King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

C-mixture and multi-constraints based genetic algorithm for collaborative data publishing

Yogesh R. Kulkarni [*], **T. Senthil Murugan**

Dept. of Computer Science and Engineering, Vel Tech University, India

Received 28 March 2016; revised 13 June 2016; accepted 14 June 2016
Available online 22 June 2016

KEYWORDS

Privacy;
Utility;
Distributed databases;
Data publishing;
Optimization;
Sensitive information

Abstract Due to increasing need of using distributed databases, high demand presents on sharing data to easily update and access the useful information without any interruption. The sharing of distributed databases causes a serious issue of securing information since the databases consist of sensitive personal information. To preserve the sensitive information and at the same time, releasing the useful information, a significant effort is made by the researchers under privacy preserving data publishing that have been receiving considerable attention in recent years. In this work, a new privacy measure, called c-mixture is introduced to maintain the privacy constraint without affecting utility of the database. In order to apply the proposed privacy measure to privacy preserving data publishing, a new algorithm called, CPGEN is developed using genetic algorithm and multi-objective constraints. The proposed multi-objective optimization considered the multiple privacy constraints along with the utility measurement to measure the importance. Also, the proposed CPGEN is adapted to handle the cold-start problem which commonly happened in distributed databases. The proposed algorithm is experimented with adult dataset and quantitative performance is analyzed using generalized information loss and average equivalence class size metric. From the experimentation, we proved that the proposed algorithm maintained the privacy and utility as compared with the existing algorithm.

1. Introduction

Many organizations bring out microdata-tables, which contain unaggregated information about persons. Medical, voter registration, census, and customer data can be included in these microdata tables. Microdata are important source of information for the allotment of public funds, medical research, and trend analysis. Still, if the persons can be individually identified in the microdata, then their private information such as their medical condition would be revealed, and it is undesirable (Machanavajjhala et al., 2007). As an example, in the healthcare field, a national agenda is to form the Nationwide Health

[*] Corresponding author.
E-mail addresses: Kulkarni.yr@gmail.com (Y.R. Kulkarni), senthilmuruganme@gmail.com (T. Senthil Murugan).

Peer review under responsibility of King Saud University.

Information Network (NHIN) to share information among hospitals and other providers, and support proper use of health information further than direct patient care with privacy protection (Goryczka et al., 2014). It is important that the sensitive information about the individuals should not be revealed when the microdata are released. Two kinds of information disclosures found out in the literature are identity disclosure and attribute disclosure. Identity disclosure happens when a person is connected to a particular record in the released table. Attribute disclosure occurs when new information about some individuals is revealed, i.e., the released data make it feasible to understand the uniqueness of an individual more precisely than it would be possible before releasing the data (Jana and Joshi, 2014; Chaytor and Wang, 2010; Soria-Comas et al., 2013; Clifton and Tassa, 2013; Wang et al., 2009; Ghasemzadeh et al., 2014; Fung et al., 2007).

The most important concepts for privacy is anonymity. Anonymity refers to a state where one's identity is completely concealed, and anonymity is sometimes used as a synonym for privacy (Byun et al., 2007). Even though k-anonymity prevents from identity disclosure, it is inadequate to avoid attribute disclosure. Anonymous data can protect individuals in two ways. One way is to protect identity privacy, for example by making it not possible to learn to whom a data record is associated. The second way is through attribute privacy, for example making it not possible to know about a specific property of individuals. When collecting databases such as health records, collected by hospitals or government organizations, anonymity has a major role to protect privacy as the information connected to individuals are very sensitive (EnamulKabir et al., 2011). There are two methods to achieve in k-anonymizing a dataset. First one is suppression, which involves not releasing an entire tuple or a value at all to the third party. Second one is generalization which involves replacing the value or tuple with a less definite but semantically constant value.

The researchers are paying attention toward the research on privacy-preserving data publishing (Li and Li, 2008; Byun et al., 2006). Anonymization techniques includes (1) hiding the identities by making each record indistinguishable from at least $k-1$ other records (Samarati and Sweeney, 1998) (k-anonymity), (2) making sure that the distance between the distribution of sensitive attributes in a class of records and the distribution of them in the whole table is not more than 't' (Li and Li, 2007) (t-closeness), and (3) making sure that there are at least l different values for a given sensitive attribute in each indistinguishable group of records (Machanavajjha et al., 2006) (l-diversity). The m-invariance is one of the representative models (Xiao and Tao, 2007). The basic idea of these techniques is unchanging the set of sensitive attribute values in the group that a tuple belongs to, even if the tuple may be put into different groups in different versions of the microdata. With these restrictions, the present privacy standard is proposed to make them to maintain the better tradeoff between data quality and privacy (Fu et al., 2014).

In this paper, a new algorithm called, CPGEN (C-mixture based privacy genetic algorithm) is developed by combining the genetic algorithm with c-mixture theory which is newly developed here for privacy measurements. C-mixture is a new privacy measure developed in this work by integrating the multiple privacy constrains such as, k-anonymity, l-diversity and m-privacy. In addition, cold-start attack is defined based on the real time fact that one provider provides a large number of records but other one share less number of data records with more distribution among every group. In order to alleviate this attack, noisy records are inserted into the databases. At first, the input data are directly given to the genetic algorithm which considered the unique encoding of chromosomes for doing anonymization process and the fitness is evaluated using the proposed multi-objective function which considered both utility through generalized information loss and privacy through average equivalence class size metric. Based on the objective function, the better chromosome is selected and it is used further to construct the anonymized data.

