Sensor Data Anonymization Based on Genetic Algorithm Clustering with L-Diversity

Ainur Abdrashitov, Anton Spivak
ITMO University
Saint-Petersburg, Russia
{abdrashitovainur, anton.spivak}@gmail.com

***Abstract*—The collecting of digital information by various organizations is producing significant volume of data. Processing by third-party companies is requiring data to be published. Published data in its initial form typically contains sensitive information about individuals. One of ways to preserve privacy level of data and save it useful is anonymization. The paper describes a method of anonymization based on genetic algorithm clustering. It uses k-anonymity and l-diversity as privacy models which are implemented in the method. Base operators of genetic algorithm are modified to satisfy the optimization problem conditions. The experimental study focuses on investigation method application area and defines the ways of future improvement.**

I. INTRODUCTION

Sensor data which produced by different kind of equipment can have several characteristics important to processing. The first one is that it can have big volume and require special methods to analyze. Another big deal that this data have to be handled with different analytics type to extract meaningful information. There is privacy risk to identify exact person based on sensor information [2]. Because the sensors spread too wide areas of people activity ranging from entertainment devices to medicine equipment. Many of this collected data have to be saved and processed by third-party organizations. Some of them focus on gain in marketing field. Another ones attempt to extract new knowledge from data. All data subject can be divided to data owner and who is processing this data. Often data owner which obtains data by sensors does not trust to data consumer. The case is not unique in processing area of computer science and sensor data especially it originates from medicine field where the association between data and person is highly confidential.

To avoid the identification of records in data, uniquely identifying information like names and social security numbers are removed from the tables. However, this first sanitization still does not ensure the privacy of individuals in the data.

Paper reviews the way of privacy providing by search the optimal distribution of values in data set.

We implemented a genetic algorithm with improved operators for solving the data anonymization problem. This paper describes the algorithm and solutions applied for its improvement.

II. RELATED WORK

Last years some methods have been developed to solve problem of distrust relations between data producer and consumer with possibility to process data [3]. To improve this type of relations third subject of relations is added which called publisher. It is who transfers data from producer to consumer and optionally changes it value and characteristics according to defined security requirements. In common organization which processes data, data is located in untrusted area. The publisher function is to change data to prevent consumer to violate privacy of producer. This paper mainly focused on problem of identity between data which consumer handles by some processing methods and exact person which used sensors to collect data. The examples of when it is critical of identify can be found in different area: geolocation of people or particular subject, person with medicine equipment, facts from the private life. Due to evolution of internet of things the problem of identify carrier of sensor or set of sensors will increase in future. The information which can be acquired from data is significant to related area of research. The process of providing privacy on dedicated level usually called anonymization [4]. It performs some actions on data by publisher such as clearing explicit identifier and defining quasi identifiers [4] which values should be changed according with some privacy model. The anonymization operations are generalization, suppression, anatomization, permutation and perturbation. Each of them modifies data by different type of change: structure, clear values, change values by taxonomy, mix values. A method of anonymization uses set of operations to reach desire level of privacy.

The problem of sensor data anonymization methods is close to data anonymization. In fact sensor data has some features which add new characteristic on technical level. That Big Data manner imposes restriction that we could not use in anonymization process on all data due to significant volume. Data has the flat view is not managed by database engine. Although some of traditional methods of anonymization can be used with modification.

In [5] authors described the method of k-anonymization based on finding all possible k-anonymous full-domain generalization operations. It uses the generalization property and assumes rollup value of parent qid consists of child of qid. It allows to have base to search criteria in generalization space and find all generalization qid meet an optimal k-anonymization.

The hybrid method is suggested by Lin in [6] through process of anonymization leads not only to generalization but

attempts to save information in resulted data. It combines two methods: OKA [7] and k-member algorithm [8]. Each part of algorithm deals with own aim. OKA directs to reduce the total information lost during anonymization and k-member locally reduces loss among qid group.

Mostly close to method of this paper is work described in [9]. It uses genetic algorithm [10] based approach to find optimal k-anonymization. The authors apply assigned-oriented method to solve anonymization problem. The implementation of method uses all population of chromosome to code whole solution and chromosome as a partitioning to clusters. It is absolutely different from approach used in current work. The genetic evolution evaluates the population based on constraint of k-anonymity which reflected of coding chromosome and minimizing information distortion.

