Modified Anonymity Model for Privacy Preserving Data Mining

P.Usha
School of Computer and Information Sciences
B.S.Abdur Rahman University

R.Shriram
School of Computer and Information Sciences
B.S.Abdur Rahman University

W.Aisha Banu
School of Computer and Information Sciences
B.S.Abdur Rahman University

ABSTRACT
Data Mining plays a vital role in today's information-oriented world where it has been widely applied in various organizations. The current trend is that organizations need to share data for mutual benefit. This has led to a lot of concern over privacy in the recent years. It has also raised a potential threat of revealing sensitive data of an individual when the data is released publicly. Various methods have been proposed to tackle the privacy preservation problem. But the recurring problem is information loss. The loss of sensitive information about certain individuals may affect the data quality and in extreme cases the data may become completely useless. In recent years Privacy preserving data mining has emerged as a key domain of research. One of the methods used for preserving privacy is k-anonymization. k-anonymity demands that every tuple in the dataset released be indistinguishably related to no fewer than k respondents. But the distribution preservation is not guaranteed. In this work a modified k-anonymity model is introduced where the privacy in a dataset is preserved while preserving the distribution also.

Keywords
Data Mining, Privacy preserving, k- anonymity, Sensitive attributes.

1. INTRODUCTION
Data mining is a technique that helps to extract useful information from a large database. As the amount of data doubles every three years, data mining is becoming an increasingly important tool to transform this data into information. Data mining tools are increasingly being used to infer trends and patterns. In many scenarios, access to large amounts of personal data is essential in order to draw accurate inferences. However, publishing of data containing personal information has to be restricted so that individual privacy is not hampered. One possible solution is that instead of releasing the entire database, only a part of it is released which can answer the adequate queries and does not reveal sensitive information. Only those queries are answered which do not reveal sensitive information. But this can be considered to be loss of data. The solution to such problems is not easy as the data utility must be taken into consideration while preserving the privacy of the data. Sanitization approach can be used to anonymize the data in order to hide the exact values of the data. But conclusion cannot be drawn with surety. Another approach is to suppress some of the data values, while releasing the remaining data values exactly. But suppressing the data may hamper the utility. A promising solution is the k-anonymity model. Under the k-anonymity, each piece of disclosed data is equivalent to at least k-1 other pieces of disclosed data over a set of attributes that are deemed to be privacy sensitive. The solution which we have proposed is an enhancement of the k-anonymity model. The difference being that the model proposed not only conserves the privacy of the dataset to a large extent, but also the data utility. And most importantly it preserves the distribution of data. The proposed solution guarantees privacy against most of the attacks known to be possible to retrieve private information of individuals. It also provides the necessary patterns to researchers and data miners without deviating from the original data values. Most importantly the solution does not disturb the distribution of the dataset.

2. LITERATURE REVIEW
Bhavana Abad (Khivsara) and Kinariwala S.A.[9], Samarati[5] and Sweeney[8] explain in their paper that a database is k-anonymous with respect to quasi-identifier attributes, if there exists at least k transactions in the database having the same values according to the quasi-identifier attributes. In practice, in order to protect sensitive dataset T, before releasing T to the public, T is converted into a new dataset T* that guarantees the k-anonymity property for a sensible attribute. This is done by generalizations and suppression on quasi-identifier attributes. Therefore, the degree of uncertainty of the sensitive attribute is at least 1/k.

Raymond Chi Wing Wong, Yubao Liu, Jian Yin, Zhilan Huang, Ada Wai Chee Fu and Jian Pei[1] proposed that projecting data onto two tables for publishing in such a way that the privacy protection for (α,k) anonymity can be achieved with less distortion. In the two tables, one table contains the undisturbed non-sensitive values and the other table contains the undisturbed sensitive values. Privacy preservation is guaranteed by the lossy join property of the two tables. And the results are shown to be better than previous approaches.

Rakesh Agrawal, Ramakrishanan Srikanth[2] considered the concrete case of building a decision-tree classifier from training data in which the values of individual records have been perturbed. The resulting data records look very different from the original records and the distribution of data values is also very different from the original distribution. While it is not possible to accurately estimate original values in individual data records, we propose a novel reconstruction procedure to accurately estimate the distribution of original data values. By using these reconstructed distributions, we are able to build classifiers whose accuracy is comparable to the accuracy of classifiers built with the original data.