The paper is organized as follows: Section 2 explains literature review and Section 3 provides C-mixture principle and its definitions. Section 4 presents the proposed C-mixture based privacy genetic algorithm for collaborative data publishing. Section 5 discusses the experimentation and outcome of the proposed method. Finally, conclusion is given in Section 6.

2. Literature review

Table 1 presents the review of recent work available for privacy enabled data publishing using different methods like, k-anonymization, l-diversity and so on.

2.1. Existing challenges

Data publishing requires preservation of privacy to protect the sensitive information hidden in the database. When doing privacy preservation within the database contributed by n-different contributors, three important considerations should be handled based on the attributes. All the quasi-identifiers should have at-least k-duplicate records of every groups. All the sensitive attributes should have l-diverse set of values in every group of data. Also, for dealing with collaborative data publishing, one important attack proposed in Goryczka et al. (2014), insider attack which explains about the way of obtaining the sensitive information by colluding with the different data providers needs to be handled. These three challenges should be handled before collaborative data publishing.

2.2. Cold start attack

The additional challenge considered here is that if one of the data providers provides large number of records but other one shares less number of data records with more distribution among every groups, then the data provider who shares more number of data can easily track out the sensitive information of a data provider who shares less number of records. This new attack, we named as, cold start attack which should be also taken into consideration to provide more privacy before data publishing.

3. C-mixture: a practical privacy definition

This section discusses the c-mixture principle about how to instantiate it with specific definitions of privacy and how to handle with sensitive attributes and number of data providers. In addition, the definitions associated with the c-mixture and example of the proposed constraints are also discussed.

Table 1 Literature review.

Authors	Contribution	Advantages	Disadvantages
Goryczka et al. (2014)	*m*-Privacy and a data provider-aware anonymization algorithm	Horizontally partitioned data are anonymized at multiple data providers	When data are distributed in a vertical or ad-hoc manner, it is difficult to handle
Fouad et al. (2014)	A personalized anonymization technique based on an aggregate formulation	Effectiveness of data disclosure is maintained while keeping its risk below an acceptable threshold	It is susceptible to bound on the estimated utility
Goryczka et al. (2013)	Secure distributed data anonymization and integration with *m*-privacy	Privacy constraint against any group of up to 'm' colluding data is satisfied by the anonymized data	Managing set-value data is difficult
EnamulKabir et al. (2011)	Systematic clustering problem for *k*-anonymization	Usability for incremental datasets	Suitability of *l*-diversity using systematic clustering algorithm is a problem
Li et al. (2012)	Slicing model for privacy in data publishing	Slicing conserves superior data utility than generalization and can be used for membership disclosure protection	It considers randomly generated links between column values of a bucket
HussainKhokhar et al. (2014)	Analytical cost model to measure trade-off between privacy and utility	Used for perturbative and non-perturbative anonymization techniques	Miss the trade-off between privacy protection and information utility
Sun et al. (2011)	Distinct (l,a)-diversity	Improve the present privacy standards to make them maintain the better tradeoff between data quality and privacy	Not fit for multiple sensitive attributes

Table 2 Sample input data.

Provider	Zip	Gender	Age	Education	Disease	Expense
P1	4351	M	25	8th	HIV	2000
P1	4353	M	28	4th	HIV	3000
P1	4362	F	32	8th	HIV	2500
P1	4362	F	33	8th	HIV	6000
P2	4354	M	26	8th	Diabetes	3500
P2	4353	M	28	5th	Diabetes	4000
P2	4362	M	34	4th	Diabetes	5000
P2	4361	M	37	6th	Diabetes	1500

3.1. C-mixture

A new privacy measure called c-mixture includes three different privacy measures. Let $T(P, B_i, B_2, B_o, Q_1, Q_2, Q_q, E_1, E_2, E_s)$ be a table and Q_i be a quasi identifier and E_i be sensitive attribute and P is index of data providers associated with it. T is said to satisfy c-mixture then, (i) (1) QIs should have at-least 'c' % of duplicate records in every groups, (2) 'c' % of well defined values in sensitive attributes of every groups, (3) every group should have 'c' % of data providers.

3.2. Relative strength

Relative strength is a parameter newly devised here for bringing the privacy constraint from the user input (c) to k-anonymity, l-diversity and m-privacy. For example, suppose, a user want to maintain c % of the mixture constraint in the table T, then, k, l and m value will be directly found out from the c based on the following equation which is defined as relative strength here.

$$k = \lfloor N * c \rfloor; \quad l = \lceil r * c \rceil; \quad m = \lceil D * c \rceil \tag{1}$$

c is user input ranging from 0 to 1.