III. Problem statement

The method suggested in paper is based on genetic algorithm as a commonly used approach to search optimal solution in NP-hard set of problems [1]. An optimal k-anonymization applies to this kind of tasks class. Unlike other works in the area of preserving privacy problem this paper focuses on anonymization of sensors data. Additionally to k-anonymity privacy model the described method includes l-diversity privacy model to prevent the record linkage attack type. The method does not operate in purpose of information distortion but it will be object for further study. The value of k and maximization of l-diversity is part of fitness function used during genetic algorithm evolution. As it is considered the sensors data the method operates in data block term. In the work data block size is strictly defined as 900 records. The aim of suggested method is to find the optimal solution for clustering data block to cluster size not less than k which meet requirements of *k*-anonymity and maximum of *l*-diversity value overall data blocks.

For $r_i \in R$, $i = \overline{1, ds}$. where *ds* is size of data block.
The task is to find partitioned $\boldsymbol{D} = \{D_1, D_2, D_3, \dots, D_{ds}\}$ where D_i is distinct cluster with a size of k. Method uses two conditions: $|D_i| = k$ and *l*-diversity value is maximal for all clusters on condition $l = \min\{l(D_i)\}$, i.e. *l*-diversity value counted as minimal value among all clusters.

The *l*-diversity value is using as its entropy representation. It is calculating per each D_i and $-\sum_{s \in S} D_i(s) \log D_i(s) \geq \log l$, where $D_i(s_i)$ is number of entries equal to s_i.

IV. Method implementation

The main parameter using for rate the data of reviewing algorithm is l-Diversity. Let's discuss issures of using this parameter.

- l-Diversity no longer requires knowledge of the full distribution of the sensitive and nonsensitive attributes.
- l-Diversity does not even require the data publisher to have as much information as the adversary. The parameter l protects against more knowledgeable adversaries; the larger the value of ℓ, the more information is needed to rule out possible values of the sensitive attribute.
- Instance-level knowledge (Bob's son tells Alice that Bob does not have diabetes) is automatically covered. It is treated as just another way of ruling out possible values of the sensitive attribute.
- Different adversaries can have different background knowledge leading to different inferences. L-Diversity simultaneously protects against all of them without the need for checking which inferences can be made with which levels of background knowledge.

Overall, we believe that ℓ-diversity is practical, easy to understand, and addresses the shortcomings of k-anonymity with respect to the background knowledge and homogeneity attacks.

As a main force of clusterization data to distinct clusters it suggests implementing genetic algorithm to this particular task. The genetic algorithm operates in own terms like gene, chromosome, population, mutation, crossover. Let's clarify each of them.

A. Data block, gene and chromosome

First of all method operates data block which consists of set of records. Each record is divided to sensitive value and qid identifiers. Gene of genetic algorithm represents sensitive value, number of record in data block and relation to cluster. The record qid values belonging to one cluster after finished evolution are subject of generalization process and suppression to apply k-anonymity.

Another important element of genetic algorithm is chromosome. In presented implementation it is all the genes (records) belonging to cluster. The algorithm manipulates the binding record and cluster through executing genetic operators mutation and crossover. A chromosome represents whole solution of distribution records per clusters. The population is set of chromosomes which are marked by algorithm and after it action of selection best of them performed. Selection action is repeated in each evolution cycle to search optimal distribution. Therefore population is set of candidates to be the best solution of distribution records per clusters. As additional structure supported through search process is set of clusters associated with each chromosome. The size of cluster strictly limits to k coefficient of k-anonymity. The constraint of k-anonymity is implemented by design of record manipulation.

The initialization of start state of chromosomes performed by gradually filling of clusters. The amount of cluster sample is defined as data block size divided by k.

B. Fitness function

The ranking of genetic algorithm is main search tool of entire research process. Usually it should help to mark desired solution to evolve in next generation genetic algorithm. Fitness function should solve this type of task with needed accuracy. It should provide quantative difference between existing solution

without knowledge of limit of evaluation them. Due to k-anonymity constraint implemented by design and not good candidate to realize valuably ranking as fitness function uses entropy l-diversity. It shows that at least l distinct values of sensitive attributes belong to one cluster. The aim of genetic algorithm is to distribute records per cluster while maximizing l-diversity. To calculate l-diversity it is needed to check all cluster and value of fitness function determined by minimum l-diversity of cluster of particular chromosome. The value of fitness function is used in selection process which based on Tournament approach to choose set of best chromosomes to next generation of evolution.