Roberto J. Bayardo and Rakesh Agrawal[3] present a new approach to exploring the space of possible anonymization that tames the combinatory of the problem, and develop data management strategies to reduce reliance on expensive operations such as sorting. Through experiments on real census data, they show the resulting algorithm can find optimal anonymization under two representative cost measures and a wide range of k. It is seen that their algorithm can produce good anonymization in circumstances where the

input data or input parameters preclude finding an optimal solution in reasonable time. Finally, they use the algorithm to explore the effects of different coding approaches and problem variations on anonymization quality and performance, that being the first result demonstrating optimal k-anonymization of a nontrivial dataset under a general model of the problem.

Benjamin C. M. Fung and Ke Wang, Philip S. Yu [4] in their paper explain that the generalization of data is implemented by specializing or detailing the level of information in a top-down manner until a minimum privacy requirement is violated. This top-down specialization is natural and efficient for handling both categorical and continuous attributes. The approach exploits the fact that data usually contains redundant structures for classification. While generalization may eliminate some structures, other structures emerge to help. The results show that quality of classification can be preserved even for highly restrictive privacy requirements. This work has great applicability to both public and private sectors.

3. PROBLEM DEFINITION

Preserving the privacy of individuals is emerging as the need of the hour as there is increasing risk of security breaches in datasets. So it is necessary to design software which preserves the privacy of a dataset when published on the net. As a solution there have been many data mining algorithms to preserve the privacy of a dataset. But it has been observed that most of these algorithms in order to conserve the privacy and enhance the security end up losing essential data to a great extent. This information loss does not solve the purpose of privacy preserving because it renders the data useless.
Thus there is a need to design a privacy preserving algorithm which not only preserves the privacy of the dataset but also does not lead to information loss. The main objective of the project is to design a privacy preserving data mining system which transforms a dataset while preserving the privacy and distribution using modified k-anonymity model.

Our model addresses the following problems:

- Preserving privacy of a dataset before publishing it for general viewing.
- Conserving utility of the data while implementing the privacy algorithms.
- Preserving the distribution of the data in the anonymized data.

4. DESIGN PROCESS

4.1 System Architecture

The figure 1 clearly outlines every module in the project. The module broadly classifies various sub topics within each of the modules. The input and output of the software form the boundaries in the given figure.

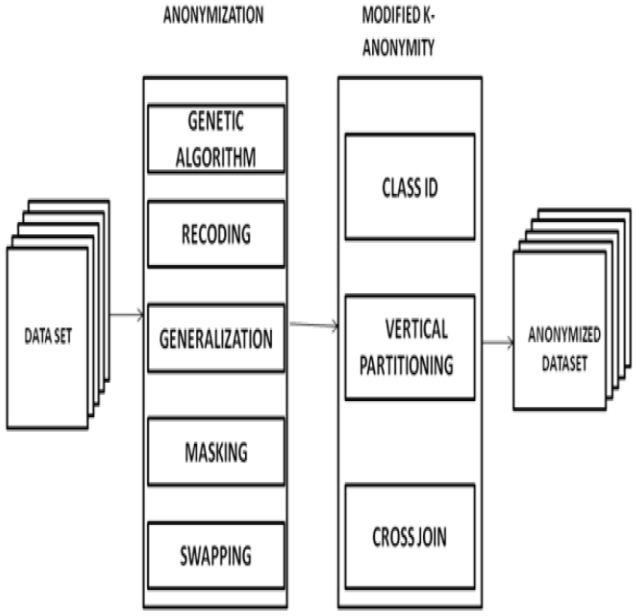


Fig.1. System architecture

4.2 Module diagram

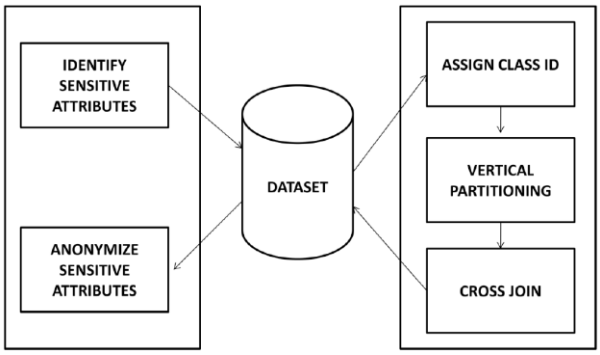


Fig.2. Module Diagram

MODULE 1: INITIAL ANONYMIZATION

Identifying the Sensitive Attributes

In figure 2, it is seen that we firstly identify the sensitive attributes out of all the present in the entire dataset .The selection of sensitive attributes is important because we need to anonymize only the most private data to avoid the overhead and data utility. In the dataset which we considered for experimentation we identified id, age, income, education, phone no. , pin code, credit card no. as the sensitive attributes which need to be anonymized.