3.3. C-mixture principle

The principle of c-mixture property is explained with a running example given below. Let us assume the Table 2 is an input table where, zip code, gender, age, education and disease are quasi identifiers. Disease is sensitive attribute and Provider is provider's name of the data records. Table 3 is known to be a c-mixture table for $c = 0.6$. If an input of c is 0.6, k value is computed by taking floor function after multiplying number of record (N) and c value. Here, the number of records is eight and c is 0.6. So, the value of k is four. Now, if we check the 4-anonymity of Table 3, the condition is satisfied for all the quasi identifiers which have four numbers of duplicate records for every unique attributes. To find the value of l, ceiling function is taken after multiplying of number of classes in sensitive attributes and c. The number of classes in sensitive attribute is two and the multiplication provides the value of 1.2. The final l-value after ceiling function is two. If we examine the sensitive attribute, we found that it has the 2-well represented sensitive values in every group. Now, m value is found out by taking ceiling function after multiplying the number of data providers with c value. Here, the number of data providers is two and input c value is 0.6 and the final m-value is two. If we check the m-privacy constraint in Table 3, every group has m number of data providers. All the three constraints based on the relative strength formula are satisfied for the c value of 0.6 so we can say that the Table 3 is 0.6 mixture data.

4. Proposed C-mixture based privacy genetic algorithm for collaborative data publishing

This section presents the proposed C-mixture based privacy genetic algorithm for collaborative data publishing. Here, a new algorithm called, CPGEN (C-mixture based privacy genetic algorithm) is developed by combining the genetic algorithm with c-mixture theory. This proposed algorithm utilizes

Table 3 0.6-mixture input data.

Provider	Zip	Gender	Age	Education	Disease	Expense
P1	435^*	Gender	[20–30]	Primary	HIV	2000
P1	435^*	Gender	[20–30]	Primary	HIV	3000
P2	435^*	Gender	[20–30]	Primary	Diabetes	3500
P2	435^*	Gender	[20–30]	Primary	Diabetes	4000
P2	436^*	Gender	[30–40]	Primary	Diabetes	5000
P2	436^*	Gender	[30–40]	Primary	Diabetes	1500
P1	436^*	Gender	[30–40]	Primary	HIV	2500
P1	436^*	Gender	[30–40]	Primary	HIV	6000

the generalization concept for anonymization purpose by doing exhaustive search. Fig. 1 shows the block diagram of the proposed collaborative data publishing.

4.1. C-mixture based privacy genetic algorithm

Let us assume that a trusted third party (TTP) receives a from the multiple data providers P_i, each contributing a subset of records T_i. Each data record coming from the data provider contains provider name, set of quasi identifier, sensitive attributes and other attributes.

$$T = \{T_i \in P_i; 1 \leqslant i \leqslant D\} \tag{2}$$

$$T = \{P, B_i, B_2, B_o, Q_i, Q_2, Q_q, E_i, E_2, E_s\} \tag{3}$$

where, P is provider's name, B is common attributes, Q is quasi identifiers and E is sensitive attributes. The criteria to be fulfilled are that the TTP should publish the data T which can be viewable by any service provider so the data to be anonymous T^* in a better way to avoid the inference of founding information from the anonymous data. The final goal is to make a database T^* from the data T by considering all the attacks and should guarantee there is no disclosure of intermediate information during the anonymization.

(a) Anonymization using solution coding

The first step of the of the c-mixture privacy genetic algorithm is how to encode the process of anonymization into a single vector to do the exhaustive search. The solution can be indicated as, S which contains the q number of elements. q is the number of quasi identifier. Every element in solution S may vary between the 1 and L. L is the number of levels of the quasi attributes based on the taxonomy tree. For example, the taxonomy tree of the example given in Table 2 is shown in Fig. 2. Here, every quasi attributes are generalized with the parent values based on the L level of generalization. The solution coding for the taken record is given in Fig. 3. Here, four elements are presented as the number of quasi attributes is four for the example. Every element may range between 1 and L. For example, if you consider age attribute, level of the age in taxonomy tree is three. So, the values to be placed in the solution may vary between 1 and L.

The solution encoding is then utilized to do anonymization for data publishing. Every levels indicated in the solution S is used to convert the original data T into anonymized data T^*. For example, if the zip code pointing solution element is indicated as 2 (as per Fig. 3), so, we can convert the zip code values presented in table T to the second level of codes presented in the taxonomy tree. This means that the 4351, 4353 and 4354 can be converted to 435^* and 4361 and 4362 can be converted

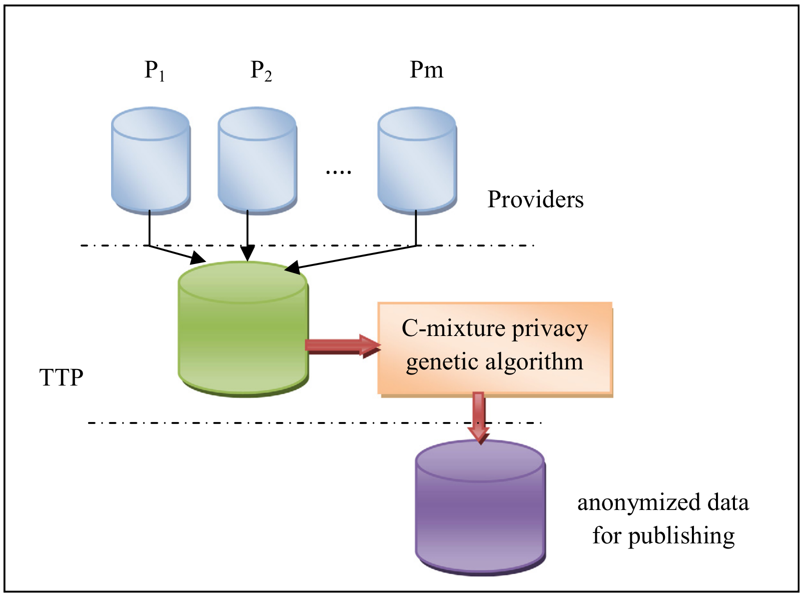
P1
P2
Pm
....
Providers
TTP
C-mixture privacy genetic algorithm
anonymized data for publishing