C. Mutation

The described algorithm has mutation operator which exchange selected genes in chromosome. It allows acting with weak impact to chromosome. There are modifications to mutation operator to direct process of search optimal solution. The selection of clusters to mutate its record based on *l*-diversity value. Each cluster associated with value that shows the probability of choosing this cluster in mutation. The cluster with higher *l*-diversity value less desirable to be chosen during mutation and opposite cluster with lower *l*-diversity is more expected in mutation. But all variants possible one more but with difference probability it allows speed up search finish solution. The mutation pseudo-code is shown below.

Algorithm 1 Mutation

0: PROCEDURE: **Mutation**
1: Input: Chromosome **Ch**, MutateRate **rate**, Cluster **Cl**
2: begin
3: *Find* **MutateNum** of mutation as **Ch.size** * **rate**
4: while **MutateNum** *is not* satisfied do
5: *Define* probability of chromosome cluster **CP**
6: **mutateCluster1**← *Select* from **Cl** using **CP**
7: **mutateCluster2**← *Select* from **Cl** using **CP**
8: **record1** ← Randomly *select* from **mutateCluster1**
9: **record2** ← Randomly *select* from **mutateCluster2**
10: *Swap* records between **mutateCluster1** and **mutateCluster2**
11: end
12: *Add* new chromosome to list of candidate
13: end

D. Crossover

The mutation operator through search process has limitation to precision localization of local maximum of fitness function. There are many local maximums but not all of them close to overall optimal. To prevent search process of concentration its attempt to reach maximum in small area genetic algorithm has crossover operator. It supports strong impact to search process to leave local maximum and search in another area of possible solution. Of course many of these attempts will fail due to lack of values of fitness function. In case of success to hit in area of more close area to optimum mutation helps to find the best solution in this local maximum. In comparison with mutation as a weak tool the crossover has strong impact to process. The implemented version of crossover operates with genes in chromosome. The first step is to combine two chromosomes to perform crossover at middle. So the first half of first chromosome connects with second half of second chromosome and opposite for remaining parts. Next it reorganizes cluster in newly created chromosome. The rule is to preserve first half of chromosome and reorganize other part according to constraint which is equal amount records in each cluster and the same cluster before perform operation crossover. The way is to walk through second half and leave unchanged records within clusters that does not exceed maximum number records in class and clear cluster information in not consistent clusters in current chromosome. On last step all cleared record fixed to cluster without records, step by step. By the end chromosome has all preserved full cluster which is located in first half of chromosome and new initialization broken clusters in second. New chromosomes add to list of candidate in selection process.

Algorithm 2 Crossover

0: PROCEDURE: **Crossover**
1: Input: Population **P**, CrossOverRate **rate**, Cluster **Cl**
2: begin
3: *Find* **CrossNum** of crossover as **P** * ***rate***
4: while **CrossNum** *is not* satisfied do
5: **chromosome1**← Randomly *select* from **P**
6: **chromosome2**← Randomly *select* from **P**
7: *Swap* half of **chromosome1** and **chromosome2**
8: *Normalize* **chromosome1** and **chromosome2**
9: end
10: *Add* **chromosome1** and **chromosome2** to list of candidate
11: end

V. EXPERIMENTAL STUDY

For experimental study scenarios were selected demonstration the range of application of described algorithm. Also the experiment setups relation between fitness function value and k coefficient values.

The first experiment shows the evolution progress during generation and fitness function value increasing. It is one run of experiment and sharp rise of fitness function depends on design of ranking solution. During experiment fitness function calculates value for all clusters and returns minimum values of them for chromosome. The reason of value increasing is increasing l of all clusters with low diversity.

Conclusion of experiment is statistical proving of effectiveness described method. It collects statistics of 50 runs and aggregates results on Fig. 1. Increasing of fitness function value shows improving distribution of sensitive values.

The second experiment describes changing contribution of genetic algorithm to find optimal solution. It shows how fitness function increases from initial condition to solution found by method with respect to k coefficient. The k is changing in range from 3 to 28 with 1 step. The contribution of suggested method stays the same with changing k.

