International Journal of Scientific Engineering and Research (IJSER)
ISSN (Online): 2347-3878
Impact Factor (2020): 6.733

ATPM: Achieving t-Closeness using Particle Swarm Optimization and Movement of Record

Megha Bisht[1], Pranav Mangla[2], Sanya Saxena[3], Vartika Puri[4]

[1]Department of Computer Science and Engineering & Information, Technology, Jaypee Institute of Information Technology Noida, India meghaa.bishtt[at]gmail.com

[2]Department of Computer Science and Engineering & Information, Technology, Jaypee Institute of Information Technology Noida, India pranav3nov[at]gmail.com

[3]Department of Computer Science and Engineering & Information, Technology, Jaypee Institute of Information Technology Noida, India 1799sanya[at]gmail.com

[4]Department of Computer Science and Engineering & Information, Technology, Jaypee Institute of Information Technology Noida, India vartika.puri[at]jiit.ac.in

Abstract: ***The amount of data available to us is increasing day by day. Data which is available online, especially the data from medical institutions, super markets and government has many uses like future research and innovation of new technologies, but this data may contain certain confidential information about a person like diagnosis details, purchase history etc. Privacy Preserving Data Publishing (PPDP) is a technique including privacy models like, k-anonymity, l-diversity and t-closeness that provide us with a scheme to publish data online without compromising the privacy of an individual. The purpose of this paper is to achieve t-closeness in such a way that the disclosure risk of patient's data is minimized while maximizing the usefulness of the data for research purposes. This paper proposes an algorithm named as Achieving t-Closeness using Particle Swarm Optimization and Movement of record (ATPM) that achieves t-closeness in two phases. In the first phase, Particle Swarm Optimization is used to make equivalence classes. Second step involves movement of records from one equivalence class to other considering distribution of sensitive attributes in the equivalence classes to achieve t-closeness. The proposed work gives high privacy guarantee with low information loss.***

Keywords: Privacy Preserving Data Publishing, T-closeness, Particle Swarm Optimization, Sensitive attribute

1. Introduction

The world we live in is consumed in data. Today, vast amounts of data are available to public from almost all fields including, science, statistics, economics, medicine, social science etc. We can derive huge benefits from the availability of all this data. Education and science fields specifically gain a lot from it. To make the data available to all, it needs to be published online to carry out experiments, studies and research. Taking an instance as given in [12], in the year 2006, Netflix, which is a huge media services provider made available a database of a hundred million customer's movie ratings to the public for research purposes. Like this, the use cases and the opportunities opened due to access to all kinds of data are innumerable. Now, data publishing happens in phases. The first phase being data collection. The data publisher collects information from individuals, that is, information holders and organizes the data. The second phase is data publishing. In this phase, the publisher publishes the data and makes it available to the public which can then use it for various purposes.

Publishing the data as is will lead to many privacy issues as the data might contain sensitive information of individuals. For example, patient data may contain information like the patient's identification number, the patient's diagnosis, address, etc. If this data gets published, the privacy of the patients will be compromised. An attacker can easily use this information against the victim. Privacy Preserving Data Publishing is a technique that aims to achieve a solution to this problem. It provides methods to anonymize data while keeping the information loss to the minimum so that the data is useful and can be published without the fear of privacy breach of an individual. Many privacy preserving models have been proposed to ensure the privacy of data such as k-Anonymity [7, 8], l- diversity [9], a n d t-closeness [10]. In this paper we focus on one particular privacy model, that is, T- closeness. While working on privacy models, we classify attributes into two categories, sensitive and non-sensitive. The attacker should not be able to find out the sensitive attributes of an individual record. T- closeness ensures the distribution of sensitive attributes in an anonymized dataset should not lead to the unnecessary knowledge gain by an adversary by keeping the distribution of the attributes as close as to the original dataset.