Figure 1 Block diagram of the proposed collaborative data publishing.

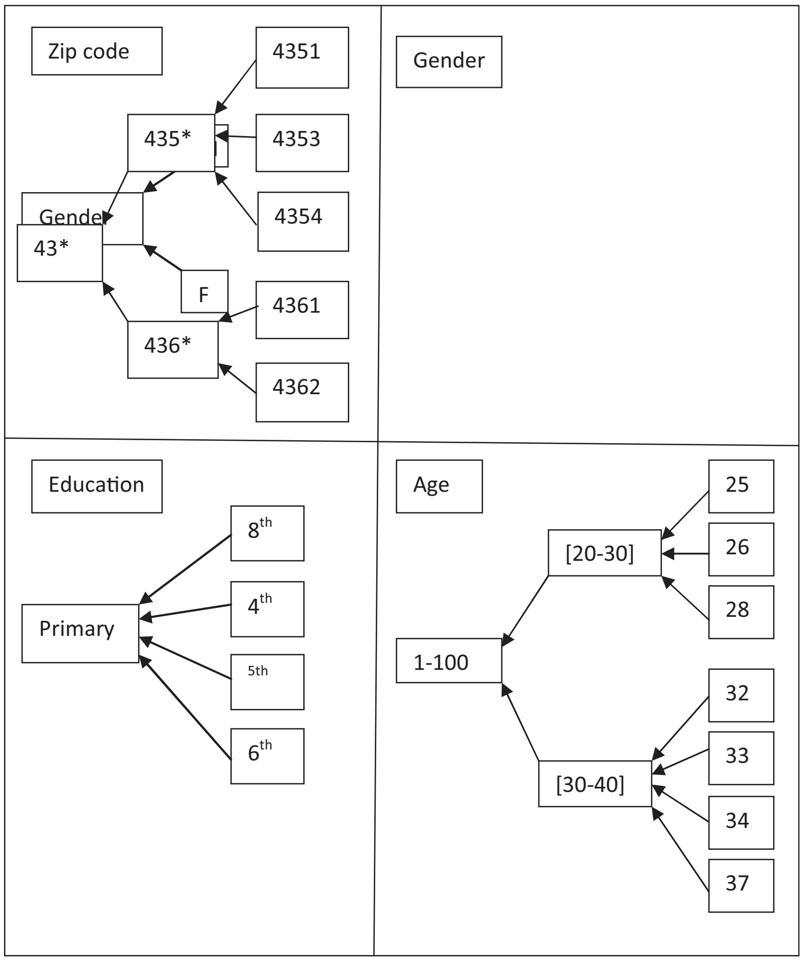

Figure 2 Taxonomy tree.

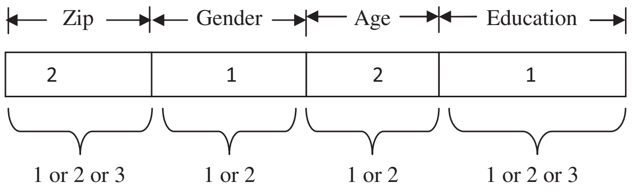

Figure 3 Solution coding.

to 436*. Similarly, gender pointing solution element is encoded as one. This means that 'M' and 'F' can be converted to gender as the first level of information for gender is gender. This mechanism is applied for all the quasi identifier according to the solution encoding procedure. If the solution representation given in Fig. 3 is applied to the original data given in Table 2, then the anonymized table T^* can be as like the table given in Table 3.

(b) Multi-objective optimization formulation for privacy fitness score

The objective evaluation of every solution is performed using the proposed multi-objective criteria which are newly proposed here based on generalized information loss (*GenILoss*) and average equivalence class size metric (C_{AVG}) Ayala-Rivera et al., 2014. The proposed multi-objective optimization framework considered three constrains such as, *k*-anonymity, *l*-diversity and *m*-privacy. These three constrains should be satisfied by every solution. Even if the solution satisfied these constraints, the objective is to minimize the utility and maximize the privacy. The three constraints can provide the privacy after anonymization but, utility can be preserved through *GenILoss* which gives the minimum value if the original table is not generalized. This means that the utility should be high for the lower values of *GenILoss*. C_{AVG} is a metric used to measure the privacy through equivalence class. The lower values of C_{AVG} are better for preserving the privacy. So, these two tradeoffs are effectively integrated into a single function to maintain the privacy and utility along with three privacy constraints. The proposed fitness function is given as follows:

$$F(S) = \alpha * GenILoss(S) + \beta * C_{AVG}(S) \tag{4}$$

subject to the following constraints:

