KAB: A new k-anonymity approach based on black hole algorithm

Mahieddine Djoudi, Lynda Kacha, Abdelhafid Zitouni

Journal of King Saud University – Computer and Information Sciences xxx (xxxx) xxx

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

KAB: A new k-anonymity approach based on black hole algorithm

Lynda Kacha [a,*], Abdelhafid Zitouni [a], Mahieddine Djoudi [b]

[a] LIRE Laboratory, University of Constantine 2, Algeria
[b] TechNE Laboratory, University of Poitiers, France

ARTICLE INFO

Article history:
Received 21 January 2021
Revised 21 April 2021
Accepted 27 April 2021
Available online xxxx

Keywords:
Privacy
Anonymization
K-anonymity
Clustering
Black hole algorithm

ABSTRACT

K-anonymity is the most widely used approach to privacy preserving microdata which is mainly based on generalization. Although generalization-based k-anonymity approaches can achieve the privacy protection objective, they suffer from information loss. Clustering-based approaches have been successfully adapted for k-anonymization as they enhance the data quality, however, the computational complexity of finding an optimal solution has shown as NP-hard. Nature-inspired optimization algorithms are effective in finding solutions to complex problems. We propose, in this paper, a novel algorithm based on a simple nature-inspired metaheuristic called Black Hole Algorithm (BHA), to address such limitations. Experiments on real data set show that data utility has been improved by our approach compared to k-anonymity, BHA-based k-anonymity and clustering-based k-anonymity approaches.

1. Introduction

Privacy preserving is one of the major concerns when data containing sensitive information is published. In this context, the data, called micro-data, consist of a relational table in which, each row consists of basic information describing an individual according to an attribute of the table. Thus, in a table representing individuals, we can find four distinct groups of attributes (explicit identifiers[1], quasi-identifiers[2], sensitive attributes and non-sensitive attributes) (Ciriani et al., 2006). The well-known technique to privacy preserving micro-data is anonymization. Anonymization is a process to hide user identities in order to protect their sensitive information. One simple practice to do this is removing explicit identifier like name or address. However, it is not sufficient, indeed data subjects can still be re-identified. The weakness of simple anonymization was demonstrated by *L. Sweeney* in (Sweeney, 2002) and was further confirmed by *Y.A. de Montjoye et al.* in (De Montjoye et al., 2015). By joining two public data sources (anonymized medical insurance dataset of the Massachusetts employees and voter's registration list), L. Sweeney successful identify William Weld, a former Governor of Massachusetts. This attack, named "linking attack", is shown in Fig. 1.

To overcome this problem, more elaborate anonymization approaches were proposed. K-anonymity (Sweeney, 2002) is one of those approaches. It is the first model proposed in the literature and remains the most used for privacy protection in micro-data. It requires that each record in a table cannot be distinguished among at least k-1 other records. In this approach, the records of the original table are first divided into several groups called equivalence classes. Then, the records of each group are generalized, to share the same values of quasi-identifiers. Therefore, it becomes difficult to identify an individual in a group, since all the individuals of the same group are similar. This idea is very similar to clustering. Based on this definition, k-anonymity can be viewed as a clustering problem where the objective is to find a set of clusters of k-records. Each equivalence class can be considered as a cluster and the centroid as a form of generalization of an equivalence class. Clustering-based k-anonymity approaches have a good data utility as they group similar records together.

Although the idea of k-anonymity-based clustering is conceptually simple, the computational complexity of finding an optimal k-anonymous solution is NP-hard (Meyerson and Williams, 2004). In this context, great efforts have been dedicated to developing approaches that are able to support an exhaustive search of optimal solution i.e., which has a minimal information loss as possible. However, the majority of them fail and suffer from bad data quality. The use of nature inspired metaheuristic algorithms to solve NP-hard problems is well known and quite effective in many areas,

* Corresponding author at: LIRE Laboratory, Faculty of Nouvelles Technologies de l'Information et de la Communiation (NTIC), University Abdelhamid Mehri Constantine 2, Nouvelle ville Ali Mendjeli, BP: 67A, Constantine, Algeria.
E-mail addresses: lyndakacha@yahoo.fr (L. Kacha), Abdelhafid.zitouni@univ-constantine2.dz (A. Zitouni), mahieddine.djoudi@univ-poitiers.fr (M. Djoudi).

[1] Attributes that can uniquely identify an individual

[2] Attributes that cannot uniquely identify an individual by themselves but combined with other external data may be re-identify the individual

https://doi.org/10.1016/j.jksuci.2021.04.014
1319-1578/

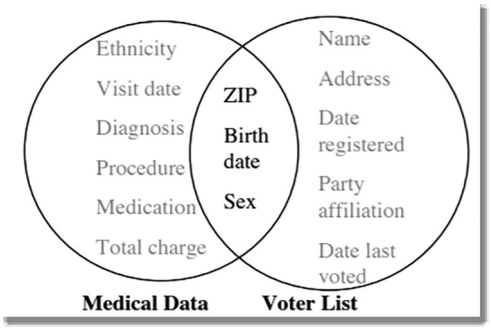

Fig. 1. Linking attack to re-identify data (*Sweeny, 2002*).

but its application in the field of privacy and anonymity has been little explored. The nature inspired metaheuristic algorithms are approximate optimization methods, inspired by nature, used to solve difficult problems when exact methods fail or require unacceptable or expensive computation time. They are based on random mechanisms to explore the research space in order to find or approach a global optimum (Talbi, 2009).

We propose, in this paper, a novel approach based on a simple metaheuristic, called Black Hole Algorithm (BHA for short) (Hatamlou, 2013). We have chosen this metaheuristic for its simplicity and its reduced number of parameters; BHA has a simple structure and is easy to implement. It is free from parameter setting issues, which boil down to a random number in [0–1]. Furthermore, BHA has not yet been applied to the area of privacy and has never been adapted to anonymization process. BHA is a population-based algorithm inspired by the black hole phenomenon. The black hole phenomenon is a region of space that forms when a massive star is formed. Its gravitational power is such that any object close to it, including light, disappears forever in the universe. The BHA applied to k-anonymization problem represents the k-anonymous solutions by the stars and the best solution by the black hole. The algorithm starts with an initial population of candidate solutions generated randomly. Its objective is to find, based on this population, the optimal k-anonymity solution i.e., which has the smallest information loss. As our objective is to address limitation of clustering-based k-anonymity, we consider the initial population of BHA as a set of clustering-based k-anonymous solutions generated by a clustering algorithm. In order to improve the data quality, the clustering algorithm should place similar records (with respect to the quasi-identifiers) in the same group, each group containing at least k records. The similarity is computed based on information loss as a distance and cost metric. This ensure that less distortion is required to anonymize the record in a cluster which enhance data quality.

The rest of the paper is organized as follows. The next section surveys related work about k-anonymity-based approaches. Our algorithm is presented in section 3 and is experimentally evaluated in section 4. We conclude this paper in section 5.

2. Related work

K-anonymity model has got interest much for the past few year. Many approaches have been proposed for k-anonymization and its variations. A survey of various k-anonymization approaches was realized by Ciriani et al (Ciriani et al., 2006). As our approach is a k-anonymity approach based on nature inspired optimization algorithm and clustering algorithm, we describe in this section k-anonymity approaches, proposed in the literature, based on these two concepts. Table 1 summaries these approaches.

2.1. k-Anonymity based on clustering approaches

Li et al. Li et al. (2006) proposed a clustering algorithm called K-Anonymization by Clustering in Attribute hierarchies (KACA), to achieving k-anonymity by local recording. The start point of KACA algorithm is a set of equivalence classes created based on sorted data set. The algorithm first chooses randomly an equivalence class C of size smaller than k. It, then, evaluates the distance between C and all other equivalences classes and finds the equivalence class C' with the smallest distance to C. Finally, the two equivalence classes C and C' are merged and generalized. This process continues until each equivalent class contains at least k records.