In this paper we propose an algorithm based on Particle Swarm Optimization and movement of records (ATPM) to achieve k-anonymity and t- closeness ensuring minimum information loss and ideal data utility. This is done by first making optimum clusters using the PSO clustering algorithm. Clusters are formed such that the records in each cluster have maximum similarity on the basis of non-sensitive attributes so that minimum information loss should occur while achieving k- anonymity. In the next step the distribution of sensitive attributes is calculated. It includes evaluation of probability distribution of a particular attribute in each cluster as well as in the dataset as a whole. Based on this, each sensitive attribute is arranged in such a way that each cluster has a similar distribution of all the attributes. This is achieved by the

movement of records. During the movement process the similarity between the non- sensitive attributes of a cluster is also considered. This is done to obtain clusters which have maximum similarity on the basis of non-sensitive attributes while also having close distributions of each sensitive attribute. Thus, t-closeness is attained.

2. Background and Motivation

A. Particle Swarm Optimization

Particle swarm optimization (PSO) [11] aims to solve various functional optimization problems by enhancing a candidate solution after each iteration.

Definition 1 (Particle Swarm Optimization). Particle swarm optimization is a population-based stochastic method that helps with optimization problems. It is modelled after natural processes, such as the flocking of birds or the movement of schools of fish.

PSO can be defined as a population-based search algorithm which is initialized with a population of random solutions, known as particles. The position of the particle corresponds to a candidate solution of the optimization problem. During each iteration of the algorithm, each candidate solution is evaluated by the objective function which is the function that is being optimized. The fitness of each candidate solution is determined. Each candidate solution may be considered as a particle going through the fitness landscape finding the maximum or minimum of the objective function as desired. The PSO algorithm uses the objective function to find its candidate solutions, and operates upon the resultant fitness values. It remembers the best fitness value it has achieved so far during the operation of the algorithm, which is known as the individual best fitness. Finally, the best fitness value achieved among all particles in the swarm, called the global best fitness is maintained by the PSO algorithm, and also the candidate solution that achieved this fitness, which is known as the global best position. The PSO algorithm consists of just three simple steps, which are repeated again and again until the stopping condition is met (for example the number of iterations specified): firstly, it evaluates the fitness of each particle. Secondly, it updates individual and global best fitness and positions. Lastly, it updates the velocity and position of each particle. Fig. 1 shows the stepwise working of PSO algorithm.

B. K-Anonymity

Definition 2 (k-anonymity)**.** K-anonymity states that each record in the published dataset must be indistinguishable from at least k-1 other records in the dataset.

As in [1], K-anonymity ensures that if an attacker wants to retrieve the information of one particular individual, he will not be able to do so because there will be k-1 other records just like the victim's record and thus the privacy of the victim stays guarded. But the k-anonymity model has certain drawbacks. This model fails if the values of the sensitive attribute are the same in one equivalence class. For example, if there is a patient dataset that has 'Disease' as the sensitive attribute and 'Zip Code' and 'Age' as the non-sensitive attribute. Suppose the values of 'Disease' are the same, that is, 'Asthma' for all entries of an equivalence class. In this case the attacker will know for sure that any individual whose record lies in this class has asthma. This is known as the homogeneity attack which can be referenced from [13]. This is one of the most famous attacks on k-anonymity model that simply asserts that there is insufficient amount of heterogeneity in the formed classes which further leads to unveiling of critical information. K-anonymity will also fail if the attacker already has some background knowledge of the victim, this is called the background knowledge attack as explained in [13]. For example, if an attacker knows the victim's zip code and age, he can easily know the equivalence class in which the victim's record lies. The attacker can then infer a lot of information about the victim's disease from that equivalence class.

C. l-Diversity

To overcome the weaknesses of k-anonymity, a model known as the l-diversity model was proposed.

Definition 3 (l-diversity) It states that to guarantee privacy, the published dataset must be l-diverse i.e. each equivalence class in the dataset must have at least l distinct values of the sensitive attribute, which can be looked upon in [1].

The l -diversity model also fails in some cases. For instance, let say, we have a dataset in which the sensitive attribute can have values- A, B or C. In an equivalence class of ten records, there is one record of A and one record of B. The rest of the eight records have C as their sensitive attribute. This equivalence class is l-diverse where l=3 but the attacker knows that 80% of the records have C as the sensitive attribute. This is especially dangerous in a patient dataset. Suppose the sensitive attribute is 'Disease, and C is 'HIV AIDS', the attacker will thus know that any patient from this equivalence class has 80% probability of having HIV AIDS. This drawback is called probabilistic inference attack. L-diversity may also be compromised by the similarity attack. Suppose in an equivalence class with l=3, the sensitive attribute i.e. disease has values Lung Cancer', 'Pneumonia' and 'Asthma'. From this, the attacker can be 100% sure that any patient in this equivalence class has a lung disease

D. t-Closeness

To overcome the weaknesses of l-diversity a refinement was proposed as in [1], known as t-closeness.