(i) $k \geqslant f_{ano}(A, T)$
(ii) $l \geqslant f_{div}(A, T)$
(iii) $m \geqslant f_{pri}(A, T)$

where, $$GenILoss(S) = \frac{1}{N*q} * \sum_{i=1}^{q}\sum_{j=1}^{N} \frac{P_{ij} - L_{ij}}{P_i - L_i} \tag{5}$$

$$C_{AVG}(S) = \frac{N}{|EQ_s| * k} \tag{6}$$

Here, P_i and L_i are lower and upper bounds of an ith quasi identifier. P_{ij} and L_{ij} is the upper and lower bound of the generalized interval. $f_{ano}(A,T)$ is a function to compute the number of duplicate records presented in table T for every sequence value after applying anonymization A. $f_{pri}(A,T)$ is a function to compute the number of data providers for every groups in the provider after applying anonymization A. $f_{div}(A,T)$ is a function to compute the number of well represented sensitive values in every group of sensitive attributes.

(c) Genetic algorithm

Genetic algorithm (GA) (McCall, 2005) is one of the traditional and popular search algorithm widely applied for optimization problems. GA is developed by taking the genetic process of natural selection, inheritance, crossover, mutation and evolution. This work aims to utilize the GA for privacy-enabled data publishing as it requires heuristic search to find the optimal database to publish to third party without violating the privacy and utility. The adapted genetic algorithm for the privacy data publishing consists of the following steps:

Population: The first step of the GA is initialization of population which has a n number of solutions. Every solution is represented as like the procedure discussed above. Solution is otherwise, called as chromosomes which, are vector representations of solutions to a particular problem. Population can be indicated as follows.

$$I = \{S_i; \quad 1 \leqslant i \leqslant n\} \tag{7}$$

where, I is population, S is solution or chromosomes, n number of chromosomes in the population.

Once the population is initialized, privacy fitness is computed as per the privacy fitness score developed newly in this work. Once the privacy fitness is computed for all the chromosomes, it is then undergone selection process.

Selection: Selection is an important step to bring new type of solution into the population according to the natural evolution process. From the population, two chromosomes are to be selected for the evolution process. The selection of two chromosomes is purely based on the probability criteria of fitness score. The fitness score computed for all the chromosomes are then utilized to find the probability of selection based on the following formula.

$$p_i = \frac{F(S_i)}{\sum_{i=1}^{n} F_i(S_i)} \tag{8}$$

where, $F(S_i)$ is the fitness score of the ith chromosomes. This means that a probability of chromosome being selected is proportional to its relative fitness.

Crossover: The two chromosomes selected from the previous step are used here to do cross over operation which produces two child chromosomes. Here, one point cross over operation is utilized. Accordingly, a random number γ is generated in the range of 1 to q which is the length of the chromosome. Random number, γ points to the location in the selected chromosomes and the solutions are interchanged accordingly. The two parent chromosomes are interchanged their values based on the random number to obtain new child chromosomes.

Mutation: The two new child chromosomes obtained from the previous steps are then given to mutation operators which act on every child chromosome to flip one or more allele values. In order to accomplish this task, random number γ is again generated within the range of 1 to q and the solution value which points based on γ is flipped to some other values which should be in the range of level of taxonomy. Now again, two new child chromosomes are generated.

Fitness assignment and updating population: For the four child chromosomes generated newly from the above steps is then utilized to find the privacy fitness score of the solution. Once the fitness is found out for the four chromosomes, the best chromosome having minimum fitness is replaced with the worst chromosome in the population and the process is continued.

Stopping criteria: The above steps are executed for the number of iterations, t which is the user input for terminating the algorithm. Once t iterations are reached, the best chromosome, S_b is taken out and the anonymization is performed based on the solution representation. Fig. 4 shows the pseudo code of CPGEN algorithm.

4.2. Adapting CPGEN to cold-start problem for data publishing

This step aims to adapt the CPGEN to handle the cold start problem to be considered in data publishing. In real time scenario, data provider supplies different kinds of data records. The distribution characteristics of data records have great fluctuation because one provider gets the updates of data frequently but, the other one seems to get less updates. In this scenario, the maintenance of m-privacy is very challenging. If m-privacy is not maintained in the published data records, the data provider which has more data distribution can easily track the information belonging to other data records (having less data distribution) since it is very less in frequency among the data groups. Also, maintaining of m-privacy is hard. On the other hand, if two data providers collude with each other, the data property of other providers can be easily tracked if the data distribution is not uniform among the data providers. So, in order to handle colluding attack and cold start attack, we randomly add the noisy data into the original database with the name of the provider having minimum distribution. This process of adding noisy records can easily overcome the colluding and cold start attack.

5. Results and discussion

This section presents the experimentation and the quantitative results of the proposed CPGEN algorithm. The performance and comparative analysis is also performed with the existing algorithm (Goryczka et al., 2014).