Chiu and Tsai (Chiu and Tsai, 2007) adapted a Weighted Feature C-means clustering algorithm for k-anonymization and proposed an algorithm called (WF-C-means). The proposed algorithm starts by calculating the number of equivalence classes (C) such that $C = no\ tuples/k\text{-}value$ and selects C random records as seeds. Then, the algorithm assigns to each quasi-identifier a weight. The next step is the formation of equivalence classes, by assigning all records to their respective closest equivalence class (i.e., closest seed), and updates feature weights to minimize information loss. This step iterates until the affectation of records to clusters stops changing. The final step consists of merging small equivalence classes (which contain less than k records) with large equivalence classes to meet the constraint of k-anonymity.

Byun et al. (Byun et al., 2007) proposed another k-anonymity based clustering called k-member clustering. The algorithm starts by building a cluster. It selects randomly a record r as the seed and adds iteratively the $k\text{-}1$ closest records to it. Then, the algorithm selects a new record that is the furthest from r, and repeats the same process to build the next cluster. If there are any unassigned records, the algorithm individually assigns these records to their respective closest clusters. The process ends while all the records are assigned to the clusters.

Loukides and Shao (Loukides and Shao, 2007) proposed another greedy k-anonymization algorithm. Like k-member algorithm (Byun et al., 2007), the algorithm starts by selecting a seed randomly. However, they differ on building clusters. In this algorithm, a cluster is formed by adding iteratively the closest tuples of seed to a cluster until a user-defined threshold is reached. If the number of records in a cluster is fewer than k, the cluster is deleted.

Lin and Wei (Lin and Wei, 2008) proposed a two-stage algorithm called One pass K-means Algorithm (OKA) for achieving k-anonymity based on clustering. The OKA starts by defining a metric space of the quasi-identifying attributes over the database records, which are then viewed as points in this space. Then, in the first stage the points are partitioned using k-means algorithm, but only for the first iteration, in a set of groups of $size >= k$. During the second stage, the sizes of clusters are adjusted to ensure that each cluster contains no fewer than k records. The adjustment is done by moving the records from clusters with more than k records to clusters with fewer than k records.

Lin et al. (Lin et al., 2008) proposed a hybrid algorithm which combines OKA (Lin and Wei, 2008) and the k-member algorithm (Byun et al., 2007), called the Hybrid Method. OKA is used in the first step to generate some clusters with small information loss, and the k-member algorithm in the second step to generate the remaining clusters with less fluctuation of information loss.

Aggarwal et al. (Aggarwal et al., 2010) proposed a clustering-based algorithm for achieving k-anonymity in hierarchical attribute structures. The basic idea of the algorithm is finding an arbitrary equivalence class of size smaller than k and merging it with the closest equivalent classes to form a larger equivalent class with the smallest distortion. This process repeats recursively until each equivalent class contains at least k tuples.

Table 1
Summary of k-anonymity approaches.

	Papers	Proposed approach	Compared approaches	Compared parameters	Compared metrics	Test dataset	Simulation results
Clustering-based k-anonymity	Li et al., 2006	K-Anonymization by Clustering in Attribute hierarchies (KACA)	Incognito (LeFevre et al., 2005)	no QID = 3…8	. execution time . Distortion ratio	Adult (UCI)	. better distortion ratio . worse execution time
	Chiu and Tsai, 2007	Weighted Feature C-Means Clustering Algorithm (WF-C-means)	. Single-link clustering (Jain et al., 1999) . Complete-link clustering (Jain et al., 1999) . Average-link clustering (Jain et al., 1999)	k = 2…16	. execution time . Distortion ratio . Classification Error Rate (CER)	. Wine (UCI) . Iris (UCI) . Zoo (UCI)	. better distortion ratio . better CER for Wine and Zoo datasets. Better CER in average-link for Iris dataset . better computational efficiency
	Byun et al., 2007	Greedy k-members clustering algorithm (K-members & K-members CM)	. Mondrian (LeFevre et al., 2006)	. k = 2…500 . no tuple = 1 k…30 k	. execution time . Total information loss (Total-IL) . Discernibility Metric (DM) . Classification Metric (CM)	Adult (UCI)	. better Total-IL, DM and CM . worse execution time
	Loukides and Shao, 2007	Greedy k-anonymization algorithm	. Mondrian (LeFevre et al., 2006) . K-members (Byun et al., 2007)	. k = 10…120 . no tuple = 50…1k	. execution time . usefulness . protection . Discernibility Metric (DM)	. Adult (UCI) . Synthetic (IBM research)	. better protection . better usefulness for adult dataset- worse usefulness for synthetic dataset . execution time falls between the two compared algorithms . better DM than Mondrian
	Lin and Wei, 2008	One-pass K-means Algorithm (OKA)	. K-members (Byun et al., 2007)	k = 50…500	. execution time . Total information loss (Total-IL)	Adult (UCI)	. better Total-IL . better execution time
	Lin et al., 2008	Hybrid algorithm (OKA + K-members)	. K-members (Byun et al., 2007) . OKA (Lin and Wei, 2008)	k = 50…500	. execution time . Total information loss (Total-IL)	Adult (UCI)	. better Total-IL . execution time falls between the two compared algorithms
	Aggarwal et al. (2010)	Method for anonymization data records	no experimentation				
	He et al., 2012	Clustering-Based Generalization Algorithm (CB)	. Nonhomogeneous Generalization (Wong et al., 2010)	. k = 50…250 . no tuple = 10 k.0.50 k . no QID = 3…6	. execution time . Global Certainty Penalty (GCP)	. SAL (ipums) . INCOME (ipums)	. better GCP . worse execution time
	Pramanik et al., 2016	Enhanced k-anonymity Algorithm (KOC)	. K-means (Jagannathan and Wright, 2005)	. k = 20…250 . no QID = 3…8	. execution time . Distortion Ratio	Adult (UCI)	. better GCP . worse execution time
	Aghdam and Sonehara, 2016	Similarity-based Clustering Algorithm (SBCA)	. Mondrian (LeFevre et al., 2006) . Datafly (Sweeney, 1998) . Incognito (LeFevre et al., 2005)	. k = 2…100	Total Normalized Certainty Penalty (Total NCP)	. Adult (UCI) . Internet Service Provider (ISP)	better Total NCP
	Bhaladhare and Jinwala, 2016a	. Approach#1: *Unequal combination of QI-SA* . Approach#2: *Equal combination of QID-SA*	. K-members (Byun et al., 2007) . Systematic clustering algorithm (Kabir et al., 2011)	. k = 20…100	. execution time . Total information loss (Total-IL)	Adult (UCI)	. better Total-IL for the two proposed approaches. Total-IL of Approach#2 better than of Approach#1 . better execution time for the two approaches
	Ni et al., 2017	Clustering Based K-anonymity (serial GCCG & parallel GCCG)	. KACA (Li et al., 2006) . Incognito (LeFevre et al., 2005)	. k = 3…10	. execution time . information loss (IL)	Adult (UCI)	. better IL . execution time of serial GCCG falls between the two compared algorithms
	Zheng et al., 2018	Improved K-anonymity Algorithm (IKA)	. Mondrian (LeFevre et al., 2006) . K-members (Byun et al., 2007) . OKA (Lin and Wei, 2008)	. k = 100…500 . no tuple = 5 k.0.30 k	Global Certainty Penalty (GCP)	Adult (UCI)	better GCP

(*continued on next page*)

Table 1 (*continued*)