Anonymizing sensitive attribute

The sensitive attributes identified in the previous step is anonymized using different anonymization techniques. ID is anonymized using genetic algorithm, pin code and credit card no. is anonymized using masking technique and then age and income are normalized using recoding and at last education field is generalized to produce the anonymized results.

MODULE 2: PROPOSED ANONYMIZATION MODEL

Assign class ID After anonymizing the sensitive attributes assign the class ID in a specific fashion such that it reflects the

k-anonymous records. This class ID acts as a foreign key in further steps.

Vertical partitioning
Next step is vertical partitioning where the sensitive attribute together with the respective class IDs is assigned to one table and then non-sensitive attribute together with its respective class IDs is assigned to another table .This vertical partitioning increases the processing speed by decreasing the overhead.

Cross join
Last step after vertical partitioning is cross join. In this step we perform SQL function called cross join to the sensitive and non sensitive tables with class ID as the foreign key. The result of this cross join enables the dataset to get multiple k-anonymous records where the intruder finds it difficult to find the exact private data of a person.

4.3 DATAFLOW DIAGRAM

4.3.1 Anonymization techniques:

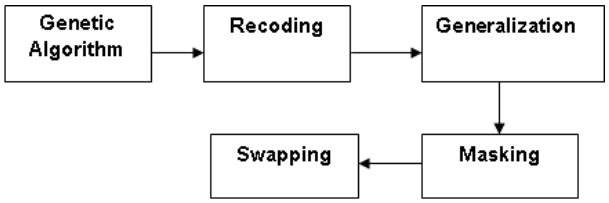


Fig 3: Anonymization techniques data flow diagram

In figure 3, initially each field is anonymized using different anonymization techniques. Firstly genetic algorithm is applied to one field and then recoding techniques like normalization is applied to another field. Other techniques include generalization, masking , swapping which are applied to other respective fields.

4.3.2 Genetic Algorithm

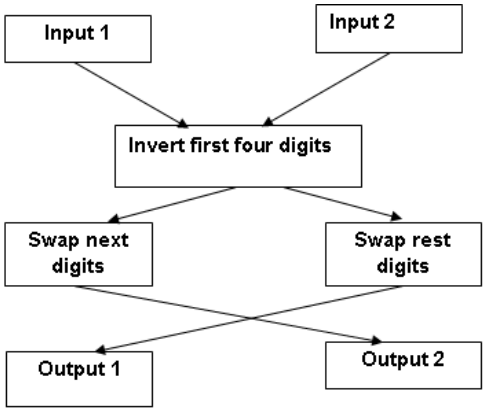
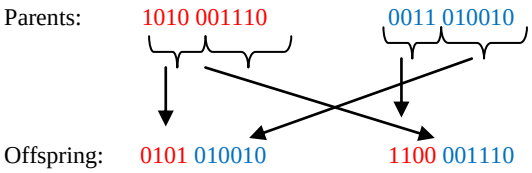


Fig. 4: Genetic Algorithm Data Flow Diagram

Figure 4 shows the flow of control in the implementation of genetic algorithm.
Two point crossover

Randomly one position in the chromosomes is chosen

-Child 1 is the head of chromosome of parent 1 with the tail of chromosome of parent 2
-Child 2 is the head of 2 with the tail of 1
Example