5.1. Experimental set up

The proposed CPGEN algorithm is implemented using Java 1.7 with netbeans IDE 7.3. The experimentation is conducted on Windows 8.1 machines with Intel Core i5 processors and 4 GB of main memory.

```
Input: $T$, $t$, $c$
Output: $S_b$
Start
    Initialize population $I$ randomly
    Perform anonymization of every chromosomes
    Find privacy fitness of every chromosomes, $F(S)$
iteration=0;
While( iteration< t )
    Select two chromosomes based on probability $p_i$
    Apply cross over operator
        Apply mutation operator
        Perform anonymization of new chromosomes
    Find the privacy fitness of new chromosomes based on $T$
    Update population $I$
    Store the best chromosome $S_b$
    iteration = iteration + 1
EndWhile
    Return $S_b$
End
```

Figure 4 Pseudo code of CPGEN algorithm.

Dataset description: Adult dataset is otherwise called as, "Census Income" dataset (Adult Data Set, 1996). This data was extracted from the census bureau database. This database consists of 48,842 instances and 14 attributes including both categorical and integer attributes namely, age, workclass, fnlwgt, education, education-num, marital-status, occupation, relationship, race, sex, capital-gain, capital-loss, hours-per-week, native-country. Irish dataset (Ayala-Rivera et al., 2014) is synthetically generated using Benerator. This dataset was created by using the frequency count distributions from the Irish Census 2011.

Evaluation metrics: The performance of the proposed algorithm through utility is evaluated through generalized information loss and average equivalence class size metric (Ayala-Rivera et al., 2014) is used to ensure the privacy of the proposed method.

5.2. Performance analysis

Census income dataset: The performance of the proposed algorithm is analyzed with the help of generalized information loss and average equivalence class size metric. Table 4 shows the performance analysis for various values of alpha and beta. The better performance is to obtain the minimum value for both the metrics to minimize the information loss and maximize the gain. For two different set of alpha and beta values, *GenILoss* is same by achieving the value of 0.4529 and 0.6734 for c value of 0.2 and 0.35 when alpha and beta is fixed as 0.4, and 0.6. But, in the case of C_{AVG}, the better value of 0.2104 is achieved when $\alpha = 0.6$ and $\beta = 0.4$.

Table 5 shows the analysis of number of population versus c value. For n value of four, *GenILoss* and C_{AVG} behave similarly. This shows that the number of chromosomes does not affect the performance of the proposed algorithm. When ana-

Table 4 Analysis of weightage constants.

	GenILoss		C_{AVG}	
	$\alpha = 0.4$, $\beta = 0.6$	$\alpha = 0.6$, $\beta = 0.4$	$\alpha = 0.4$, $\beta = 0.6$	$\alpha = 0.6$, $\beta = 0.4$
$C = 0.2$	0.4529	0.4529	1.25	0.3125
$C = 0.25$	0.4529	0.4529	1.002	0.2505
$C = 0.3$	0.4529	0.4529	0.98	0.2104
$C = 0.35$	0.6734	0.6734	0.625	0.2104

Table 5 Analysis of number of population.

	GenILoss		C_{AVG}	
	$n = 4$	$n = 8$	$n = 4$	$n = 8$
$C = 0.2$	0.4529	0.4529	1.25	1.25
$C = 0.25$	0.4529	0.4529	1.002	1.002
$C = 0.3$	0.4529	0.4529	0.98	0.98
$C = 0.35$	0.6804	0.6804	0.625	0.625

Table 6 Low values of c.

	GenILoss	C_{AVG}
$C = 0.001$	0.4529	104.45
$C = 0.002$	0.4529	101.76
$C = 0.003$	0.4529	96.15
$C = 0.004$	0.4529	76.49

Table 7 Analysis of weightage constants.

	GenILoss		C_{AVG}	
	$\alpha = 0.4$, $\beta = 0.6$	$\alpha = 0.6$, $\beta = 0.4$	$\alpha = 0.4$, $\beta = 0.6$	$\alpha = 0.6$, $\beta = 0.4$
$C = 0.2$	0.5	0.5	5.2	4
$C = 0.25$	0.5132	0.52	4	3.2
$C = 0.3$	0.5132	0.52	3.2	2
$C = 0.35$	0.52	0.52	1.2	1.2

Table 8 Analysis of number of population.

	GenILoss		C_{AVG}	
	$n = 4$	$n = 8$	$n = 4$	$n = 8$
$C = 0.2$	0.5	0.5132	5.2	5.2
$C = 0.25$	0.5132	0.5132	4	4.2
$C = 0.3$	0.52	0.52	3.2	4
$C = 0.35$	0.52	0.52	2	1.2

Table 9 Low values of c.

	GenILoss	C_{AVG}
$C = 0.001$	0.5	98.1
$C = 0.002$	0.5132	80.2
$C = 0.003$	0.52	40.1
$C = 0.004$	0.55	10.5

lyzing the Table 5, the better performance is achieved when c is set to 0.2 by reaching the value of 0.4529 and 1.25 for *GenILoss* and C_{AVG} respectively. From the above analysis, we set $\alpha = 0.6$, $\beta = 0.4$ and n value as four and the experimentation is done with the lower values of c. Table 6 shows the values reached by the proposed algorithm for lower values of c. Here, *GenILoss* is almost constant but the C_{AVG} value is changing frequently. For $C = 0.001$, the proposed algorithm obtained the value of 104.45 as C_{AVG}.