	Papers	Proposed approach	Compared approaches	Compared parameters	Compared metrics	Test dataset	Simulation results
	Arava and Lingamgunta, 2019	Adaptive k-Anonymity Approach (AKA)	. Hybrid algorithm (Lin et al., 2008) . KOC (Pramanik et al., 2016)	. k = 100. . .500	. execution time . Total information loss (Total-IL)	Adult (UCI)	. better Total-IL . better execution time
	Guo et al., 2019	k-anonymity based on Natural Equivalent Class (NEC)	. K-members (Byun et al., 2007) . OKA (Lin and Wei, 2008) . NEC-based OKA . NEC-based K-members	. k = 2. . .100 . no tuple = 5 k..total . no QID = 1. . .8	. execution time . Global Certainty Penalty (GCP)	Adult (UCI)	. better GCP and execution time of NEC-based K-members than of K-members in average . better GCP and execution time of NEC-based OKA than of OKA in average
	Yan et al., 2021	Weighted K-Member Clustering Algorithm (WKMCA)	. K-members (Byun et al., 2007) . OKA (Lin and Wei, 2008) . IKA (Zheng et al., 2018)	. k = 2. . .50	. execution time . Total information loss (Total-IL) . Average intra/ intra cluster dissimilarity . Standard Deviation (SD)	Adult (UCI)	. better intra-cluster dissimilarity, SD and Total-IL than of K-members and OKA and nearly equal to IKA. . equal inter-cluster dissimilarity in average for the four algorithms . better execution time than for k-members/IKA and worse than OKA
Metaheuristic-based k-anonymity	Lunacek et al., 2006	New crossover operator for Genetic algorithm-based k-anonymity (GA)	. GA with traditional crossover . GA with new crossover	Population size = 200. . .1k	Convergence rate	Adult (UCI)	faster convergence for the same solution quality
	Lin and Wei, 2009	Genetic Algorithm-based hybrid algorithm (GA)	. Hybrid algorithm (Lin et al., 2008) . GA-based Hybrid algorithm	. K = 100. . .500	. Total information loss (Total-IL)	Adult (UCI)	better Total-IL
	Run et al., 2012	Hybrid method based on Genetic Algorithm and Tabu Search algorithm (GA-TS)	. GA-based k-anonymity . TS-based k-anonymity	K = 3. . .15	. Discernability Metric (DM) . Information Loss Metric (ILM)	Pima Indians Diabetes	better ILM and DM in most the case
	Bhaladhare and Jinwala, 2016a	Fractional Calculus-Bacterial Foraging Optimization (FC-BFO)	. Mondrian (LeFevre et al., 2006) . Hilbert space-filling (Moon et al., 2001) . K-members (Byun et al., 2007) . Systematic Clustering (Kabir et al, 2011) . BFO Algorithm (Das et al, 2009)	. K = 20 . . . 100 . no iteration = 5. . .20	. execution time . Total information loss (Total-IL)	. Adult (UCI) . University (UCI)	. better Total-IL . worse execution time than Hilbert algorithm and better than the remains compared algorithms
	Wai et al., 2017	Hierarchical Particle Swarm Optimization for clustering similar data	/	. K = 5–20-30 . no iteration = 300. . .10 k	. execution time . ILoss	Adult (UCI)	/
	Madan and Goswami, 2018	Dragon Particle Swarm Optimization (Dragon-PSO)	. K-Anonymity (Fung et al., 2007) . K-Diversity (Eliabeth and Sarju, 2015) . Genetic Algorithm-based k-anonymity . Dragonfly Algorithm-based k-anonymity	. K = 2 . . . 3 . no cluster = 2. . .5	. Total Information Loss (IL) . Classification Accuracy (CA)	Adult (UCI)	better IL and CA
	Madan and Goswami, 2019a	Duplicate-Divergence-Different properties enabled dragon Genetic algorithm (DDDG)	. GA-based K-Anonymity . GA-based K-Diversity . GA-based k-DDD	. population size = 4 . . . 16 . no iteration = 5. . .20	. Total Information Loss (Total-IL) . Classification Accuracy (CA)	Adult (UCI)	. better Total-IL . CA between from better to equal

Table 1 (*continued*)

Papers	Proposed approach	Compared approaches	Compared parameters	Compared metrics	Test dataset	Simulation results
Madan and Goswami, 2019b	Hybrid algorithm based on Grey wolf optimizer- Cat Swarm Optimization (GWO-CSO)	. K-Anonymity (Fung et al., 2007) . K-Diversity (Eliabeth and Sarju, 2015) . Genetic Algorithm-based k-anonymity . Dragon-PSO . DDDG (Madan and Goswami, 2019a)	. K = 2. . . 5 . no cluster = 2. . .5	. Total Information Loss (Total-IL) . Classification Accuracy (CA)	Adult (UCI)	better Total-IL and CA, except for k = 3 where IL of Dragon-PSO is better

He et al. (He et al., 2012) proposed a clustering-based k-anonymity algorithm. The algorithm is an iterative process. At each round, the data set is partitioned into two subsets *G1* and *G2* based on *Normalized Certainty Penalty* (*NCP*) as a distance measurement. If the size of one of subset is smaller than *k*, assume that $G1 < k$, the sizes of the two sub-sets are adjusted by borrowing $K\text{-}|G1|$ tuples from *G2* to make sure that *G1* has a cardinality upper or equal to *k*. The adjustment continues until each subset contains at least *k* tuples.

Pramanik et al. (Pramanik et al., 2016) proposed a clustering approach to achieve k-anonymity called KOC. The algorithm starts with computing the n- closest neighbors for each record and sorts the records in descending order. Then creates *p* clusters (*p = no tuples/k value*) and chooses the top ranked records (i.e., the records with the most n-closest) in the sorted list as initial centroids. The remaining (*no tuples-p*) records are assigned to theirs closest centroids such that every cluster should fulfil k-anonymity property. If the last cluster created contains less than k records, its records are dispersed to other closest clusters. Finally, the clusters are individually anonymized.

Aghdam and Sonehara (Aghdam and Sonehara, 2016) proposed a bottom up greedy algorithm called Similarity-based Clustering Algorithm (SBCA), to achieve k-anonymity through local recording generalization, for datasets with numerical and categorical attributes without hierarchical taxonomy trees. SBCA starts by sorts the dataset and separates the numerical attributes from categorical one. The next step is clustering the dataset with respect to k value and anonymizing it. For this, the algorithm chooses the first tuple in sorted dataset and adds to it the k-1 closest tuples to form a cluster which is anonymized through local recording. The tuples belonging to the cluster are, then, delated from sorted dataset. The process is repeated until the total tuples in the sorted dataset is none or less than k. the remaining tuples are suppressed or joined to the closest equivalence classes.

Bhaladhare and Jinwala (Bhaladhare and Jinwala, 2016a) proposed two approaches for achieving k-anonymity based on systematic clustering (Approach#1 and Approach#2). Both approaches decompose the original database using a combination of quasi-identifiers (QID) and sensitive information (SA) into separate individual sub-databases then anonymize and publish the generated sub-databases. The difference between the two is that Approach#1creates unequal combination of QID and SA, and Approach#2 creates equal combination of QID and SA. The two approaches start by calculating the number of clusters which is equal to (no tuples/k value), then generate a sub-database using a combination of QID and SA. In Approach#1 an unequal combination of QID and SA is created. In Approach#2 an equal combination of QID and SA is created. From each generated sub-database, a partition of all records into k groups is created. Then Systematic clustering algorithm (Kabir et al., 2011) is applied to generate the clusters. For this, a record is randomly selected from the first group for the creation of the first cluster. Other records from the first group are selected and added to the closest cluster. Similarly, the remaining cluster are created by randomly selecting the records from the remaining groups and other records are selected and added to the closest cluster. If some cluster exceeds to the k size, the extra elements should be added in the corresponding closest clusters.