4.3.3 Modified k-anonymity Model

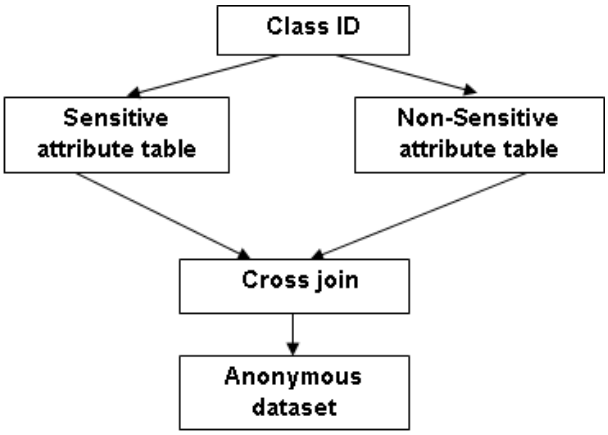


Fig 5 Modified k- anonymity model

In figure 5, we assign an identifier to the raw table and form a new table with an added attribute to each tuple using which forms the basis for performing the lossy cross join. After assigning class ID we divide the table into one containing sensitive data and the other with non sensitive data with class ID assigned to both the tables. Keeping the class ID as the foreign key we perform the join function to get the final anonymized dataset.

4.4 PROGRAM DESIGN LANGUAGE

4.4.1 Genetic Algorithm

Input
A dataset with an attribute of data type integer.
Output
An anonymized dataset.

ALGORITHM
for each input values from field1
store in array of string s1;
convert each value to binary;
for each binary number less than length 12
add zeros to the most significant position;
invert first four digits;
store remaining digits in temp1;
store same positions of following value in temp2;
swap temp1 and temp2;
repeat till last
convert to decimal value;
display;
end

Explanation
The genetic algorithm is used to enhance the privacy and security of an attribute. The mechanism behind this algorithm is to first take a decimal number as input. This decimal digit is converted into a 12 digit binary equivalent and then two-point crossover is applied to the 12 digit number. That is, the first four digits are inverted left to right and the remaining 8 digits are swapped with the corresponding 8 digits of the following number. This renders the number completely impossible to track down.
The reason why genetic algorithm is preferred over other algorithms is that most of the algorithms use a mathematical equation to transform or anonymize the number. So the adversary can easily track down the original data if the mathematical equation used by the algorithm implemented is known. But, in case of a genetic algorithm, there is no particular mathematical equation used. So, there are n possible outcomes for a single encoded value. Thus, the difficulty in choosing between the n possibilities makes genetic algorithm the strongest to break through.

4.4.2 Normalization

Input
Dataset with attribute of string and integer data type.

Output
Dataset with normalized attributes.

Algorithm
input data values from field f1;
store in array sa1;
set range from lower limit to upper limit;
if sa1[i] lies in range ri
assign respective value to the array ;
repeat test in every range
repeat till last
update array sa1
end

Explanation
The normalization technique used in our project involves normalization. The concept of normalization involves sorting individual data values into a range of values. Thus, by normalizing individual values to lie between a particular range, we protect individual data and also do not let any data to go useless or largely deviate from the original value.

4.4.3 Modified k-Anonymity algorithm

Input
Raw cleaned dataset.
Output
Anonymized dataset.

Algorithm
Input dataset with n no. of fields
Separate sensitive and non sensitive fields //vertical partitioning
Store in two different tables t1 and t2;
Anonymize the sensitive fields to (k-1) records each;
Update table1;
Assign class_id to each record in t1 and t2;
Bubble sort records with class id as the quasi-identifiers;
Join both tables t1 and t2 by join function;
Store in a new table t3;
Update table t3 as output table;
End

Explanation
Initially unique class IDs are assigned in a specific fashion. Then the vertical partitioning of the table is performed to split the table into two, of which one will contain sensitive data and its class ID and the other containing non-sensitive data and its class ID. Then finally cross join is performed on the vertically partitioned tables to obtain the privacy preserved anonymous table.

5. DEVELOPMENT PROCESS

5.1 k-ANONYMITY AND k-ANONYMOUS TABLES

k-anonymity

- If the information for each person contained in the release cannot be distinguished from at least k-1 individuals whose information also appears in the release.
- Ex. If you try to identify a man from a release, but the only information you have is his birth date and gender. There are k people meet the requirement. This is k-anonymity.

The concept of k-anonymity requires that the released private table (PT) should be indistinguishably related to no less than a certain number of respondents. But there still exists chances of extracting information by linking tables available in different databases. The set of attributes included in the private table, also externally available and therefore exploitable for linking, is called quasi-identifier.
The k-anonymity states that every tuple released cannot be related to fewer than k respondents.