Definition 4 (t-closeness). A dataset is said to have t-closeness if the difference between the distribution of the sensitive attribute in each equivalence class and the distribution of the sensitive attribute in the entire table is no more than a threshold t.

A dataset is t-close when all equivalence classes are t-close as well. It interrupts attribute disclosure that protects data privacy. It also protects against homogeneity and

background knowledge attacks mentioned in k-anonymity and identifies the semantic closeness of attributes, which is a limitation of l-diversity.

For example, Table I is the original data containing records of 3,000 individuals. Table II is an anonymized version of Table I. The sensitive attribute is Disease and there is a column called Count that indicates the number of patients that have a particular disease. The threshold value t is AIDS among the population in the data set is 700/3000 = 0.23.

From Table II we can see that the probability of HIV AIDS among patients in the first equivalence class is 300/600 = 0.5, in the second equivalence class it is 200/2000 = 0.1 and in the last equivalence class it is 200/400 = 0.5. Since the values of distribution of HIV AIDS for the three equivalence classes are 0.5, 0.1 and 0.5. The distribution in first and third cluster > t (0.3), thus, anonymized table, Table II is not 0.3 - close. In the original dataset, the distribution of sensitive attribute was 23% whereas in an anonymized version this has become 50% in the first and last cluster, so if an adversary locates the individual in first or third cluster, he/ she will have 50% chances of being identified with the sensitive disease HIV AIDS. By achieving t-closeness, we want to achieve a stage where we safeguard the sensitive information about a particular individual from the adversary while we allow a researcher to learn information about a large population.

Table 1: Original Patients Table

S.No.	ZIP Code	Age	Disease	Count
1	51273	29	HIV AIDS	100
2	51274	21	Asthma	100
3	51205	25	HIV AIDS	200
4	51202	23	Asthma	200
5	51505	43	HIV AIDS	100
6	51504	48	Asthma	900
7	51506	47	HIV AIDS	100
8	51507	41	Asthma	900
9	51203	34	HIV AIDS	100
10	51205	30	Asthma	100
11	51202	36	HIV AIDS	100
12	51207	32	Asthma	100

Table 2: Anonymized Version of Table 1 Violating 0.3 Closeness

S.No.	ZIP Code	Age	Disease	Count
1	512**	2*	HIV AIDS	300
2	512**	2*		300
3	515**	4*	HIV AIDS	200
4	515**	4*	Asthma	1800
5	512**	3*	HIV AIDS	200
6	512**	3*	Asthma	200

3. Related Works

In the past various methods have been employed to achieve t-closeness. One of the ways is to apply the constraints of t-closeness to the existing k-anonymity algorithms that use generalization and suppression to achieve anonymity. For example - Incognito algorithm as given in [1, 20] and Mondrian algorithm explained in [2, 20] can be applied with the t- closeness constraint to make the table t-close.

Unlike earlier known methods of generalization and suppression, micro-aggregation has been used to achieve t-closeness [3]. Micro-aggregation is a technique to limit disclosure. It is aimed at protecting the privacy of data subjects in microdata releases. Two algorithms are proposed in this paper. In the first one, each cluster is first made to satisfy k-anonymity and after this process, t-closeness is achieved. In the second one, each cluster is made to satisfy t-closeness and k-anonymity simultaneously.
A privacy measure inspired from t-closeness is defined by Rebollo-Monedero et al. [4] which is achieved using the technique of perturbation unlike existing methods of generalization and suppression.

An anonymization algorithm called SABRE [6], a Sensitive Attribute Bucketization and Redistribution framework for t-closeness is introduced. SABRE first partitions a table into buckets of similar sensitive attribute values and then redistributes the tuples of each bucket into equivalence class that are created dynamically. While redistributing, the buckets and the records from each bucket that are included in an equivalence class are chosen keeping t-closeness in mind.