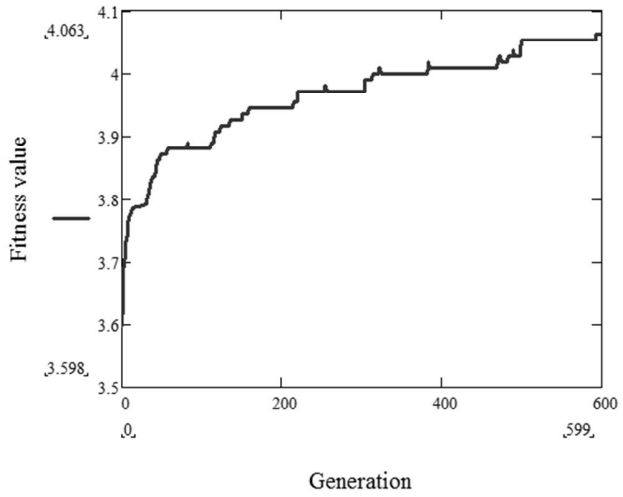

Fig. 1. Fitness function progress during evolution on 50 runs

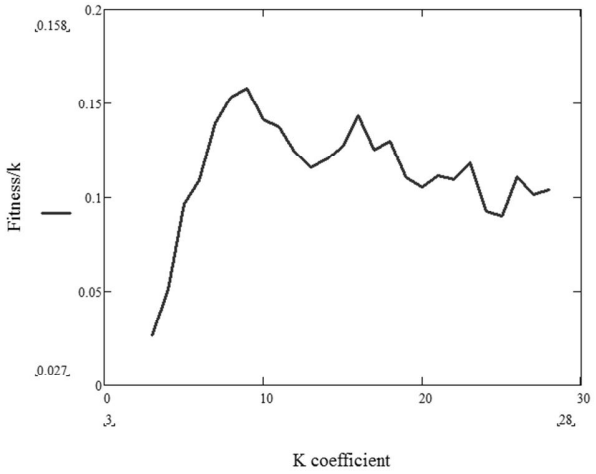

Fig. 2. Correlation between growth fitness value and k

As shown, the suggested method has advantages in high values of k coefficient when heuristics algorithm does not demonstrate gain for anonymization. Obviously that increase of the amount of sensitive attribute will dramatically increase complexity of search of optimal solution for heuristics approaches and do the genetic algorithm more acceptable for this type of research task. The next step of current work is to extend algorithm to multidimensional optimization problem based on different privacy models. The possibility of count information loss during anonymization process can help to apply suggested algorithm model to production use in real system. It is the additional constraint coupled with privacy based k-anonymity, l-diversity and other privacy aspect models.

The third experiment shows relation between the algorithm runtime and data block size (Fig. 3). For performance description used parameters k=10, v=100 and data block size (b) from 100 to 1000.

The graph shows average runtime on 50 runs of algorithm.

The experiment displays linear decrease of the algorithm performance on data block size increasing.

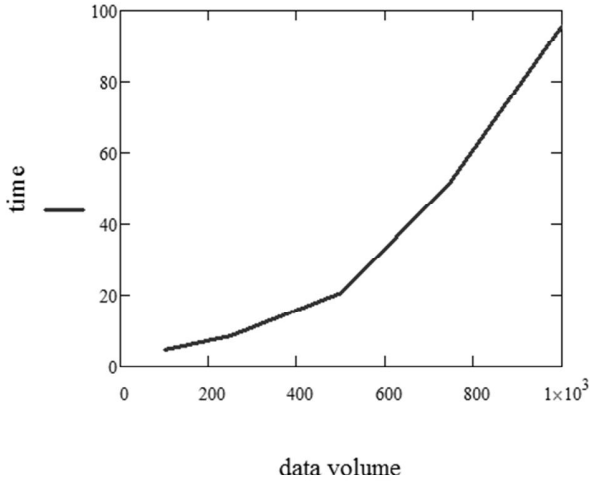

Fig. 3. Relation between the algorithm runtime and data block size

The fourth experiment shows relation between the algorithm runtime and k parameter value at constant data block size b=1000 and variability v=100 and k parameter from 10 to 250 (Fig. 4).

