifier values in a group would be generalized to the entire group extent in the Quasi-ID space. If at least two transactions in a group have dissimilar values in a certain column, then all information about that item in the current group is lost and QID used in this process includes all possible items in the log. Because of high-dimensionality of the quasi-identifier with different possible items in thousands of order, generalization method will cause high information loss and also symbol the data in useless. To improve generalization in well-organized manner, arrange the similar records in the same bucket to avoid loss of information. [15][16][17]

To achieve data analysis on the generalized table, the data analyst has to construct the uniform distribution assumption that in each value with generalized interval/set is equally possible, as no other distribution assumption can be justified. This drastically decreases the data utility of the generalized data. And also because each attribute is generalized disconnectedly, association between different attributes is lost. In order to study attribute correlations on the generalized table, the data analyst has to guess that every possible combination of attribute values is uniformly possible. This is an inbuilt Problem of generalization that avoids successful analysis of attribute association.

2.2.1 Limitation of Generalization

Two main problems of generalization are:

1. Fails on high-dimensional data due to the curse of dimensional
2. Too much information loss due to uniform-distribution.

Microdata Set

id	name	age	sex	zipcode	disease
1012	velu	54	M	47302	flu
1013	mani	60	M	47302	dyspepsia
1014	nilays	60	M	47304	dyspepsia
1015	meenashi	64	F	47304	gastritis
1016	maheshwari	22	F	47906	flu
1017	muthu	33	M	47905	flu
1018	sathya	52	F	47905	bronchitis
1019	velu	54	M	47302	flu
1020	mani	60	M	47302	dyspepsia
1021	nilays	60	M	47304	dyspepsia
1022	meenashi	64	F	47304	gastritis

Figure2 Microdata Set Table

Generalization

AGE	SEX	ZIPCODE	DISEASE
20-52	*	479**	bro
20-52	*	479**	flu
20-52	*	479**	bro
20-52	*	479**	flu
20-52	*	479**	flu
20-52	*	479**	bro
20-52	*	479**	flu
20-52	*	479**	flu
20-52	*	479**	dysp
52-64	*	473**	flu
52-64	*	473**	flu
52-64	*	473**	dys
52-64	*	473**	flu

Figure3 Generalization Table

2.3 Bucketization:

Bucketization is to screen the tuples in T into buckets and then to split the sensitive attribute from the non-sensitive ones at random permuting the sensitive attribute values within each bucket. The sterile data then consists of the buckets with permuted sensitive values. A set of buckets of permuted sensitive attribute values are filled with anonymized data and bucketization technique are used for anonymizing high-dimensional data. However, their approaches explains a clear separation between QIs and SAs and because of the exact values of all QIs are released, membership information is disclosed. At the final process, it returns a set of disjoint buckets and least ℓ distinct impressionable values. Bucketization preserves better data utility than generalization.

2.3.1 Limitations of Bucketization

1. Does not avert membership disclosure.
2. Necessitate a clear partition between QIs and SAs.

3. Split the attribute association between the QIs and the SAs by sorting out the SA from the QI attributes.

Bucketized Table

Age	Sex	Zipcode	Disease
64	F	47304	gastritis
54	M	47302	gastritis
60	M	47302	gastritis
60	M	47304	flu
64	F	47304	flu
54	M	47302	flu
60	M	47302	flu
60	M	47304	flu
64	F	47304	flu
54	M	47302	dyspepsia
60	M	47302	dyspepsia

Figure4 Bucketization Table

2.4 Disadvantages of Existing System:

1. On hand anonymization algorithms are capable of using for column generalization, e.g., Mondrian. The algorithms are used on the sub table containing only attributes in one column to ensure the anonymity requirement
2. Ongoing data analysis (e.g., query answering) methods can be easily used on the sliced data.
3. Present privacy measures for membership disclosure protection include differential privacy and presence. [7]

3. PROPOSED SYSTEM:

We present a technique called slicing, which partitions the data both horizontally and vertically. We explain that slicing protect better data utility than generalization and can be used for membership disclosure protection and also handle high-dimensional data. [8][9][10] We prove that slicing are used for attribute disclosure protection and develop an efficient algorithm for computing the sliced data which obey the ℓ-diversity requirement. Our workload experiments authenticate that slicing preserves improved utility than generalization and is more effective than bucketization in workloads concerning the sensitive attribute. With further enhancement we implement the entity resolution for row reduction to provide better preservation.