In recent years, a tool – ARX tool has come into existence which is anonymization software that can be employed for implementation of numerous privacy methods in a supremely efficient fashion. Few of the recent research work done on the mentioned tool have been documented in [16]. Introducing a convenient approach towards t-closeness in [18], multiple sensitive attributes are used to calculate the value of 't'. This method helps to reduce the needless anonymization of data beyond need. A novel method, TCS (t-Closeness Slicing) is introduced in [19] where a vertical partition of given dataset A is coupled with a horizontal partition in a manner that all subsets of A fulfil the criteria for t closeness.

In [21], an improvement over the existing sanitization technique is shared, which hides raw information presented by users. It involves optimally generating a key using a novel Particle Swarm Velocity aided GWO (PSV-GWO) algorithm.

In this paper, we use an optimization method - Particle Swarm Optimization to give us appropriate clusters of our dataset. The distribution of sensitive attribute in each cluster is calculated. Further, we use a record-swapping method to swap the records between clusters, a source cluster and a destination cluster. The destination cluster is chosen in such a way that the difference in the Quasi-identifiers in [1] of the record to be swapped and the records in the destination cluster is minimal.

4. Detailed Design

The proposed algorithm achieves t-closeness in two steps-

- In the first step, Particle Swarm Optimization on non-

sensitive attributes is performed to obtain optimal clusters. Optimal clusters mean that every data point in a particular cluster shares the maximum similarity on the basis of non- sensitive attributes. In the dataset users in this paper, the non-sensitive attributes are age, gender and zip code.

- In the second step, distribution of sensitive attributes is calculated in each cluster and a threshold value is chosen. Movement of records is performed in each cluster until the t-closeness property is satisfied.

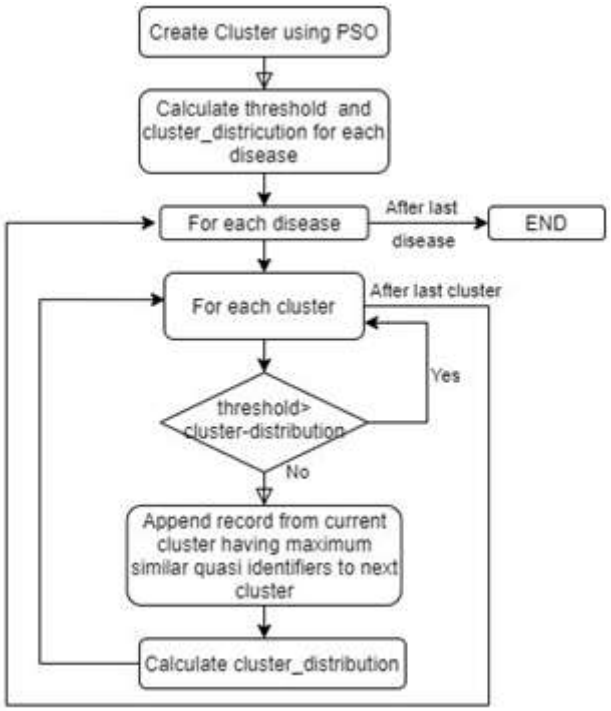

Figure 1: Flow chart of proposed design

A. Using PSO To Form Optimal Clusters

PSO algorithm that has been explained earlier can also be used to form clusters as explained in [14, 17]. Each particle represents a position in Nd dimensional space. The particle then moves through this multi-dimensional search space, adjusting its position toward both the particle's best position found thus far and the best position in the neighbourhood of that particle. Each particle i maintains the following information:

- xi: The current position of the particle
- vi: The current velocity of the particle
- yi: The personal best position of the particle

Using the above notation, a particle's position is adjusted according to:

$$v_{i,k}(t+1) = wv_{i,k}(t) + c1r1_{,k}(t)\,(y_{i,k}(t) - x_{i,k}(t)) + c2r2_{,k}(t)(\hat{y}k(t) - x_{i,k}(t)) \ldots\ldots(1)$$

[14, eq. (3)]

$$x_i(t+1) = x_i(t) + v_i\,(t+1)\,) \ldots\ldots\ldots(2)$$

[14, eq. (4)]

Where w is the inertia weight, c1 and c2 are the acceleration constants,

r1,j(t), r2,j(t) ~ U(0,1) and k = 1,….., Nd.