Guo et al. (Guo et al., 2019) proposed a new concept named Natural Equivalence Class (NEC) and designed a k-anonymization algorithm based on this concept. The idea behind the algorithm is that in raw microdata, there exists naturally equivalence classes i.e. which include all the records with the same quasi-identifier attributes. Those equivalences classes are referred to Natural Equivalence Classes. NEC is used to perform clustering before the anonymization process. The algorithm starts by finding all NECs in the dataset. It takes each NEC which is larger than k as an

independent cluster. The remains NEC are clustered by applying a clustering algorithm (K-members (Byun et al., 2007) or OKA (Lin et al., 2008)). For NECs that cannot be clustered, they are assigned to their respective nearest clusters. The last step is generalization of clusters. Applying NEC to K-members/OKA enhances the utility and efficacity of the original approaches.

Ni et al. (Ni et al., 2017) proposed a clustering-based K-anonymity algorithm called GCCG composed of four steps: *Grading*, *Centering*, *Clustering*, and *Generalization*. In Grading and Centering steps, the records are sorted based on the score computed of each record then the first *X* records are chosen as centroids. The third step is formation of clusters by adding to each centroid the *k-1* closest records. In the final step, the records are generalized. To enhance the performance of GCCG algorithm, the authors also propose a parallelized version of GCCG.

Zheng et al. (Zheng et al., 2018) proposed a clustering-based K-anonymity algorithm which considers the overall distribution of quasi-identifier groups in a multidimensional space. The proposed algorithm first picks randomly a record *r* as a centroid of the first cluster and adds the *k-1* closest records to it, in order to form the first cluster. Then the algorithm chooses the record which has the largest distance between itself and the first centroid and set it to the second centroid. The ith centroid is created by in the same way, based on the distance between the ith record and all the existed centroids. After each step of centroid creation, the algorithm adds the *k-1* closest records to the centroid to form the clusters. At the end of this process, all the clusters created contain *k* records. If there are ungrouped records. The algorithm iterates the remaining records and insert each record into the closest cluster i.e., having the smallest distance with its centroid.

Arava and Lingamgunta (Arava and Lingamgunta, 2019) proposed an adaptive k-anonymity algorithm, called AKA. It is based on KOC's systematic approach (Pramanik et al., 2016) for finding the best seed values. AKA starts with computing the number of clusters p = no tuples/k value. For all record in each group, it computes k-closeness with every other record and sorts them in descending order. Then, it sets in every group the records with minimum and maximum closeness as initial centroids (i.e., 2*p seeds) and builds the clusters. The remaining (np tuples-2p) records are assigned to their nearest clusters, such that every cluster should have k cluster members. The additional records (i.e., which have sizes different to k) are reorganize and append to their nearest clusters. For the clusters with sizes superior to k, the algorithm creates new clusters with minimum of k records. An existing clustering algorithm will be applied to the remain clusters, with sizes inferior to k, to distribute their records.

Yan et al. (Yan et al., 2021) proposed a Weighted k-member clustering algorithm called (WKMCA). The proposed algorithm is a modified k-members (Byun et al., 2007) to reduce the influence of outliers on the clustering effect. For this, WKMCA adds a weighted stage in which a series of weighting indicators have been assigned to evaluate the outlyingness of records in order to facilitate filtering out the outliers. Thereby, k-members is based on those indicators to achieve k-anonymity. The proposed algorithm takes place in three stages: *the weighting phase*, *the grouping phase* and *the adjustment phase*. In the weighting phase, the algorithm calculates the ARscore (Average ranking score) of all the records and chooses β (β = 0.05*no tuples) records with the highest ranking of ARscore as outliers. The outliers are stored separately and re-inserted into the release data table during the adjustment phase. In the grouping phase is clustering-based k-anonymity phase. It consists of applying k-members with modified distance and information loss based on weight score. At the end of this phase, all the clusters contain exactly K records. In the adjustment phase, the algorithm inserts the remaining records and outliers into their respective closest clusters.

2.2. k-anonymity based on nature inspired optimization approaches

Lunacek et al. (Lunacek et al., 2006) proposed a new crossover operator and implemented a Genetic Algorithm-based k-anonymity approach with the proposed crossover operator in order to demonstrate the advantage of using the new operator over traditional crossover operators. In the paper, the new crossover operator was compared with the traditional *2-point reduced surrogate operator* for the various population sizes. In each case, the new crossover operator converges faster to more effective solutions.

Lin and Wei (Lin and Wei, 2009) proposed a Genetic Algorithm (GA)-based clustering approach for achieving k-anonymity. In this approach, the initial population of GA is created based on *Hybrid Method* proposed in (Lin and Wei, 2008). A candidate solution of population encoded by a chromosome and contains no fewer than k genes, where each gene indicates the index of a record in the original dataset. The algorithm uses only selection and crossover operations of GA. Mutation is not performed due to the algorithm uses the original record indexes which cannot be altered.

Run et al. (Run et al., 2012) proposed an hybrid search method based on Tabu Search (TS) and Genetic Algorithm (GA) to achieve k-anonymity. In the proposed method, a TS is embedded into a traditional GA to perform the role of mutation. The objective is to overcome the limitation of "climbing" ability of traditional TS from a single start point, by implementing local search from multiple start points getting from GA. The proposed algorithm starts with the construction of the solution space lattice (domain generalization hierarchies) representing generalization strategies. Based on the lattice, genetic algorithm creates the initial population which is used by tabu search to find the optimal generalization strategy.

Bhaladhare and Jinwala (Bhaladhare and Jinwalab, 2016) proposed a Fractional Calculus-based Bacterial Foraging Optimization Algorithm called FC-BFO to generate an optimal clustering. The objective of FC-BFO is to improve the optimization ability and convergence speed of BFO algorithm (Das et al., 2009) by applying to it the concept of FC in its chemotaxis step. Since the fractional order derivative has inherent memory capacity, the fractional calculus perspective leads to a smoother variation and a longer memory effect. Effectively, the FC-BFO presents a better information loss and execution time than BFO. BC-BFO takes place in two main stages; The first step consists of generating initial population based on a clustering algorithm and evaluating the objective functions and the privacy factors of each solution. The next step is applying BFO algorithm on the generated population, which consists of three steps such as *modified chemotaxis step based on a fractional calculus*, *reproduction step* and *elimination dispersal step*. These steps are used to generate clusters in the k-anonymized database.

Wai and al. (Wai and al., 2017) proposed a big data privacy preservation approach based on Hierarchical Particle Swarm Optimization (HPSO). The proposed approach is built upon MapReduce Hadoop infrastructure to address the scalability issues of big data. It consists of two stages; The first stage is HPSO clustering. The algorithm creates a MapReduce job to produce the predefined numbers of intermediate clusters, represented by particles, then a MapReduce job of HPSO clustering is executed on each cluster by iteratively performing Map and Reduce steps until the number of data members in each particle exceed k. In the second stage, the resulted clusters are generalized to be transformed into their anonymized forms. The Map step simply passes all data members of each intermediate cluster to its respective Reduce step which performs HPSO clustering job to produce k-anonymized clusters.

Madan and Goswami proposed two hybrid optimization algorithms to achieve k-anonymity called Dragon-PSO (Madan and Goswami, 2018) and GWO-CSO approach (Madan and Goswami, 2019a). The Dragon-PSO algorithm combines the Dragonfly Algorithm (DA) and Particle Swarm Optimization (PSO) by modifying

the update process of DA using PSO. More precisely, the update's formula of the proposed Dragon-PSO algorithm is the sum of the update's formulas of the two original algorithms divide by two. The GWO-CSO algorithm combines Grey Wolf Optimizer (GWO) and Cat Swarm Optimization (CSO). In GWO-CSO algorithm, the update is performed using a modified GWO update by adding a new term corresponding to the update of CSO algorithm.