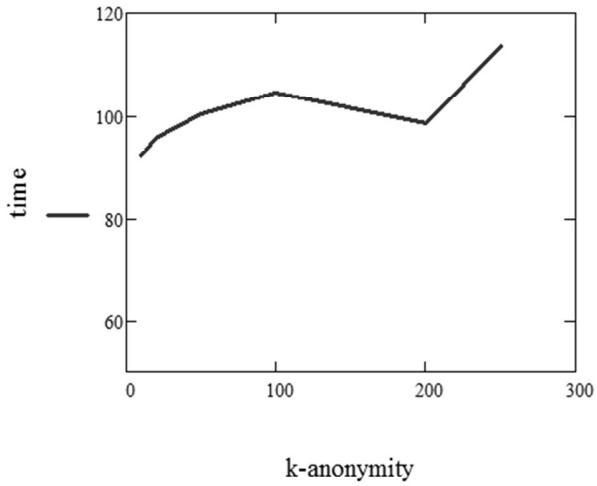

Fig. 4. Relation between the algorithm runtime and k parameter value

The experiment shows that changing of the k-anonymity parameter does not affect the algorithm performance. The graph shows average runtime on 50 runs of algorithm.

The fifth experiment shows relation between the algorithm runtime and variability parameter value at constant data block size b=500 and k-anonymity k=10 and v parameter from 10 to 100 (Fig. 5).

The experiment shows that changing the variability of the input data does not affect the algorithm performance. The graph shows average runtime on 50 runs of algorithm.

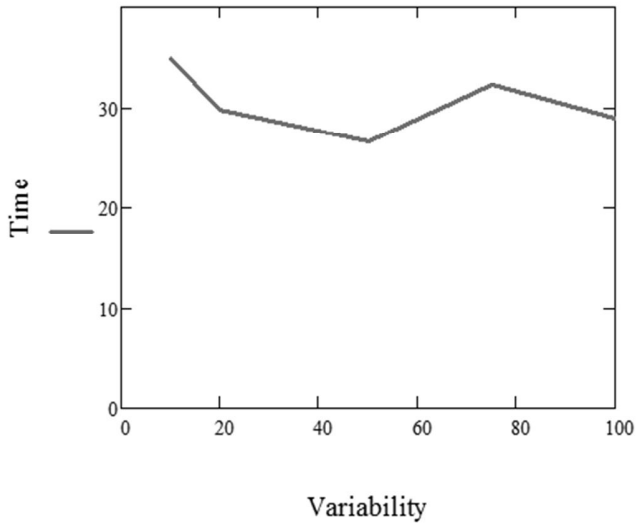

Fig. 5. Relation between the algorithm runtime and variability parameter value

The sixth experiment shows the growth of the fitness function value and increasing variability parameter value at constant data block size b=1000 and k-anonymity k=10 and v parameter from 10 to 75 (Fig. 6).

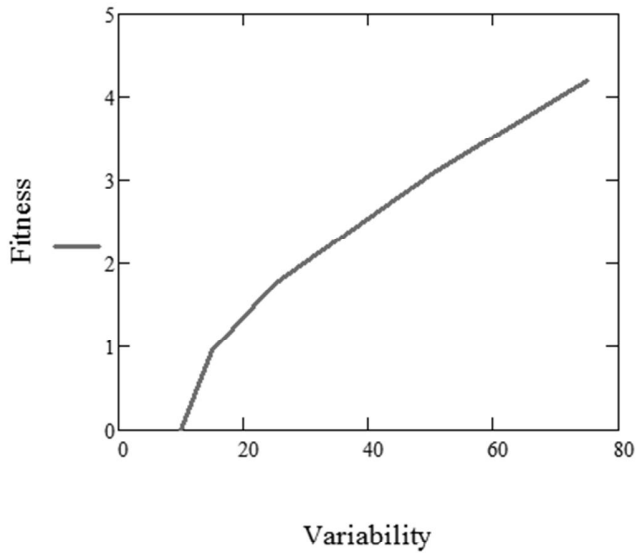

Fig. 6. Relation between the fitness function value and variability parameter value

The experiment shows how the changing the variability of the input data affects on the output data l-diversity value. There is a dependency between variability of the input data and the growth of fitness function value. The greater diversity of data allow to further increase the value of l-diversity. The graph shows average runtime on 50 runs of algorithm.

VI. Conclusion and Future Work

Information sharing has become part of the routine activity of many individuals, companies, organizations, and government agencies. Privacy-preserving data publishing is a promising approach to information sharing, while preserving individual privacy and protecting sensitive information. In this survey, we reviewed the recent developments in the field. The general objective is to transform the original data into some anonymous form to prevent from inferring its record owners' sensitive information.