3.1 System Model:

The work flow of the slicing and its extension of row reduction is given below is following steps

3.1.1 Functional procedure:-

Step 1: Extract the data set from the database.
Step 2:Anonymity process divides the records into two.
Step 3: Interchange the sensitive values.
Step4:Multiset values generated and displayed.
Step 5: Attributes are combined and secure data Displayed.

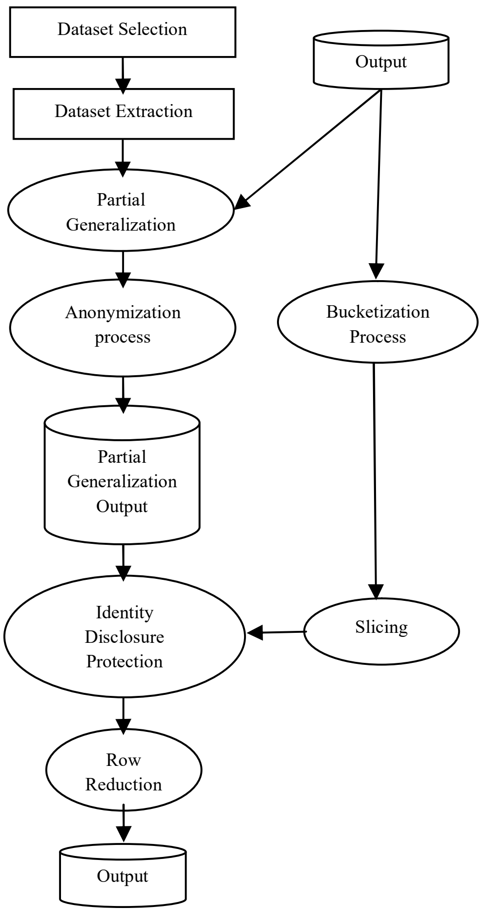
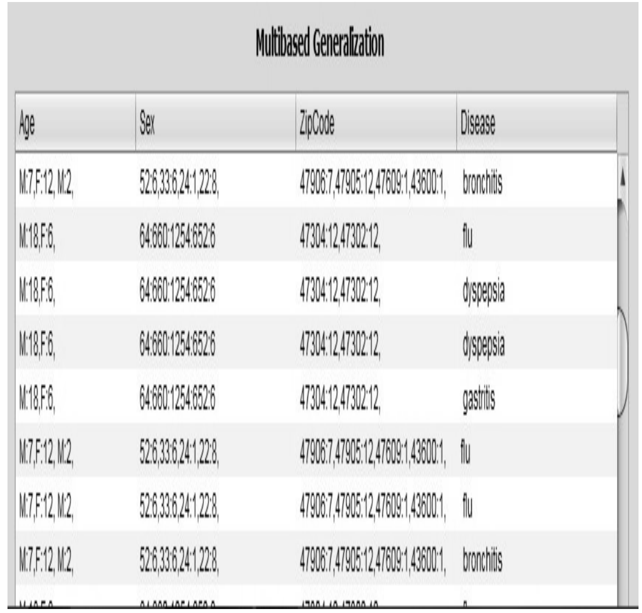


Figure1 Architecture Design

3.2 Multi-Set Generalization

The multiset of accurate values affords information about the allotment of values in each attribute than the generalized interval. For example Age attribute of the first bucket, we use multiset of accurate values {22, 22, 33, and 52} rather than the generalized interval [22-52]. Finally, multisets of accurate values rather than generalized values progress performance as well as confidentiality. Multiset based generalization is comparable to a slight slicing method, where each column contains exactly one attribute, since both approaches preserve the exact values in each one attribute but break the association between them within one bucket