Madan and Goswami (Madan and Goswami, 2019b) proposed an anonymity model for data publishing based on K-DDD measure, Dragonfly operators-based Genetic Algorithm called Duplicate-Divergence-Different properties enabled Dragon Genetic (DDDG) algorithm. The first step, called k-DDD anonymization, is the transformation of original database to k-DDD database based on the proposed k-DDD measure. k-DDD measure modifies the original database by creating 'k' number of Duplicate records, 'k' number of Divergence in sensitive attributes and 'k' number of Different service providers in each cluster of the database. The next step is applied D-Genetic algorithm on k-DDD database. D-Genetic algorithm is formed through the modification of Genetic Algorithm (GA) with the Dragonfly Algorithm (DA). The proposed D-Genetic algorithm is an iterative process composed of six stages: (1) selection of two clusters as parents, (2) application of crossover operator, (3) application of mutation operator, (4) application of the Dragon operators and update the new positions of records, (5) fitness Evaluation. The process terminates when the total number of iterations is meet. An adapting version of DDDG algorithm based on MapReduce framework is also proposed in the paper.

The clustering-based approaches enhanced the data quality of k-anonymization based on generalization, however an exhaustive search for an optimal clustering solution is potentially exponential. The nature inspired optimization approaches meet this limit as they are more suitable for exploring a large search space. Therefore, their limit lies to their computational complexity. In this paper we try to address these limitations by combining the two approaches.

3. Proposed approach

This section proposes a novel approach for k-anonymization based on black hole algorithm, called **K**-**A**nonymity based on **B**lack hole algorithm (KAB for short). The KAB algorithm starts with an initial population of stars, representing a clustering-based k-anonymous solutions and then evolves the population to find the best k-anonymous solution i.e., having the smallest information loss. As the proposed algorithm derives from black hole algorithm, the evolution of population to an optimal solution is done by moving all the stars to the best solution, represented by the black hole.

3.1. Clustering algorithm

Our approach is based on a clustering algorithm to create candidate solutions. Each solution represents a k-anonymous clustering. In order to obtain a good data quality, the records in a cluster should be as similar as possible. This ensures that less distortion is needed to generalize the records from the same cluster thus resulting in good data quality. To reach this objective we adopt the *Normalized Certainty Penalty* (NCP) (Xu et al., 2006) as distance and cost measurement of clustering algorithm. NCP is an efficient and easy to use metric which measures the degree of information loss caused by the anonymization process.

The distance function measures the dissimilarities between the data points and is generally determined by the type of data (numeric or categorical). Let D the domain of one numerical/categorical attribute. The distance between any two values v_i and $v_j \in$ D is defined as

- Numeric attributes

$$\delta_N(v_i, v_j) = NCP_N = \frac{|v_i - v_j|}{|D|} \tag{1}$$

where |D| is the range of all tuples on D. i.e., in the whole table.

- Categorical attributes

$$\delta_C(v_i, v_j) = NCP_C = \frac{size(c)}{|D|} \tag{2}$$

where c is the closest common ancestor of $v_i$and v_j in the taxonomy tree and size (c) is the number of leaf nodes in c. |D| is the number of distinct values on D.

- Distance between records

The distance between two records $r_1$and r_2 is the sum of distances between corresponding numeric and categorical quasi-identifiers in the two records. Le$QI = \{N_1, \cdots, N_m, C_1, \cdots, C_n\}$t be the quasi-identifiers of dataset T, where $N_i (i = 1, \cdots, m)$ is numeric attribute, and $C_j (j = 1, \cdots, n)$ is categorical attribute. Then the distance between records $r_1$and r_2 is defined as

$$\delta_R(r_1, r_2) = NCP_R = \sum_{i=1}^{m} \delta_N(r_1[N_i], r_2[N_i]) + \sum_{j=1}^{n} \delta_C(r_1[C_j], r_2[C_j]) \tag{3}$$

The cost function is defined by the objective of the clustering problem. Since our objective is to minimize information loss, we use the eq. (3) as a cost function.

The pseudo-code of clustering algorithm is summarized in Fig. 2.

The clustering algorithm starts by calculating the number of clusters N. The number of clusters is an important criterion as it has an influence on the quality of clustering. The larger the number of clusters the smaller the information loss due to the small NCP value resulting. This is why we have chosen the greatest possible number of clusters i.e., M/k (where M is the total number of records in the dataset and k is k-anonymity parameter). After calculation of N, the algorithm runs the clustering process (step 2–5). First, it selects a record randomly and sets it as a centroid. The randomness ensures that the solutions created, i.e., candidate solutions composing the population, are different. Then, it chooses, iteratively, the $k-1$ records having the smallest NCP values with the centroid to form a cluster. The process iterates until all the records are treated. If there are unassigned records, the algorithm affects each of them to its respective closest cluster i.e., the cluster which involves the smallest NCP between its centroid and the unassigned record.

3.2. Solution encoding

After the creation of candidate solutions with clustering algorithm, they must be encoded in order to be used by optimization algorithm i.e., BHA. As mentioned previously, the population is composed of a set of stars which corresponds to feasible solutions created by the clustering algorithm. To properly represent a solution, each star, in our approach, is encoded[3] by an integer array of one dimension, of size equal to the number of records in dataset, where the ith index indicates the ith record and the ith element indicates the id of the cluster to which the ith record belongs.

For example, let X be a candidate solution composed of 15 records and partitioned on 6 clusters. X= {(1,2), (3,5), (4,6,7), (8,9), (10,12,13), (11,14,15)}. The coding solution of X is X' such as

[3] The encoding represents the location of the star

X'=	1	1	2	3	2	3	3	4	4	5	6	5	5	6	6	Cluster id
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Record id

3.3. Fitness evaluation

For evaluate the quality of stars and select the black hole, we use a fitness function (Madan and Goswami, 2018), (Madan and Goswami, 2019a) based on privacy and utility parameters. The fitness function used is considered as a minimization function. The fitness of a star X is calculated using the following equation,

$$fitness(X)_{min} = \frac{1}{2} \times (privacy + utility) \quad (4)$$

The privacy and the utility measures are expressed as follows,

$$privacy(X) = \begin{cases} 0; & if\ X\ satisfy\ k-anonymity \\ 1; & otherwise \end{cases} \quad (5)$$

$$utility(X) = NCP(X) = \sum_{i=1}^{N} NCP_{\alpha_i} \quad (6)$$

where α_i is the ith cluster of X. N is the number of clusters of X. NCP_{α_i} is the NCP of cluster α_i. NCP_{α_i} is calculated by eq.7, where NCP_{record} is the NCP of a *record* in α_i and $|\alpha_i|$ is the size of cluster α_i.

$$NCP_{\alpha_i} = NCP_{record} * |\alpha_i| \quad (7)$$

Given, after anonymization process, all records of a cluster will be similar[4] according to quasi-identifiers, we compute NCP for any record and multiply it by the number of records in the cluster $|\alpha_i|$.

NCP_{record} is calculated by eq.8, where $NCP_{attribute_j}$ is the NCP of attribute j in the record and d is the number of attributes.

$$NCP_{record} = \sum_{j=1}^{d} NCP_{attribute_j} \quad (8)$$

The NCP of $attribute_j$ is calculated based on eq.1 or eq.2 according to its type. If $attribute_j$ is numeric attribute, NCP is calculated by eq.1, such that v_i and v_j correspond to the minimum and maximum values of $attribute_j$ in α_i. If $attribute_j$ is categorical attribute, NCP is calculated by eq.2, such that c corresponds to closest common ancestor of all values of $attribute_j$ in α_i.

3.4. Proposed k-anonymity based black hole algorithm (KAB algorithm)

Like all population-based metaheuristics, KAB algorithm proceeds in three main stages: (1) initialization of the population of stars, (2) evaluation of stars fitness and (3) update the stars locations. The pseudo-code of KAB algorithm is depicted in Fig. 3.