In the sense of PSO clustering, a single particle represents the Nc cluster centroid vectors. Each particle x_i is constructed as:

$x_i = (m_i1, \ldots\ldots\ldots, m_iNc) \ldots\ldots\ldots(3)$

[14, eq. (7)]

where mij refers to the j^{th} cluster centroid vector of the i^{th} particle in the cluster Cij. Thus, a swarm represents a number of candidates clustering for the current data vectors. The fitness of particles is measured as the quantization error,

$$Je = \frac{\sum_{j=1}^{Nc}[\sum_{\forall zp \in C_{ij}} d(Z_p, m_j)/|C_{ij}|]}{N_c} \ldots\ldots\ldots(4)$$

[14, eq. (8)]

where d is the distance to the centroid and is defined as:

$$d(Z_p, m_i) = \sqrt{\sum_{k=1}^{Nc}(Z_{pk} - m_{jk})^2} \quad \ldots\ldots\ldots(5)$$

[14, eq. (1)]

and |Cij| is the number of data vectors belonging to cluster Cij i.e. the frequency of that cluster. Therefore, we can say that the quantization error is our objective function.

Using the standard PSO, data vectors can be clustered using Algorithm 1 as explained in [14].

For each iteration of PSO, the data points are evaluated according to the objective function and with each iteration, better clusters are formed. After PSO runs for the specified number of iterations, we get optimal clusters, that is, clusters whose data points have maximum similarity based on their non-sensitive attributes.

B. Movement of Records

Once the clusters are formed by PSO algorithm, next step is to distribute the sensitive attributes in each cluster such that no one cluster has high concentration of one sensitive attribute that might lead to breach of individual's privacy. For this, the distribution of the sensitive attribute is calculated in the whole dataset and also in each cluster. The distribution of a sensitive attribute in a cluster which we called as item_distribution will be equal to the number of records having that particular attribute in the cluster divided by the total number of records in the cluster.

$$item\_distribution = \frac{\text{No of records having the sensitive attribute in cluster}}{\text{Total no of records in the cluster}} \quad \ldots\ldots(6)$$

The distribution of a sensitive attribute in the overall dataset (dataset_distribution) is equal to the number of records having that particular sensitive attribute in the entire dataset divided by the total number of records in the entire dataset.

$$dataset\_distribution = \frac{\text{No of records having the sensitive attribute in dataset}}{\text{Total no of records in the dataset}} \quad \ldots\ldots..(7)$$

According to the t-closeness property, we have to ensure that theitem_distribution (6) of a sensitive attribute in each cluster is close to its dataset_distribution (7) i.e. the difference between the item_distribution (6) of a disease in each cluster and its dataset_distribution (7) is less than a threshold value t which is defined formally in equation (8).

$$abs(item\_distribution - dataset\_distribution) \leq t \ldots\ldots.. (8)$$

Suppose, the difference between the item_distribution (6) calculated for a disease in a cluster (A) and the dataset_distribution (7) is greater than the threshold value t. In this case, the records in the cluster need to be move from one cluster to other to achieve t-closeness property. For movement of records, we select those records from A whose non-sensitive attributes are the most similar to the non-sensitive attributes of the records in the next cluster (B).

We then delete the selected records from A and move them to B. Now, the item_distribution (6) is calculated again for both A as well as B and is updated. We now move on to the next cluster. This cluster becomes our new 'A' cluster. The entire process mentioned above is carried out for the new 'A'. This process continues iteratively for all the clusters again and again until the distance between the item_distribution and dataset_distribution of a sensitive value becomes less than the threshold value in each cluster. This process repeats for each sensitive value in the dataset. Algorithm 2 shows the steps of achieving t-closeness.

Algorithm 1: PSO Cluster Algorithm

1. Initialize each particle to contain N_C randomly selected cluster centroids.
2. For f = 1 to f_{max}, do
   (a) For each particle i do
   (b) For each data vector z_p

i. calculate the Euclidean distance d(z_p, m_{ij}) to all cluster centroids C_{ij}

ii. assign z_p to cluster C_{ij} such that d(z_p, m_{ij}) =min∀c=1,….,Nc{d(z_p, m_{ic})}

iii. calculate fitness using equation (4)

   (c) Update the global best and local best positions.
   (d) Update the centroid using equations (1) and (2).Here, f_{max} is the maximum number of iterations

Algorithm 2: Achieving **T**-Closeness using **P**SO and Movements of records (ATPM)

1. Form clusters using PSO clustering algorithm.
2. For each sensitive values:
3. Calculate distribution of the sensitive value in the entire dataset i.e. dataset distribution (7) and select threshold value.
4. Calculate distribution of the sensitive value in each cluster i.e. item distribution (6).
5. For each cluster check if

Abs (item distribution – dataset distribution) <threshold value (8).