Irish dataset: Table 7 shows the analysis of weightage constants for the proposed algorithm. Here, we have fixed two different values for the alpha and beta. Then, C is varying from 0. l2 to 0.35 for performance analysis. Here, the better performance in terms of *GenILoss* is achieved when the C is fixed to 0.2 and the alpha and beta are fixed to 0.4 and 0.6 respectively. Similarly, the better performance of 1.2 is achieved for C_{AVG} when the alpha and beta is fixed to 0.4 and 0.6 for the C value of 0.35. Table 8 shows the performance analysis for varying number of population. Here, the better performance in terms of *GenILoss* is achieved when the C value is fixed to 0.2. From Table 9, the value of C is increased; the performance is decreased in terms of *GenILoss*.

5.3. Comparative analysis

The comparative analysis is also performed with the existing algorithm (Goryczka et al., 2014) and the values are shown in Table 10 for census income dataset. Here, genetic algorithm is executed 100 times and the average performance is computed. From Table 10, we understand that the proposed algorithm obtained the minimum value of 0.4529 for *GenILoss* when c value is from 0.2 to 0.3. When the value of c is fixed as 0.35, the proposed algorithm obtained the value of 0.6734 as *GenILoss* but the existing algorithm reached the higher value of 0.6734. When analyzing the performance of both algorithms using C_{AVG}, the proposed algorithm obtained the value of 1.25 and existing algorithm achieved the value of 5 when c is fixed as 0.2. So, the proposed algorithm obtained the best performance for all the values of c parameters.

Table 11 shows the effectiveness of the proposed algorithm on Irish dataset. Here, the values of C are varied from 0.2 to 0.35 and the performance is plotted. For all the values of C, the proposed algorithm outperformed the existing algorithm in term of *GenILoss*. The better performance achieved by the proposed algorithm in terms of *GenILoss* is 0.45 but the existing algorithm obtained only 0.48. Similarly, in terms of C_{AVG}, the better performance of 78.8 is achieved by the proposed algorithm. Again, the computation time required by the proposed algorithm is only 80 s which is less than the existing algorithm which required 92 s. Tables 12 and 13 shows the performance analysis of both the algorithms using variance. The output generated by the genetic algorithm (executed for 100 times) is then used to find the variance of those measurements. From Tables 12 and 13; we proved that the variance is less for the proposed algorithm when compared with the existing algorithm in all the three measures considered.

Table 10 Effectiveness analysis on census income dataset (Mean).

	GenILoss		C_{AVG}		Time	
	Proposed	Existing	Proposed	Existing	Proposed	Existing
$C = 0.2$	0.4529	0.6804	1.25	5	82	85
$C = 0.25$	0.4529	0.6804	1.002	4.008	74	76
$C = 0.3$	0.4529	0.6804	0.98	3.3374	78	79
$C = 0.35$	0.6734	0.6804	0.625	2.5	65	68

Table 11 Effectiveness analysis on Irish dataset (Mean).

	GenILoss		C_{AVG}		Time (in sec)	
	Proposed	Existing	Proposed	Existing	Proposed	Existing
$C = 0.2$	0.5	0.55	10.5	10.8	98	99
$C = 0.25$	0.45	0.5	10.5	10.8	96	98
$C = 0.3$	0.45	0.5	8.1	8.9	89	95
$C = 0.35$	0.45	0.48	7.8	8.9	80	92

Table 12 Effectiveness analysis on census income dataset (Variance).

	GenILoss		C_{AVG}		Time	
	Proposed	Existing	Proposed	Existing	Proposed	Existing
$C = 0.2$	0.05	0.059	0.25	1.2	15	21
$C = 0.25$	0.09	0.094	0.3	0.4	14	20
$C = 0.3$	0.029	0.03	0.18	0.25	15	18
$C = 0.35$	0.03	0.04	0.25	0.18	18	17

Table 13 Effectiveness analysis in Irish dataset (Variance).

	GenILoss		C_{AVG}		Time (s)	
	Proposed	Existing	Proposed	Existing	Proposed	Existing
$C = 0.2$	0.09	0.1	0.4	0.38	19	25
$C = 0.25$	0.07	0.08	0.45	0.41	18	25
$C = 0.3$	0.08	0.09	0.3	0.38	15	26
$C = 0.35$	0.07	0.09	0.28	0.31	14	24

6. Conclusion

This paper presented a new privacy measure, called c-mixture for collaborative data publishing problem. Here, we considered a new kind of "cold start problem" which occur when the data distribution is not same for multiple data providers. In order to handle this new type of problem in collaborative data publishing, a new measure called, c-mixture is proposed. This new measure considered the multiple privacy constraints into a single formula by considering the relative strength. Then, an algorithm called, CPGEN is developed using genetic algorithm and multi-objective constraints. The multi-objective optimization function considered the multiple privacy constraints along with the utility measurement to ensure high utility and privacy. The proposed algorithm is extensively analyzed using generalized information loss and average equivalence class size metric and the performance is compared with existing algorithm to prove the better or comparable utility and privacy than previous algorithms. The proposed CPGEN algorithm can be further enhanced to reduce the constraints defined in the objective function without affecting the privacy and utility.