Definition 1 (k-anonymity requirement)
Each release of data must be such that every value of quasi-identifiers can be indistinctly matched to at least k respondents.

Definition 2 (k-anonymity)
Let T (A1…….Am) be a table, and QI be a quasi-identifier associated with it. T is said to satisfy k-anonymity with respect to QI if each sequence of values in T[QI] appears at least with k occurrences in T[QI].
This is a sufficient condition for k-anonymity requirement. If a set of attributes of external tables appears in the Quasi identifier associated with the private table PT, and the table satisfies Definition 2, the combination of the released data with the external data will never allow the recipient to associate each released tuple with less than k respondents. Thus, it will guarantee that no information is extracted by an adversary through any data mining technique. For k-anonymization we need to identify the quasi identifier from a set of attributes present in the original table. The quasi-identifier depends on the external information available to the recipient which determines the extent of linking (not all possible external tables are available to every possible data recipient).
Therefore, although the identification of the correct quasi-identifier for a private table can be a difficult task, it is assumed that the quasi-identifier has been properly recognized and defined.

State	Dept	C.G.	Age	Roll No.
Orissa	CIV	>7	>20	106010**
Bihar	CIV	>7	>20	106010**
Delhi	ELE	6.*	23	106020**
Maharashtra	ELE	6.*	23	106020**
Orissa	ELE	8.*	2*	106020**
Bihar	ELE	8.*	2*	106020**
Bihar	MEC	>8	>20	106030**
West Bengal	MEC	>8	>20	106030**
Delhi	MET	<8	22	106040**
Orissa	MET	<8	22	106040**
Orissa	MET	>8	2*	106040**
Maharashtra	MET	>8	2*	106020**
West Bengal	MIN	<8	<25	106050**
Bihar	MIN	<8	<25	106050**
Maharashtra	C.S.E.	<9	<25	106060**
Bihar	C.S.E.	<9	<25	106060**
Orissa	C.S.E.	>9	21	106060**
Delhi	C.S.E.	>9	21	106060**
West Bengal	C.S.E.	>7	<25	106060**
Delhi	C.S.E.	>7	<25	106060**

Table 1 k-anonymous table

The above table 1 is an example of how the original attribute values are each anonymized to form an anonymized table with each value being privacy preserved

5.2 ATTACKS ON k-ANONYMIZED DATASETS

Sufficient care must be taken while selecting the quasi identifier because a solution that adheres to k-anonymity can still be vulnerable to attacks. Some possible attacks identified by Sweeney are described below.

5.2.1. Unsorted matching attack against k-anonymity

This attack is based on the order in which tuples appear in the released table. It can be corrected of course, by randomly sorting the tuples of the solution table. Otherwise, the release of a related table can leak sensitive information.

For example a PT having two attributes is released twice. The quasi identifier is different in the two released table T1 and T2. If the orders of tuples are same in T1 and T2 then both tables can be linked to get back the original table.

5.2.2 Complementary release attack against k-anonymity

It is more common that the attributes that constitute the quasi-identifier are themselves a subset of the attributes released. Therefore, subsequent releases of the same privately held information must consider all of the previously released attributes of T, so that it can prohibit linking on T. The example is depicted in table 2.

Example

Table 2: Example table

Hospital Patient Data

DOB	Sex	Zipcode	Disease
1/21/76	Male	53715	Heart Disease
4/13/86	Female	53715	Hepatitis
2/28/76	Male	53703	Brochitis
1/21/76	Male	53703	Broken Arm
4/13/86	Female	53706	Flu
2/28/76	Female	53706	Hang Nail

Table 3 Example Data

Vote Registration Data

Name	DOB	Sex	Zipcode
Andre	1/21/76	Male	53715
Beth	1/10/81	Female	55410
Carol	10/1/44	Female	90210
Dan	2/21/84	Male	02174
Ellen	4/19/72	Female	02237

Inference is that Andre has heart disease.

5.2.3 Temporal attack against k-anonymity

Data collections are dynamic. Tuples are added, changed, and removed constantly. As a result, releases of generalized data over time can be subject to a temporal inference attack.