6. If yes, check for the next cluster.
7. If no, select a record from the current cluster having the highly sensitive values and maximum similar Quasi- identifiers with the next cluster.
8. Append the selected record to the next cluster and delete from the current cluster.
9. Calculate the item_distribution (6) of the current cluster and the next cluster again.
10. Repeat steps 5-9 till an optimum item_distribution (6) is achieved for each cluster.

Repeat steps 2-10 until t-closeness is achieved for all sensitive values in the dataset.

5. Implementation & Results

The proposed approach has been implemented in python in the system with Intel Core™ i3 processor, 1 GB RAM. We have tested the approach in a patient dataset[1] which has 30,000 patient entries. Fig 3. shows the snapshot of the dataset. We considered Age, Sex and Zip attributes as a Quasi-identifiers and disease as a sensitive attribute in this dataset.

The goal of this paper is to make the difference between the distribution of 'Disease' in each cluster and its distribution in the entire dataset less than a specific threshold value.

A. Global Best Value

The global best values obtained after obtaining clusters from particle swarm optimization algorithms signifies the closeness of records in a particular cluster on the basis of non-sensitive attributes, that is, AGE, SEX, ZIP in our case. The decreasing g-best score signifies the increase in similarity score for our data points in a particular cluster after each iteration. Fig. 4 shows the g-best scores obtained after 100 iterations.

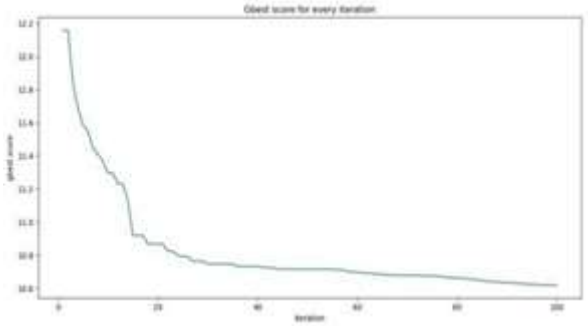

Figure 4: The Global Best Values Obtained

B. Distribution Of Sensitive Disease

The total five clusters have been obtained. We calculated the item_distibution (6) for each disease in each cluster. We have here shown the item_distibution we achieved for the disease 'Thyroid Disorders'. The dataset_distribution (7) for disease 'Thyroid Disorders' was calculated to be 0.18

[1]https://health.data.ny.gov/Health/Hospital-Inpatient-Discharges-SPARCS-De-Identified/82xm-y6g8

Table 3: Distributions of 'Thyroid Disorders' in Each Cluster after the Application of Algorithm 1 and 2 Respectively

Cluster No.	Values After Application of Algorithm 1	Values After Application of Algorithm 2
0	0.4333	0.19047
1	0.2	0.15789
2	0.0769	0.14285
3	0.0	0.16
4	0.0	0.15789

The values of distribution of the disease in each cluster after application of algorithm 1 in Table III show that the distribution of 'Thyroid Disorders' in all clusters is diverse. The goal is to form clusters in such a way that these values of item_distribution (6) and the dataset_distribution, that is, 0.18 are almost equal.

Values after application of Algorithm 2 in Table III shows the distributions of 'Thyroid Disorders' in each cluster after the movements of records. Each of these values is almost equal to the value of dataset_distribution (7) that is 0.18. This can be seen in Table IV. Thus, t-closeness has been achieved as the distribution of an individual diseases in each cluster has become almost equal to their distribution in the entire dataset.