Multibased Generalization

Age	Sex	ZipCode	Disease
M:7,F:12, M:2,	52:6,33:6,24:1,22:8,	47906:7,47905:12,47609:1,43600:1,	bronchitis
M:18,F:6,	64:660:1254:652:6	47304:12,47302:12,	flu
M:18,F:6,	64:660:1254:652:6	47304:12,47302:12,	dyspepsia
M:18,F:6,	64:660:1254:652:6	47304:12,47302:12,	dyspepsia
M:18,F:6,	64:660:1254:652:6	47304:12,47302:12,	gastritis
M:7,F:12, M:2,	52:6,33:6,24:1,22:8,	47906:7,47905:12,47609:1,43600:1,	flu
M:7,F:12, M:2,	52:6,33:6,24:1,22:8,	47906:7,47905:12,47609:1,43600:1,	flu
M:7,F:12, M:2,	52:6,33:6,24:1,22:8,	47906:7,47905:12,47609:1,43600:1,	bronchitis

Figure5 Multi-set Generalization Table

3.3 Overlapping slicing

A new data anonymization technique called slicing is used to separate the data set by both vertically and horizontally. Vertical partitioning is assemblage of attributes into columns based on the association among the attributes and every column inside the original table contains a subset of attributes which are highly interrelated with each other. Horizontal partitioning is assemblage of tuples into buckets and finally, values in every column are randomly permutated inside each bucket to break the connecting between different columns. [15][16]

The key perception of slicing is to provide privacy protection by ensuring that for any tuple, there are several multiple matching buckets. Given tuple: t =<v1, v2. . . vc> where c = number of columns, vi = value for the ith column, bucket = a matching for t if and only if for each i (1<= i<= c) and vi appears at least once in the ith column of the bucket which also contains, alike bucket due to containing additional tuples of each but not all .

The important advantage of slicing to handle high-dimensional data by partitioning attributes into columns and each column of the table be able to observed as a sub-table with a minor dimensionality. The steps to be followed are as follows.

Step 1: Retrieve the records from large databases.

Step 2: Anonymity method divides the records into two.

Step 3: Exchange the sensitive values.

Step 4: Combine the attributes.

Step 5: Overlap the attribute combination. Step 6: Display secured data

3.3.1 Slicing Algorithm used: [18][19][20]

Our algorithm consists of three phases:

1. Attribute partitioning
2. Column generalization
3. Tuple partitioning

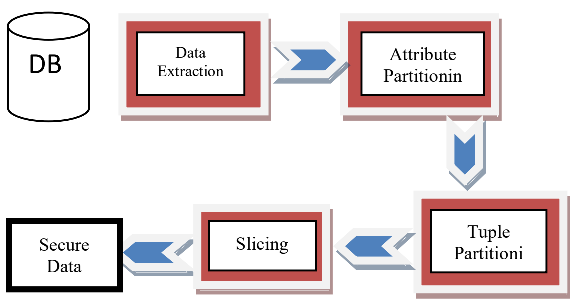


Figure6 Slicing Architecture

3.3.2 Attribute partitioning

Algorithm screen attributes so that highly associated attributes are in the same column so that it gives high performance with both data utility and data privacy. In case of data utility, assemblage highly connected attributes preserves the association among those attributes. In terms of privacy preserving method, association of uncorrelated attributes produce advanced identification hazard than association of highly correlated attributes due to the relations of uncorrelated attribute values which are less regular and also more individual.

3.3.3 Column generalization

Column generalization is necessary for uniqueness or membership disclosure safety. If a column value is distinctive, then a tuple with in this distinctive column value can have only one matching bucket. It will not be proficient for isolation safeguard, in the case of generalization and bucketization of each tuple belonging to only one equivalence bucket.

3.3.4 Tuple partitioning

The algorithm has two data structures: Q = a queue of buckets and SB = a set of sliced buckets SB. At initial step Q contains only one bucket which consist of all tuples and sliced bucket with empty value. In each process a bucket from Q are removed by algorithm. If the sliced table after splitting satisfies l-diversity technique, then algorithm affords two buckets at the end of the queue Q. Otherwise, we are unable to split the bucket further. Then algorithm puts the bucket into set of sliced buckets.

3.3.5 Algorithm tuple-partition(T,ℓ)

1. Q = {T}; SB = ∅.
2. while Q is not empty
3. remove the first bucket B fromQ; Q = Q − {B}.
4. split B into two buckets B1 and B2, as in Mondrian.
5. if diversity-check(T, Q ∪ {B1,B2} ∪ SB, ℓ)
6. Q = Q ∪ {B1,B2}.
7. else SB = SB ∪ {B}.
8. return SB.