For example, let table T0 be the original privately held table at time *t*=0. Assume a k-anonymity solution based on T0, which is called table RT0, is released. At time *t*, assume additional tuples were added to the privately held table T0, so it becomes Rt. Let RTt be a k-*a*nonymity solution based on Tt that is released at time *t*. Because there is no requirement that RTt respect RT0, linking the tables RT0 and RTt may reveal sensitive information and thereby compromise k-*a*nonymity protection.

To combat this problem, RT0 should be considered as joining other external information. Therefore, either all of the attributes of RT0 would be considered a quasi identifier for subsequent releases, or subsequent releases themselves would be based on RT0.

5.2.4 Homogeneity Attack

When the non-sensitive information of an individual is known to the attacker then sensitive information may be revealed based on the known information. It occurs if there is no diversity in the sensitive attributes for a particular block. This method of getting sensitive information is also known as positive disclosure.

This suggests that in addition to k-anonymity, the sanitized table should also ensure "diversity" – all tuples that share the same values of their quasi-identifiers should have diverse values for their sensitive attributes. The example is depicted in table 4.

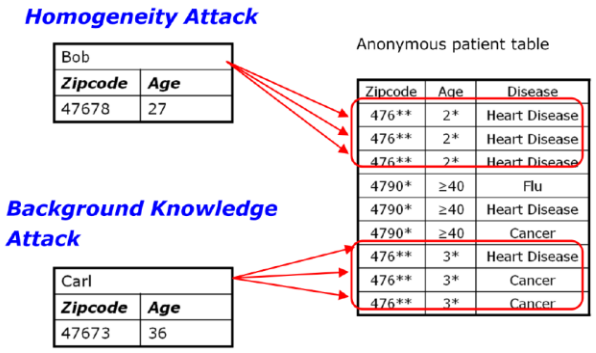


Table 4 Example Table

5.2.5 Background Knowledge Attack

If the user has some extra demographic information which can be linked to the released data which helps in neglecting some of the sensitive attributes, then some sensitive information about an individual might be revealed. This method of revealing information is also known as negative disclosure.
Thus, keeping all these attacks in mind, we have proposed a model which will, to the best of our knowledge, never let out the private information. Every step undertaken in anonymization, multiplies the degree of anonymization to n times and makes the resultant dataset more complex.

The system is programmed in Java using Net Beans IDE. The development process includes programming the two modules which together form the system and brought to screen by using Java Applet. The development process is done step by step as follows:

1. The user first enters the user id and password.
2. The software checks for validation of the details and logs in if correct information is provided.
3. The update query takes each field's input from the user and adds on to the database using the insert command.
4. The view database command is executed by the select all command and displayed in the text area provided in the applet (user interface screen).
5. The anonymization is initially done by implementing anonymization techniques such as generalization, normalization, recoding, swapping, masking etc. Genetic algorithm is used to anonymize and to enhance the security to greater levels.
6. Our model then contains three steps namely, assigning class id, vertical partitioning and cross join implementation.
Class id is assigned after testing two attributes to lie within the specified criteria. If it does then the class ID is assigned accordingly.
7. The table is split vertically which reduces the overhead and hence increases the performance level.
8. The Cross join is the concluding step where in the two tables with class ID as the foreign key are joined using the join function.
9. The anonymized table is displayed to guest user for access.
These steps summarize the entire development process of our model for privacy preserving data mining using modified k-anonymity model.

6. IMPLEMENTATION

The Java application developed can be implemented on all personal computers enabled with Microsoft Windows OS.

The executable file of the project can be directly loaded into the computer, installed and the application can be executed.