Table 4: Difference between Distributions of 'Thyroid Disorders' in Each Cluster and the Entire Dataset

Item_distribution	Dataset_distribution	Difference
0.190476	0.18	0.010476
0.157894	0.18	0.022106
0.142857	0.18	0.037143
0.16	0.18	0.02
0.157894	0.18	0.022106

6. Performance Analysis

A. Cluster Analysis Using Silhouette Index

Silhouette analysis can be considered as a method of

interpretation of consistency within clusters of data. It indicates a measure of how similar an object is to its own cluster compared to other clusters. It can be used to study the separation distance between the resulting clusters.

$$S = \frac{b-a}{\max(a,b)} \ldots\ldots. (9)$$

The Silhouette Coefficient is defined for each sample and is composed of two scores: a: The mean distance between a sample and all other points in the same cluster. b: The mean distance between a sample and all other points in the next nearest cluster.

The Silhouette Coefficient for a set of samples is given as the mean of the Silhouette Coefficient for each sample. The score is bounded between -1 and +1, -1 indicating incorrect clustering and +1 for highly dense clustering. Scores around zero indicate overlapping clusters. A higher score represents dense and well separated clusters, which relates to a standard concept of a cluster.

1) Comparison of performance of ATPM using PSO and traditional k means algorithm

We compared the performance of the proposed ATPM method using PSO and traditional K Means algorithm. Table V shows that nature-based algorithm, PSO creates better distinguished clusters than K means as the silhouette index is nearer to 1. Post application of ATPM on the clusters formed by both algorithms, we see that PSO gives a better silhouette score.

So, clustering using globalized search method offered by PSO helps to enhance the performance of ATPM in comparison to the conventional clustering technique of K means.

2) Comparison of Quality of Clusters before and After Application of ATPM

When we form clusters based on non-sensitive attributes, we ensure that most similar entries are clustered together. Post this, we reorder records using ATPM which will lead to some distortion of the clusters initially formed.

Table V shows the values of silhouette index before and after application of ATPM on clusters formed by PSO, i.e., 0.684 and 0.513 respectively. As both the values are comparable, we can say that ATPM doesn't hugely affect the quality of clusters, in terms of how well separated the clusters are while also achieving t closeness. Therefore, we achieve both, the similarity of records in a cluster based on non-sensitive attributes and a good level of dissimilarity between records in a cluster based on sensitive attributes, thus, achieving t closeness.

Table 5: Silhouette Indices

S.NO.	After Clustering	After applying ATPM
Particle Swarm Optimization	0.684	0.513
K Means	0.643	0.481

7. Conclusion

The privacy models like k-anonymity and l-diversity are not sufficient for privacy protection in published data because these models don't consider the distribution of sensitive attributes. This makes the t-closeness model very crucial. The proposed model, termed ATPM presents an effective method to maintain a good level of data privacy without affecting or reducing the value of the data published. In the past, various attempts have been made to achieve t-closeness, but in those the clusters were first made and then were refined in later stages. In this study, we first use the PSO algorithm, thus ensuring that we get refined clusters in the beginning itself. The records in each cluster have maximum similarity based on the non-sensitive attributes. These refined clusters help in achieving better results. Our proposed algorithm was able to cluster the medical data in an effective way such that the similarity between the non-sensitive attributes in a cluster is maximized and the differences between the distribution of the sensitive attribute in each equivalence class and the distributions of the sensitive attribute in the entire table are less than threshold t. The numerical results also show that this model is efficient in safeguarding privacy while minimizing information loss.

In the future work we can extend our model to other known datasets like ADULT, CUPS etc. Looking at the impacts and results that the model gives when run on the mentioned datasets might provide newer insights and directions to further polish the proposed algorithm. Another direction is to test our algorithm for other nature inspired clustering techniques and analyse which technique give the most optimal results.

References

[1] N. Li, T. Li and S. Venkatasubramanian, "t-Closeness: Privacy Beyond k-Anonymity and l-Diversity," 2007 IEEE 23rd International Conference on Data Engineering, Istanbul, 2007, pp. 106-115, doi: 10.1109/ICDE.2007.367856.
[2] N. Li, T. Li and S. Venkatasubramanian, "Closeness: A New Privacy Measure for Data Publishing," in IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 7, pp. 943-956, July 2010, doi:10.1109/TKDE.2009.139.
[3] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez and S. Martínez, "t-Closeness through Microaggregation: Strict Privacy with Enhanced Utility Preservation," in IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 11, pp. 3098-3110, 1 Nov. 2015, doi: 10.1109/TKDE.2015.2435777.
[4] D. Rebollo-Monedero, J. Forné and J. Domingo-Ferrer, "From t-Closeness-Like Privacy to Postrandomization via Information Theory," in IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 11, pp. 1623-1636, Nov. 2010, doi: 10.1109/TKDE.2009.190.
[5] Vartika Puri, Shelly Sachdeva, Parmeet Kaur, "Privacy preserving publication of relational and transaction data: Survey on the anonymization of