3.3.6 Algorithm diversity-check(T,T_, ℓ)

1. for each tuple t ∈ T, L[t] = ∅.
2. for each bucket B inT_
3. record f(v) for each column value vin bucket B.
4. for each tuple t ∈ T
5. calculate p(t,B) and find D(t,B).
6. L[t] = L[t] ∪ {hp(t,B),D(t,B)i}.
7. for each tuple t ∈ T
8. calculate p(t, s) for each s based on L[t].
9. ifp(t, s) ≥ 1/ℓ, return false.
10. return true.

One-attribute-per-column slicing

age	sex	zipcode	disease
33	M	47905	flu
33	M	47905	flu
33	M	47905	flu
33	M	47905	flu
52	F	47905	bron
52	F	47905	bron
52	F	47905	bron
52	F	47905	bron
52	F	47905	bron
52	F	47905	bron
54	M	47302	flu

Two-attribute-per-column slicing

(Age,Sex)	(Zipcode,Disease)
(33,M)	(47905: flu)
(52,F)	(47905: bron)
(52,F)	(47905: bron)
(52,F)	(47905: bron)
(52,F)	(47905: bron)
(52,F)	(47905: bron)
(52,F)	(47905: bron)
(54,M)	(47302: flu)
(54,M)	(47302: flu)
(54,M)	(47302: flu)
(54,M)	(47302: flu)
(54,M)	(47302: flu)
(54,M)	(47302: flu)
(54,M)	(47302: flu)

3.4.1 Algorithm: MakePoints-toClosure

Three-attribute-per-column slicing

(Age,Sex)	(Zipcode,Disease)	(Age,Sex,Disease)
(33,M)	(47905: flu)	(33:M: flu)
(52,F)	(47905: bron)	(52:F: bron)
(52,F)	(47905: bron)	(52:F: bron)
(52,F)	(47905: bron)	(52:F: bron)
(52,F)	(47905: bron)	(52:F: bron)
(52,F)	(47905: bron)	(52:F: bron)
(52,F)	(47905: bron)	(52:F: bron)
(54,M)	(47302: flu)	(54:M: flu)
(54,M)	(47302: flu)	(54:M: flu)
(54,M)	(47302: flu)	(54:M: flu)
(54,M)	(47302: flu)	(54:M: flu)
(54,M)	(47302: flu)	(54:M: flu)
(54,M)	(47302: flu)	(54:M: flu)
(60,M)	(47302: dysp)	(60:M: dysp)

Figure7 Sliced Table

Slicing through Tuple assemblage algorithm affords resourceful random tuple grouping for micro data publishing. Every one column include sliced bucket (SB) that permutated random values for each partitioned data. It furthermore permutated the occurrence of the value in each one of the diversity algorithm ensure the diversity when the each sliced table.[15-20]

3.4 Entity resolution

Using entity resolution technique total n-number of rows will be reduced and the distinct value will be taken. Points-to analysis, widely used in program analysis and compiler optimizations, is an analysis technique for computing a relation between variables of pointer types and their allocation sites. It is frequently used points-to sets of the reference variables, while it may not be easily used in approximate optimizations because they usually necessitate quantitative information on the likelihood of the points-to relation. Pseudo code implemented for row reduction is given below.

Input:A probablistic points-to-graph PPG and a reference set RefSet
Output: Void

```
begin
 if RefSet isEmpty() then
  return;
 end
 foreach var € RefSet do
   if var.GetSupportSet(PPG).isEmpty()
then
     setAdd(var);
 foreach object €
var.GetSupportSet(PPG)
    do
      WorkSet.Add(object);
     end
   end
 end
 While WorkSet isEmpty ()
 do
    oi←WorkSet.GetElement ();
    WorkSet.Remobe(oi);
 Foreach field € FieldSey do
   If not
(oi.field).GetSupportSet(PPG).isEmpty ()
then
   setAdd(oi.fielf);
foreach oj € (oi.field).GetSupportSet
(PPG) do
     WorkSet.Add(oj);
      end
     end
   end
end
 foreach var € PPG.GetRefSet () do
  if var !€ set then
   PPG.RemoveRef(var)
    end
   end
end
```

Algorithim : Update PPG

Input : A probablistic points-to graph PPG at a call site and the corresponding probablistic point-to graph rPPG from method returen.