7. PERFORMANCE MEASURE

7.1 CONTRIBUTION OF THE WORK

Original dataset

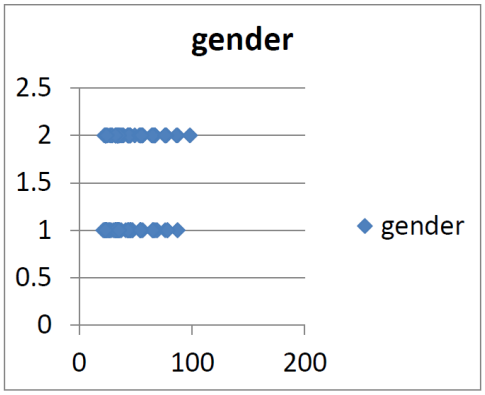


Fig 6 Performance Measure Graph

Anonymized dataset

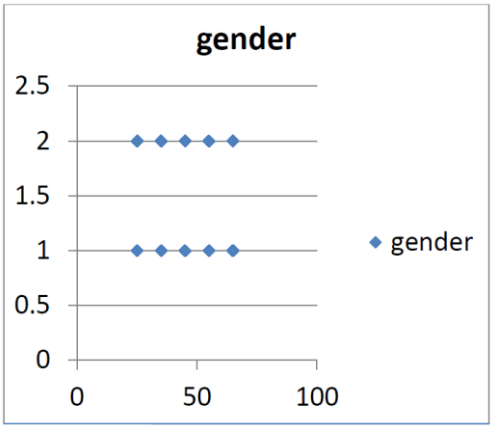


Fig 7 Performance Measure Graph

In the given Figure 6 and Figure 7 the two graphs represent pattern analysis of the clusters formed from the original raw dataset and the anonymized dataset respectively. It has been observed that, by taking the same criteria for analysis, both the graphs show the same pattern. The graph obtained by the pattern analysis of the anonymized dataset is found to have a more generalized pattern of that of the original dataset pattern. Thus it is concluded that the model proposed by us does not deviate the miner from the original result but also does not give away the exact information.

8. CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION

In order to improve the privacy offered by the dataset, utility of the data suffers. So a model is introduced wherein the privacy of a dataset is preserved as well as its utility. We do this by implementing our modified k-anonymity model. By this model a way is found to preserve the privacy of any dataset and also maintain the distribution as well as the utility of the data.

It was found that through these experiments only a few attributes in the whole dataset are considered to be sensitive. So the key to privacy preservation is to anonymize these sensitive attributes alone and leave the rest. In this model the same is implemented, by anonymizing the sensitive attributes alone and leaving the rest. Finally the whole dataset to k records was anonymized.
The software thus successfully implements the aimed privacy measures without disturbing the privacy as well as the distribution of the dataset.

FUTURE ENHANCEMENT

Extension and Enrichment of the Definition: Some approaches to attribute disclosure attack exist but still research is on.
New technique:
The k-anonymity model is not tied to any specific technique. So new areas and methods can be investigated. k-anonymity did not model any external knowledge that can be further exploited for inference and expose the data to identity or attribute disclosure.
Thus a lot of scope exists in the area of privacy preserving data mining and in enhancing the k-anonymity model to a much greater extent.

9. REFERENCES

[1] R. C. Wong, Y. Liu, J. Yin, Z. Huang, A. W. Chee and J. Pe, " (α,k)-anonymity Based Privacy Preservation by Lossy join" , APWeb/WAIM'07 Proceedings of the joint 9th Asia-Pacific web and 8th International Conference on Web-age Information Management Conference (2007) .

[2] R. Agrawal, R. Srikant, " Privacy preserving data mining ", ACM SIGMOD International Conference on Management of data, SIGMOD '00 Proceedings(2000).

[3] R. J. Bayardo, R. Agrawal, " Data privacy through optimal k- anonymization", 21st International Conference on Data Engineering, ICDE '05 Proceedings(2005)

[4] C. M. Fung, K. Wang, P. S. Yu, "Top-down specialization for information and privacy preservation", 21st International Conference on Data Engineering, ICDE '05 Proceedings(2005)

[5] P. Samarati, "Protecting respondents identities in micro data release", IEEE Transactions on Knowledge and Data Engineering archive, Volume 13, Issue 6(November 2001)

[6] A. C. Chaaru and P. S. Yu , "On Static and Dynamic Methods for Condensation-Based Privacy-Protection Data Mining", ACM transaction on Database Systems, Volume 33,no 1 article 2 2,Publication Date: March 2008

[7] V. Jaideep, C. Clifton, M. Kantarcioglu , P. A. Scott " Privacy-preserving decision trees over vertically partitioned data**",** ACM transaction on Knowledge Discovery from data**,** vol 2 No 3, article 14, Publication date : October 2008.

[8] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression" International Journal of Uncertainty, Fuzziness and Knowledge Based Systems, Vol 10(5), pp. 571–588(2002)

[9] Bhavana Abad (Khivsara) and Kinariwala S.A. , "A Novel approach for Privacy Preserving in Medical Data Mining using Sensitivity based anonymity" InternationalJournal of Computer Applications (0975 – 8887) Volume 42–No.4, March 2012