References

Adult Data Set, 1996. from <https://archive.ics.uci.edu/ml/datasets/Adult>.

Ayala-Rivera, McDonagh, Patrick, Cerqueus, Thomas, Murphy, Liam, 2014. A systematic comparison and evaluation of k-anonymization algorithms for practitioners. Trans. Data Privacy 7 (3), 337–370.

Byun, J.W., Sohn, Y., Bertino, E., Li, N., 2006. Secure Anonymization for Incremental Datasets. Secure Data Management, SDM, p. 4863.

Byun, J.W., Kamra, A., Bertino, E., Li, N., 2007. Efficient k-anonymization using clustering techniques. Adv. Databases 4443, 188–200.

Chaytor, R., Wang, K., 2010. Small domain randomization: same privacy, more utility. In: VLDB, pp. 608–618.

Clifton, C., Tassa, T., 2013. On syntactic anonymity and differential privacy. Trans. Data Privacy 6 (2), 161–183.

EnamulKabir, Md., Wang, Hua, Bertino, Elisa, 2011. Efficient systematic clustering method for k-anonymization. Acta Inf. 48 (1), 51–66.

Fouad, Mohamed R., KhaledElbassioni, Bertino, Elisa, 2014. A supermodularity-based differential privacy preserving algorithm for data anonymization. IEEE Trans. Knowl. Data Eng. 26 (7), 1591–1601.

Fu, Ada Wai-Chee, Wang, Ke, Wong, Raymond Chi-Wing, Wang, Jia, Jiang, Minhao, 2014. Small sum privacy and large sum utility in data publishing. J. Biomed. Inform. 50, 20–31.

Fung, B.C.M., Wang, K., Yu, P.S., 2007. Anonymizing classification data for privacy preservation. IEEE Trans. Knowl. Data Eng. (TKDE) 19 (5), 711–725.

Ghasemzadeh, M., Fung, B.C.M., Chen, R., Awasthi, A., 2014. Anonymizing trajectory data for passenger flow analysis. Transp. Res. Part C: Emerg. Technol. (TRC) 39, 63–79.

Goryczka, Slawomir, Xiong, Li, Sunderam, Vaidy, 2013. Secure multiparty aggregation with differential privacy: a comparative study. In: Proceedings of the Joint EDBT/ICDT 2013 Workshops, pp. 155–163.

Goryczka, Slawomir, Xiong, Li, Fung, Benjamin C.M., 2014. m-privacy for collaborative data publishing. IEEE Trans. Knowl. Data Eng. 26 (10), 1–14.

HussainKhokhar, Rashid, Chen, Rui, Benjamin, C.M., Fung, C., Lui, Siu Man, 2014. Quantifying the costs and benefits of privacy-preserving health data publishing. J. Biomed. Inform. 50, 107–121.

Jana, Assema, Joshi, Shubham, 2014. Enhanced m-privacy for collaborative data publishing. Int. J. Innov. Res. Comput. Commun. Eng. 2 (11), 6590–6594.

Li, T., Li, N., 2007. t-closeness: privacy beyond k-anonymity and l-diversity. In: Proceedings of the IEEE International Conference on Data Engineering (ICDE), pp. 106–115.

Li, T., Li, N., 2008. Injector: mining background knowledge for data anonymization. In: Proceedings of International Conference on Data Engineering (ICDE), pp. 446–455.

Li, Tiancheng, Li, Ninghui, Zhang, Jian, Molloy, Ian, 2012. Slicing: a new approach for privacy preserving data publishing. IEEE Trans. Knowl. Data Eng. 24 (3), 561–574.

Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M., 2006. l-diversity: privacy beyond k-anonymity. In: Proceedings of the IEEE International Conference on Data Engineering (ICDE), p. 24.

Machanavajjhala, Ashwin, Gehrke, Johannes, Kifer, Daniel, Venkitasubramaniam, Muthuramakrishnan, 2007. l-diversity: privacy beyond k-anonymity. ACM Trans. Knowl. Discov. Data (TKDD) 1 (1).

McCall, John, 2005. Genetic algorithms for modelling and optimisation. J. Comput. Appl. Math. 184 (1), 205–222.

Samarati, P., Sweeney, L., 1998. Data to provide anonymity when disclosing information. In: Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), p. 188.

Soria-Comas, J., Domingo-Ferrer, J., Sánchez, D., Martínez, S., 2013. Improving the utility of differentially private data releases via k-anonymity. In: 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, pp. 371–379.

Sun, Xiaoxun, Li, Min, Wang, Hua, 2011. A family of enhanced (L,a)-diversity models for privacy preserving data publishing. Future Gen. Comput. Syst. 27 (3), 348–356.

Wang, K., Xu, Y., Fu, A.W.C., Wong, R.C.W., 2009. FF-anonymity: when quasi-identifiers are missing. In: Proceedings of the 25th IEEE International Conference on Data Engineering (ICDE), pp. 1136–1139.

Xiao, X., Tao, Y., 2007. m-invariance: towards privacy preserving republication of dynamic datasets. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. SIGMOD, pp. 689–700.