Output : void

```
begin
 foreach var € rPPG.GetRefSet () do
   if var is a static field or a field access with the form
"object field" then
  PPG.RemoveRef(var);
    PPG.AddRef(var.Copy())
   end
 end
 end
```

3.4.2

One-attribute-per-column slicing

age	sex	zipcode	disease
22	M	47906	dyspe
22	F	47906	flu
22	M	43600	Bron
22	F	47906	flu
22	F	47906	flu
22	F	47906	flu
22	F	47906	flu
22	F	47906	flu
24	M	47609	Flu
33	M	47905	flu
33	M	47905	flu

Two-attribute-per-column slicing

(Age,Sex)	(Zipcode,Disease)
(22,M)	(47906:dyspe)
(22,F)	(47906:flu)
(22, M)	(43600: Bron)
(24, M)	(47609: Flu)
(33,M)	(47905: flu)
(52,F)	(47905: bron)
(54,M)	(47302: flu)
(60,M)	(47302: dysp)
(64,F)	(47304: gast)

Three-attribute-per-column slicing

(Age,Sex)	(Zipcode,Disease)	(Age,Sex,Disease)
(22,M)	(47906:dyspe)	(22:M:dyspe)
(22,F)	(47906:flu)	(22:F:flu)
(22, M)	(43600: Bron)	(22: M: Bron)
(24, M)	(47609: Flu)	(24: M: Flu)
(33,M)	(47905: flu)	(33:M: flu)
(52,F)	(47905: bron)	(52:F: bron)
(54,M)	(47302: flu)	(54:M: flu)
(60,M)	(47302: dysp)	(60:M: dysp)
(64,F)	(47304: gast)	(64:F: gast)

Figure8 Row Reduction

3.5 Advantages:

1. Slicing can be successfully used to evade attribute disclosure, based on the privacy requirement of ℓ-diversity.
2. A proficient algorithm used for computing the sliced table that satisfies ℓ- diversity. Our algorithm partitions attributes into columns by applying column generalization and partitions tuples into buckets. Attributes that are highly-correlated are in the same column. [11]
3. Enhanced workload experiments were conducted and results that slicing preserves much better data utility than generalization and workloads that involving with sensitive attribute,provides more efficient in membership disclosure protection than bucketization.
4. The entity resolution concepts implemented for row reduction of high dimensional data.

4. COMPARITIVE REPORT GENERATION

Report generation module can be used to find the classification accuracy between Original data, Generalization, Bucketization and Overlapping slicing. Overlapping slicing prove better precision than generalization and the target attribute is the sensitive attribute were overlapping slicing even achieve better than bucketization. In this proposed system we generated the report that slicing with entity resolution produce less fake tuples rate and better performance of identification of matching comparatively with other techniques.