We presented our views on the difference between privacy-preserving data publishing and privacy-preserving data mining, and gave a list of desirable properties of a privacy-preserving data publishing method. We reviewed existing methods in terms of privacy models, anonymization operations, information metrics, and anonymization algorithms. Most of these approaches assumed a single release from a single publisher, and thus only protected the data up to the first release or the first recipient. We also reviewed several works on more challenging publishing scenarios, including multiple release publishing, sequential release publishing, continuous data publishing, and collaborative data publishing.

Privacy protection is a complex social issue, which involves policy-making, technology, psychology, and politics. Privacy protection research in computer science can provide only technical solutions to the problem. Successful application of privacy preserving technology will rely on the cooperation of policy makers in governments and decision makers in companies and organizations. Unfortunately, while the deployment of privacy-threatening technology, such as RFID and social networks, grows quickly, the implementation of privacy-preserving technology in real-life applications is very limited. As the gap becomes larger, we foresee that the number of incidents and the scope of privacy breach will increase in the near future. Below, we identify a few potential research directions in privacy preservation, together with some desirable properties that could facilitate the general public, decision makers, and systems engineers to adopt privacy-preserving technology.

Most previous privacy-preserving techniques were proposed for data publishers, but individual record owners should also have the right and responsibility to protect their own private information. There is an urgent need for personalized privacy-preserving tools, such as privacy-preserving web browsers and minimal information disclosure protocols for e-commerce activities.

The current sensors area of generating data covers privacy level of individuals. There is problem of processing sensor data with preserving privacy level designated by persons whom belongs information. The anonymization of data for processing can solve this type untrusted relations. Due to significant volume of data there is additional aspect related to flat store data where a lot of existing methods are not applicable. The method of anonymization suggested in paper is based on genetic clusterization algorithm is one of the approaches to reach aim of preserved privacy level. The criteria of satisfying is implemented according k-anonymity and l-diversity privacy model. The main part of algorithm with improvement related to the anonymization process is described in paper. There are several ways to improve current algorithm

by implementing multi objected privacy model and extending number of ranking criteria.

In the future development expects following ways of algorithm progress.

The opportunity of well-known repository data using expected. It requires transition from simulated data using to real data import methods.

The transition to real data sets is allowing to compare the algorithm with existing solutions in this area on such parameters as k-anonymity and l-diversity.

The transition to clusterization on full attributes set. It will give the opportunity to algorithm working on real data also.

REFERENCES

[1] G. Panchal, D. Panchal, "Solving NP hard Problems using Genetic Algorithm", *International Journal of Computer Science and Information Technologies*, vol. 6 (2), 2015, pp. 1824-1825.

[2] A. Stopczynski, R. Pietri, A. Pentland, D. Lazer, S. Lehmann (2014) "Privacy in Sensor-Driven Human Data Collection: A Guide for Practitioners", *arXiv preprint arXiv*:14035299.

[3] Benjamin C. M. Fung, K. Wang, R. Chen, P. S. Yu. "Privacy-preserving data publishing: A survey on recent developments", *ACM Computing Survey*, 2009.

[4] Dalenius, T. "Finding a needle in a haystack — or identifying anonymous census records", *Journal of Official Statistics 2* (1986): pp. 329-336.

[5] Kristen LeFevre , David J. DeWitt , Raghu Ramakrishnan, "Incognito: efficient full-domain K-anonymity", *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, June 14-16, 2005, Baltimore, Maryland.

[6] Lin, J.-L., Wei, M.-C., Li, C.-W., Hsieh, K.-C., "A hybrid method for k-anonymization", *In Proceedings of IEEE Asia-Pacific Services Computing Conference (APSCC'08)* Yilan, Taiwan: IEEE Computer Society, 2008 pp. 385-390.

[7] Lin, J.-L., Wei, M.-C., "An efficient clustering method for k-anonymization", *In International workshop on privacy and anonymity in the information society (PAIS)*, 2008.

[8] Byun, J.-W., Kamra, A., Bertino, E., & Li, N., "Efficient k-anonymization using clustering techniques", *In International conference on database systems for advanced applications (DASFAA)*, 2007.

[9] Jun-Lin Lin, Meng-Cheng Wei. "Genetic algorithm-based clustering approach for k-anonymization", *Expert Systems with Applications*, 2009(36), pp. 9784-9792.

[10] Goldberg, D. E., *Genetic algorithms in search, optimization and machine learning*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[11] Sweeney, L., "K-Anonymity: A model for protecting privacy", *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5) 2002, pp. 557–570.