Step 1. Generation of initial population

The initial population of KAB algorithm consists of a set of stars representing feasible clustering-based k-anonymous solutions. Each star X is first created with the NCP-based clustering algorithm describes in Fig. 2, and then encoded as described in section 3.2. The initial population X is represented as the following solution vector,

$$X = \{X_1, X_2, \cdots, X_i, \cdots X_N\} \quad (9)$$

where X_i refers to the ith star in the solution space and N to the population size.

After this step, each star of the population is represented by its *location* (the location of each record, i.e., cluster id) and its *fitness* (cost of the location, i.e., quality of the solution encoded by the star).

Algorithm 1. NCP-based Clustering algorithm

1. Calculate the number of clusters N such as $N = M/k$
2. Pick a random record r from data set
3. Create an equivalence class C_i with r
4. Find the $k-1$ closest records $R_{closest}$ of r based on eq. (3)
5. Form a cluster by adding $R_{closest}$ to C_i
6. Repeat 2-5 until all records are treated
7. If there are not affected records, assign each remains record to the closest cluster[3] with respect to eq. (3)

Fig. 2. Pseudo-code of NCP-based Clustering algorithm.

Algorithm 2. K-Anonymity based Black hole algorithm (KAB)

1. Generate initial population of *stars*
2. Evaluate the fitness of each *star* according to eq. (4) and set the best star as the *b-hole*
3. Move all the *stars* toward the *b-hole* according to eq. (10) and update their fitness
4. If a *star* reaches a best location than the *b-hole*, it becomes the *b-hole* and vice versa
5. If a *star* gets too close to the *b-hole*, a new *star* is created to replace the old one and its fitness is evaluated
6. If maximum number of iterations is met, stop the algorithm, else go to 3.

Fig. 3. Pseudo-code of KAB algorithm.

Step 2. Evaluation and selection of the black hole

After initializing the population, the fitness function of each star is evaluated based on eq. (4) and the best star, which has the best fitness value, is selected as the black hole.

Step 3. Update the positions and the fitness of the stars

The population evolution is done by moving all the candidates towards the best candidate i.e., the black hole. This moving is simulated by changing the location of each star X_i according to eq. (10) (Hatamlou, 2013).

$$X_i(t+1) = X_i(t) + rand * (X_{BH} - X_i(t)) i = 1, 2, 3, \cdots, N \quad (10)$$

where $X_i(t)$ and $X_i(t+1)$ are the locations of the ith star at iterations t and $t+1$, respectively. *rand* is a random number in the interval [0,1]. $X_{BH}$is the location of the black hole. N is the number of stars.

After moving the stars to their new locations, three scenarios are possible: (1) a star reaches a best location than the black hole i.e., with lower fitness value. In such a case the star becomes the black hole and vice versa (exchange their locations and fitness). (2) a star crosses the event horizon[5] of the black hole; In such a case the star will be swallowed by the black hole and replaced by a new star. The radius of the event horizon (R) is formulated in eq. (11) (Hatamlou, 2013). (3) neither of the two previous scenarios and in this case the locations and the fitness are just updated. Once all the stars are moved, next iteration takes place with the new locations of stars and black hole and their corresponding objective functions. The algorithm terminates when max number of iterations is met.

$$R = \frac{fitness(BH)}{\sum_{i=1}^{N} fitness(Xi)} \quad (11)$$

where $fitness(BH)$ and $fitness(X_i)$ are the fitness values of the black hole and the ith star, respectively. N is the number of stars. when the star's distance with black hole is less than a defined radius (R), this star is swallowed by the black hole.

[4] All attribute values on each dimension have the same generalized value

[5] Distance between a star and the black hole.

4. Experimental results

In this section, we evaluate the quality of our proposed algorithm. Since data privacy and utility are the two objectives of any anonymization algorithm, we measure quality of KAB by evaluating the trade-off between these two conflicting objectives. For this, we measure anonymized data utility, according to different privacy levels, represented by k parameter.

4.1. Evaluation metrics

Data privacy of our algorithm is ensured by k-anonymization. We can quantify it with Classification Accuracy (CA) (Madan and Goswami, 2019b), which calculates the rate of k-anonymous clusters in the anonymized data. CA is defined in equation (12); where a cluster is considered *correctly classified* if it satisfies k-anonymity criterion.

$$CA = \frac{Number\ of\ clusters\ correctly\ classified}{Total\ number\ of\ clusters} \tag{12}$$

Given all clusters in the anonymized data produced by KAB algorithm are k-anonymous, its CA is equal to 1 (i.e., the best case).

Data utility is, generally, estimated by measuring the degree of accuracy degradation of anonymized data compared to original data. There are many metrics to evaluate data utility. However, the majority of them are not well suitable for privacy preserving data publishing, due to the lack of knowledge regarding the exact usage scenarios of published data (Ayala-Rivera et al., 2014). To evaluate the data utility of our algorithm, we have used two general-purpose metrics, namely *Classification Metric* (CM) (Iyengar, 2002) and *Average Equivalence Class Size Metric* (C_{AVG}) (LeFevre et al., 2006). We have also chosen two common metrics to measure information loss incurred by the original data after anonymization process, which are *Total Information Loss* (Total-IL) (Byun et al, 2007) and *Normalized Certainty Penalty* (NCP) (Xu et al., 2006).

- ***Total-IL metric***. This metric captures the loss of precision when generalizing a specific attribute. In our experimentation, we use the normalized version of Total-IL, which was transformed to percentage for clarification. The normalized Total-IL (noted *GenTotal_IL*) of a table T, divided into ε number of equivalence classes and composed of quasi-identifiers $QI = \{N_1, \cdots, N_m, C_1, \cdots, C_n\}$ such as N_i is numeric attribute and C_j is categorical attribute, is expressed by the following equation,

$$\text{GenTotal\_IL} = \frac{1}{|T| * d} * Total\_IL(T) * 100 \tag{13}$$

$$Total\_IL(T) = \sum_{e \in \varepsilon} \left(|e| * \left(\sum_{i=1,\cdots,m} \frac{\left(MAX_{N_i} - MIN_{N_j}\right)}{|N_j|} \right) + \left(\sum_{j=1,\cdots,n} \frac{H(\Lambda(\cup C_j))}{H(\Gamma C_j)} \right) \right) \tag{14}$$

where d is the number of quasi-identifiers.$|T|$ and $|e|$ are the number of records in T and e, respectively. MIN_{N_j} and MAX_{N_i} are min and max values in e with respect to attribute N_i. $\cup C_j$corresponds to the union set of values in e with respect to attribute C_j. The term $\Lambda(\cup Cj)$ indicates the subtree rooted at the lowest common ancestor of every value in $\cup C_j$. $H(T)$ is the height of taxonomy tree T .

- ***NCP metric.*** We use the normalized version of NCP described in section 3.3 called Global Certainty Penalty (GCP). In our evaluation GCP was also transformed to percentage. NCP of a table T is expressed by the following equation,

$$GCP = \frac{NCP(T)}{|T| * d} * 100 \tag{15}$$

where d is the number of quasi-identifiers and $|T|$ the number of records in T.