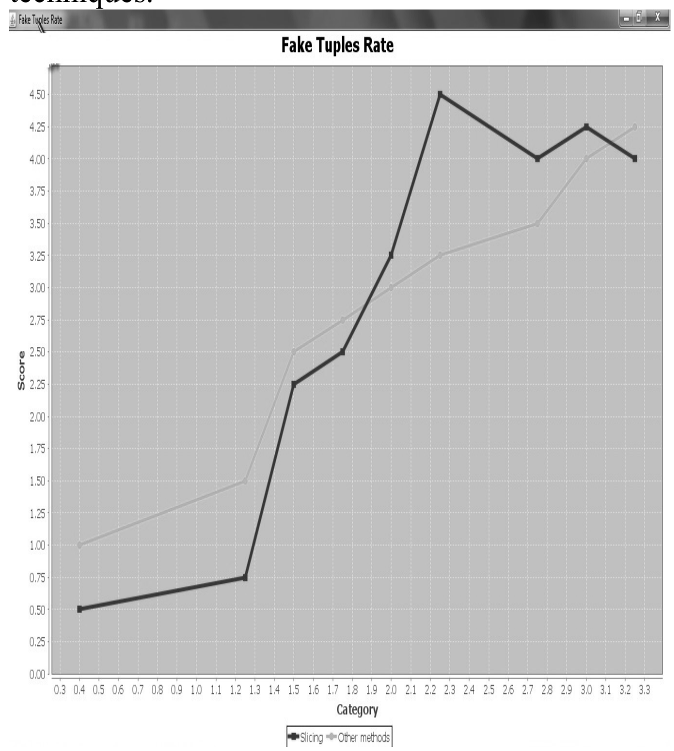


Figure9 Fake Tuple Rates

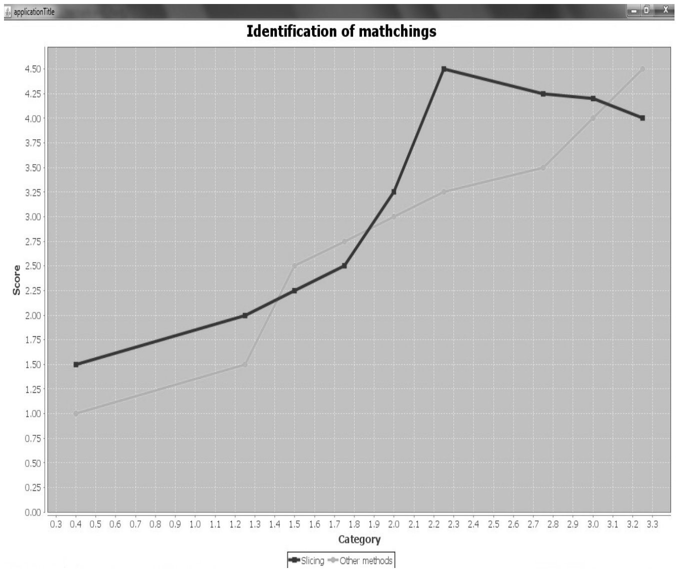


Figure10 Identification of matching

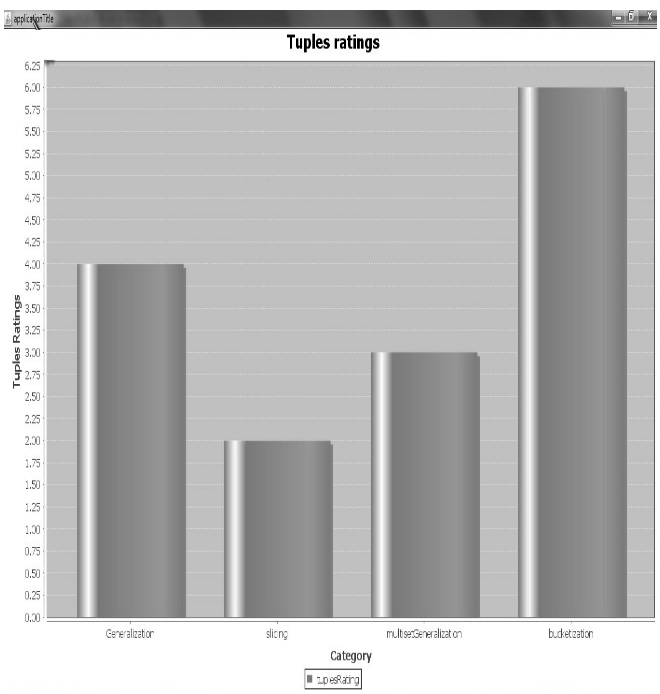


Figure11 Comparison report with other techniques

5. CONCLUSION & FUTURE WORK

This paper proposed a technique called overlapped Slicing with entity resolution for data anonymization. Overlapped slicing and row reduction overcomes the limitations of generalization and bucketization and which also preserves better data utility while protecting against privacy pressure. In this paper we compared slicing techniques with other generalization and bucketization techniques by the parameters of fake tuple rates and identification matching and proved that slicing produce far better performance by protecting membership disclosure and handling high dimensional data. As the future enhancement we may try to prevent the diversity attacks, because there may have access to differently perturbed copies of the same data through various means.

REFERENCES

[1] Zahid Perviz , Arif Ghafoor andWalid G. Aref, "Percision-bounded acess control using sliding-window query views for privacy-preserving data streams", 1041-4347 (c) 2015 IEEE.

[2] L. Golab and M. € Ozsu, "Issues in data stream management," ACM Sigmod Rec., vol. 32, no. 2, pp. 5–14, 2003.