patient data", vol 32, 2019.
[6] J. Cao, P. Karras, P. Kalnis, and K.-L. Tan. SABRE: a Sensitive Attribute Bucketization and REdistribution framework for t-closeness. The VLDB Journal, 20(1):59- 81, 2011.
[7] Sweeney, L. 2002. K-anonymity: A model for protecting privacy. Int. J. Uncertain. Fuzz., 10(5):557–570.
[8] P. Samarati, "Protecting respondents identities in microdata release," in IEEE Transactions on Knowledge and Data Engineering, vol. 13, no. 6, pp. 1010-1027, Nov.- Dec. 2001, doi: 10.1109/69.971193.
[9] Machanavajjhala, A., Gehrke, J., Kifer, D. and Venkitasubramaniam, M., 2006. l-diversity: Privacy beyond k-anonymity. In Proc. 22nd Intnl. Conf. Data Engg. (ICDE):24.
[10] N. Li, T. Li and S. Venkatasubramanian, "t-Closeness: Privacy Beyond k-Anonymity and l-Diversity," 2007 IEEE 23rd International Conference on Data Engineering, Istanbul, 2007, pp. 106-115, doi: 10.1109/ICDE.2007.367856.
[11] James Kennedy and Russell Eberhart, "Particle Swarm Optimization," in Proceedings of ICNN'95 - International Conference on Neural Networks,doi: 10.1109/ICNN.1995.488968.
[12] https://www.nytimes.com/2006/10/02/technology/02netflix.html
[13] Rajendran, Keerthana & Jayabalan, Manoj & Rana, Muhammad Ehsan. (2017). A Study on k-anonymity, l- diversity, and t-closeness Techniques focusing Medical Data. 17.
[14] D. W. van der Merwe and A. P. Engelbrecht, "Data clustering using particle swarm optimization," The 2003 Congress on Evolutionary Computation, 2003. CEC '03., Canberra, ACT, Australia, 2003, pp. 215-220 Vol.1, doi: 10.1109/CEC.2003.1299577
[15] Wang, Shuihua & Yang, Jianfei & Liu, Ge & Du, Sidan & Yan, Jie. (2016). Multi-objective path finding in stochastic networks using a biogeography-based optimization method. SIMULATION. 92. 10.1177/0037549715623847
[16] Churi, Prathamesh & Pawar, Dr. Ambika. (2019). A Systematic Review on Privacy Preserving Data Publishing Techniques. Journal of Engineering Science and Technology Review. 12. 17-25. 10.25103/jestr.126.03.
[17] Zhao M., Tang H., Guo J., Sun Y. (2014) Data Clustering Using Particle Swarm Optimization. In: Park J., Pan Y., Kim CS., Yang Y. (eds) Future Information Technology. Lecture Notes in Electrical Engineering, vol 309. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-55038-6_95
[18] Roy, Debaditya & Jena, Sanjay. (2013). Determining t in t-closeness using Multiple Sensitive Attributes. International Journal of Computer Applications. 70. 47-51. 10.5120/12179-8291.
[19] Wang, Mingzheng & Jiang, Zhengrui & Zhang, Yu & Yang, Haifang. (2018). T-Closeness Slicing: A New Privacy Preserving Approach for Transactional Data Publishing. INFORMS Journal on Computing. 30. 10.1287/ijoc.2017.0791.
[20] Ayala-Rivera, Vanessa & Mcdonagh, Patrick & Cerqueus, Thomas & Murphy, Liam. (2014). A Systematic Comparison and Evaluation of k-Anonymization Algorithms for Practitioners. Transactions on Data Privacy. 7. 337-370.
[21] Mandala, J. and M. Rao. "PSV-GWO: Particle Swarm Velocity AidedGWO for Privacy Preservation of Data." (2019)