- ***Classification Metric.*** This metric captures the classification errors by penalizing equivalence classes that contain rows with different class labels. The CM of an anonymized table T^* is defined in equation (16),

$$CM(T^*) = \frac{\sum_{allrows} penalty(rowr)}{|T|} \tag{16}$$

where $|T|$ is the number of records in the original table T. A row r is penalized, if it is suppressed or its class label $class(r)$ is different of the majority class label $majority(EC)$ of its equivalence class EC.

$$penalty(rowr) = \begin{cases} 1; if rissuppressed \\ 1; if class(r) \neq majority(EC(r)) \\ 0; otherwise \end{cases} \tag{17}$$

- ***C_{AVG} metric.*** This metric evaluates data utility based on sizes of the equivalence classes. It measures the similarity between the equivalence classes created and the best case, where each equivalence class contains k records. CAVG score for an anonymized table T^* is given by:

$$C_{AVG}(T^*) = \frac{|T|}{|EC| * k} \tag{18}$$

where $|T|$ is the number of records in the original table T, $|EC|$ is the number of equivalence classes created and k is the privacy level.

To obtain a good data utility, *GCP* and *GenTotal_IL* metrics must be small; 0% means no transformation (original data) and 100% means full generalization/suppression of the data. A good data utility must also minimize *CM* and C_{AVG} metrics; In the ideal anonymization *CM* is equal to 0 and C_{AVG} to 1.

4.2. Experimental setup

The experimentations were performed on a Desktop PC with Intel Core2Duo 2.10 GHz CPU and 2 GB of RAM under Windows 7 operating system. The implementations were built and run with Java platform Standard Edition.

We use in our experimentation the *adult dataset* from the UCI machine learning repository, which is considered as a standard benchmark for anonymization. The adult dataset is based on census data and contains a total of 32,561 instances under 15 attributes. Each record describes the personal information of an American and it involves predicting personal income levels as above or below $50 k per year based, on this personal information. Before the experiments, we apply a preprocessing stage on raw dataset to remove duplicate records and records with missing values, and retained only nine of the original attributes, wherein eight attributes[6] are considered as quasi-identifiers and one attribute as sensitive information[7]. Among Quasi-identifiers, *age* and *education* were treated as numeric attributes and the remainder attributes as categorical attributes. The size of the resulting dataset is 30,162 records.

To evaluate the quality of our algorithm, we conduct three experimentations, and observe quality and scalability of KAB

[6] {*age*, *work class*, *education*, *marital status*, *occupation*, *race*, *gender*, *native country*}

[7] {*salary class*}: salary above or below 50k

Table 2
Experimental Setup.

#	Experiment	Parameter settings	Metrics	Compared algorithms
1	Data utility and efficiency	Data size = 30162 k-value∈ [2...40]	GenTotal-IL, GCP, CM, C_{AVG}	K-anonymization (LeFevre et al., 2006) BHA-based k-anonymization KAB-based k-anonymization
2	Scalability and performance	Data size∈ [5000...30162] k-value∈ [2...40]	Execution time, GenTotal-IL, GCP,	
3	Evaluation vs clustering algorithms	Data size = 30162 k-value∈ [2...40]	GenTotal-IL, Execution time	K-members (Byun et al., 2007) OKA (Lin and Wei, 2008) IKA (Zheng et al., 2018) KAB

algorithm compared to k-anonymity (implemented with Mondrian Multidimensional (LeFevre et al., 2006)), k-anonymity with BHA and k-anonymity with clustering algorithms. We set, in the three experimentations, the number of stars as 3 and the maximum number of iterations as 10. The parameters used in our evaluation and compared algorithms are described in Table 2.

4.3. Results and discussion

4.3.1. Data utility and efficiency

In this experiment, we analyze the data utility, of the three algorithms, according to different level of privacy. The simulation results are depicted in Fig. 4, Fig. 5, Fig. 6 and Fig. 7.

Fig. 4 and Fig. 5 report data utility, with respect to information loss, as the value of k increases, represented by *GenTotal-IL* and *GCP*, respectively. They show that information loss, of the three algorithms, increases with the increase of k-value. Given k represents the minimal number of records in each cluster, the larger k is, the more records in one cluster are, the larger the difference between minimal and maximal values in this latter, and the higher the information loss it.

We can observe, from figures, that KAB-based k-anonymization introduces the least information loss whatever privacy level. BHA-based k-anonymization has the worst information loss, because it does not rely on any metric to place records in clusters; This placement is done randomly. Bad results of BHA-based k-anonymization can be explained by the fact that BHA has a low convergence rate and therefore, needs a greater number of iterations to converge. KAB algorithm can be seen as an improvement of BHA; By creating optimal starting solutions with clustering algorithm, KAB

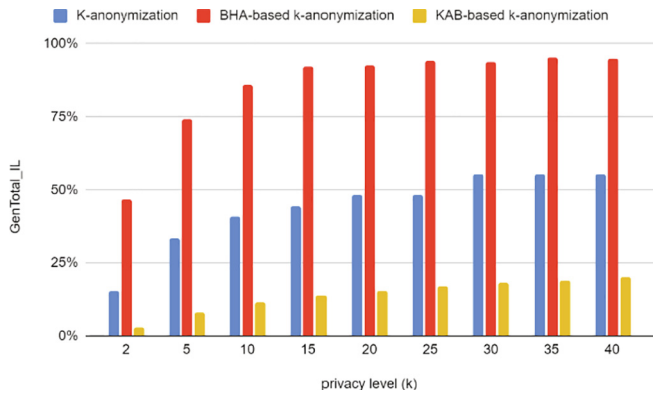

Fig. 4. Information Loss (*GenTotal_IL*) vs. privacy level.

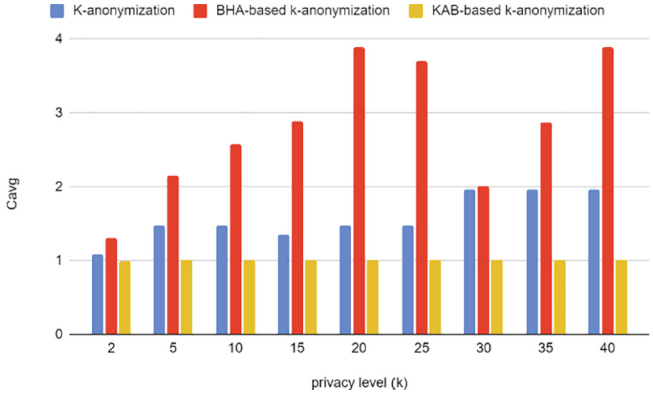

Fig. 6. Average Equivalence Class Size (C_{AVG}) vs. privacy level.

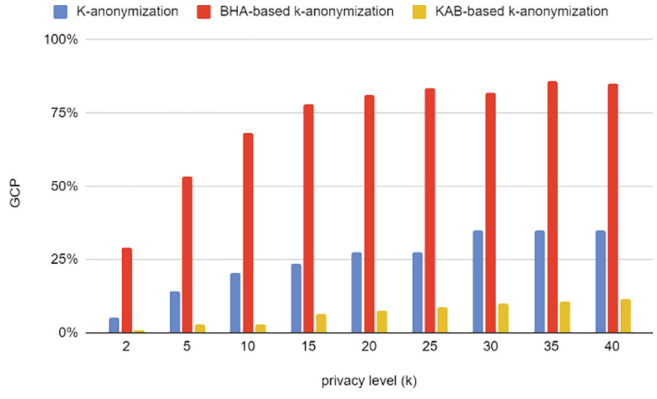

Fig. 5. Information Loss (*GCP*) vs. privacy level.

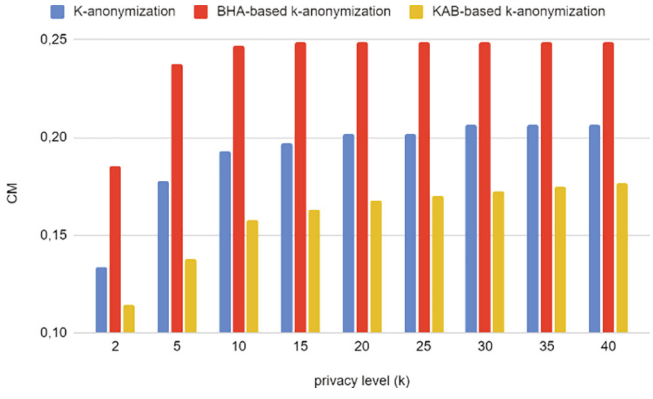

Fig. 7. Classification Error (*CM*) vs. privacy level.

contributes to decrease the distance between starting solutions and optimal solution and thus accelerate the convergence rate BHA.