[3] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems," in Proc. 21st ACM SIGMOD-SIGACT-SIGART Symp. Principles Database Syst., 2002, pp. 1–16.

[4] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "ldiversity: Privacy beyond k-anonymity," ACM Trans. Knowl. Discov. Data, vol. 1, no. 1, p. 3, 2007.

[5] J. Cao, B. Carminati, E. Ferrari, and K. Tan, "Castle: Continuously anonymizing data streams," IEEE Trans. Dependable Secure Com-put., vol. 8, no. 99, pp. 337–352, May/Jun. 2011.

[6] K. LeFevre, D. DeWitt, and R. Ramakrishnan, "Workload-aware anonymization techniques for large-scale datasets," ACM Trans. Database Syst., vol. 33, no. 3, pp. 1–47, 2008.

[7] N. Li, T. Li, and S. Venkatasubramanian, "Closeness: A new pri-vacy measure for data publishing," IEEE

Trans. Knowl. Data Eng., vol. 22, no. 7, pp. 943–956, Jul. 2010.

[8] Z. Pervaiz, W. G. Aref, A. Ghafoor, and N. Prabhu, “Accuracy-constrained privacy-preserving access control mechanism for relational data,” IEEE Trans. Knowl. Data Eng., vol. 26, no. 4

[9] J. Cao, Q. Xiao, G. Ghinita, N. Li, E. Bertino, and K.-L. Tan, “Efficient and accurate strategies for differentially-private sliding window queries,” in Proc. 16th Int. Conf. Extending Database Tech-nol., 2013, pp. 191–202.

[10] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis, “A framework for efficient data anonymization under privacy and accuracy constraints,” ACM Trans. Database Syst., vol. 34, no. 2, p. 9, 2009.

[11] Amar Paul Singh and Ms. Dhanshri Parihar. “A Review of Privacy Preserving Data Publishing Technique”, International Journal of Emerging Research in Management &Technology ISSN: 2278-9359 (Volume-2, Issue-6), June 2013.

[12] J. Brickell and V. Shmatikov, “The Cost of Privacy: Destruction of Data-Mining Utility in Anonymized Data Publishing,” Proc. ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (KDD), pp. 70-78, 2008.

[13] P. Mayil Vel Kumar , M. Karthikeyan, ,”L Diversity on K-Anonymity with External Database for improving Privacy Preserving Data Publishing”, International Journal of Computer Applications (0975 – 8887) Volume 54– No.14, September 2012

[14] Slawomir Goryczka, Li Xiong and Benjamin C. M. Fung, “m-Privacy for Collaborative Data Publishing”, IEEE transactions on knowledge and data engineering vol:pp no:99 year 2013.

[15] Vijay R. Sonawane1, Kanchan S. Rahinj,“A New Data Anonymization Technique used For Membership Disclosure Protection”, International Journal of Innovative Research in Science, Engineering and Technology Vol. 2, Issue 4, April 2013, ISSN: 2319-8753.

[16] Tiancheng Li , Ninghui Li, Jian Zhang and Ian Molloy, “Slicing: A New Approach for Privacy Preserving Data Publishing “,IEEE Transactions on Knowledge and Data Engineering Volume:24 , Issue: 3 , ISSN: 1041-4347

[17] Ashwini Andhalkar and Pradnya Ingawale, “Slicing: Privacy Preserving Data Publishing Technique”, International Journal of Computer & Organization Trends – Volume 5 Number 2 – February 2014, ISSN: 2249-2593

[18] T.Malathi and S. Nandagopal,”Enhanced Slicing Technique for Improving Accuracy in Crowdsourcing Database”, International Journal of Innovative Research in Science, Engineering and Technology , Volume 3, Special Issue 1, February 2014, ISSN (Print) : 2347 – 6710

[19] K.Mounica and G.V.S.S.P.Raju, ”Privacy Preserving using Slicing Technique”, International Journal of Research in Computer and Communication Technology, Vol 2, Issue 10, October- 2013 ISSN (Print) 2320- 5156.

[20] Deepa B. Mane and Prof. Emmanuel M, “Review on Privacy and Utility in High Dimensional Data Publishing” , International Journal of Emerging Trends & Technology in Computer Science, Volume 3, Issue 1, January – February 2014 ISSN 2278-6856.