Fig. 6 reports data utility, of the algorithms, with respect to C_{AVG} metric as the value of k increases. It reflects the results of information loss seen in the previous figures and can explain the results obtained in them. It shows that Mondrian Multidimensional creates equivalence classes, of sizes close to ideal case which explains, among other things, the moderate information loss introduced by K-anonymization. KAB algorithm creates equivalence classes of ideal sizes, i.e., equal to 1, which has contributed to reducing information loss of KAB-based k-anonymization. BHA-based k-anonymization creates equivalence classes of variable sizes, because the number of clusters, which determines the size of the equivalence classes, is calculated randomly; The smaller the number of clusters is, the larger the sizes of equivalence classes are and the higher the information loss it.

Fig. 7 reports data utility, of the algorithms, with respect to *CM* as the value of k increases. It shows that classification errors of the KAB-based k-anonymization and k-anonymization increase with the increase of k-value. Intuitively, the larger the class sizes are, and the greater the probability of finding classification errors is. Overall, CM introduced by BHA-based k-anonymization is the same. We can observe from the figure that KAB-based k-anonymization introduces less classification errors than the other algorithms.

4.3.2. Scalability and performance

In this experiment, we analyze the performance, of the three algorithms, with respect to execution time. To measure the execution time, we ran the experiment 10 times and used the average

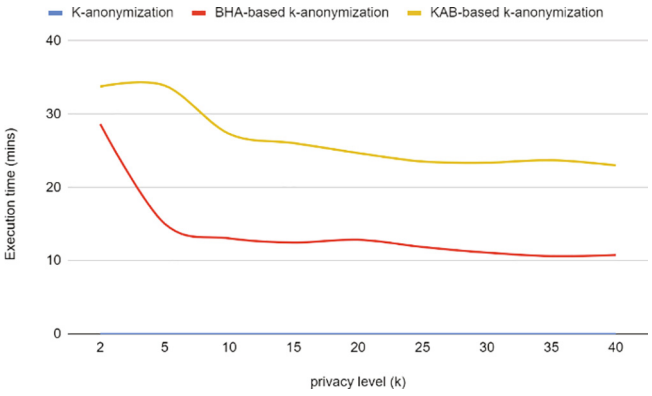

Fig. 8. Execution time vs. privacy level.

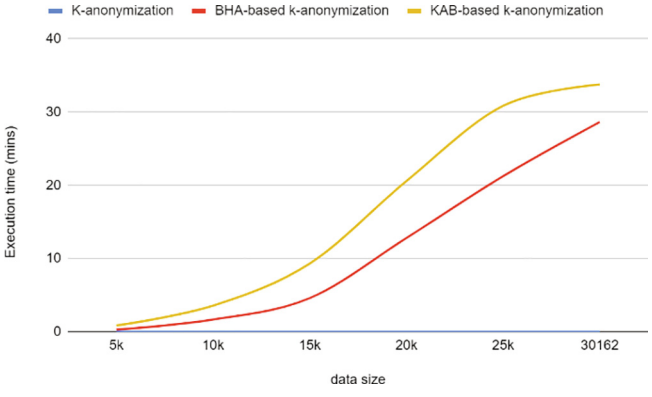

Fig. 9. Execution time vs. data size.

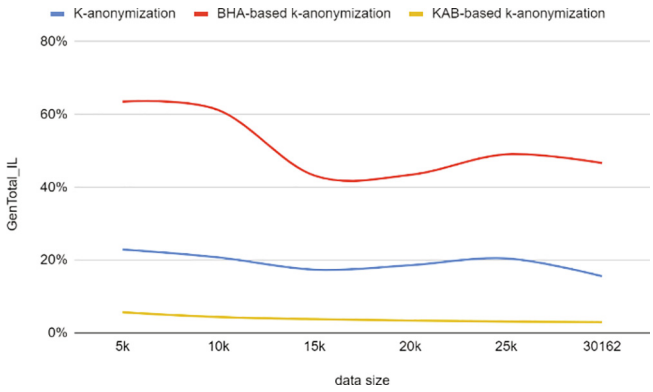

Fig. 10. Information Loss (*GenTotal_IL*) vs. Data size.

Fig. 11. Information Loss (*GCP*) vs. Data size.

value to ensure the reliability and consistency of the results. We also used subsets of the Adult dataset with different sizes. The simulation results are depicted in Fig. 8, Fig. 9, Fig. 10 and Fig. 11.

Fig. 8 reports execution time, of the algorithms, as the value of k increases. We can observe, that the execution time of BHA-based k-anonymization and KAB k-anonymization evolve in the same way and, overall, the privacy level does not have much influence on the execution time, which is rather influenced by the number of stars and of iterations. As these are the same regardless of the privacy level, execution time is, almost, constant. We can note that BHA-based k-anonymization takes less time than KAB-based k-anonymization because, unlike BHA, KAB algorithm uses metrics to form the clusters, thus consuming more time. Mondrian multidimensional is the fastest algorithm and has negligible time compared to the other two algorithms. This makes sense, because Mondrian procures lesser time in selecting the records as it does not rely on any heuristics for partitioning.

Fig. 9 reports execution time, of the algorithms, as the value of data size increase. It demonstrates that, as in Fig. 8 figure, BHA-based k-anonymization and KAB -based k-anonymization follow the same curve, with a better execution time for the former. We can notice that the computing efficiency, of KAB-based k-anonymization and BHA-based k-anonymization decreases with the increase of data size. As the dataset size grows, the algorithms need to process more records which results in increasing execution time. Mondrian multidimensional is still the best performer according to execution time, which remains negligible compared to the two other algorithms.

Fig. 10 and Fig. 11 report data utility with respect to information loss, represented by *GenTotal-IL* and *GCP* respectively, as the value of data size increases. We can observe, that the three

algorithms follow the same tendencies in the two figures, with smaller values for GCP-based information loss. The information loss of KAB-based anonymization and K-anonymization decreases slowly, compared to BHA-based anonymization, with the increase of data size. As the dataset size grows, the algorithms need to perform less generalization/suppression and then, less changes are made to the original table, which results in decrease information loss.

4.3.3. Evaluation vs clustering algorithms

As our approach has been designed to meet the limit of clustering techniques in finding optimal solution, we have compared KAB with two well-known clustering algorithms *k-members* (Byun et al., 2007), for its good data utility, and *OKA* (Lin and Wei, 2008), for its speed. We have also chosen a third algorithm: *IKA* (Zheng et al., 2018) which presents a better data utility than the other two. The comparison has been done in term of information loss, based on *GenTotal-IL*, and execution time. The simulation results are depicted in Fig. 12 and Fig. 13.

Fig. 12 and Fig. 13 report information loss and execution time respectively, of the four algorithms, with respect to different privacy level. Fig. 12, shows that KAB algorithm outperforms all clustering algorithms. The average information loss of KAB is about 14% against 32% for OKA, 28% for k-members and 25% for IKA. From Fig. 13, we can observe, that OKA is the fastest algorithm, followed by KAB, then k-members and finally IKA.

4.3.4. Summary and discussion

Overall, the simulation results indicate that our proposed algorithm performs better than comparative approaches. KAB algorithm is the best performer, in term of data utility, across different privacy level and data size. The *GenTotal_IL* of KAB algorithm is about 2 times lower than clustering algorithms, 3 times lower than *k-anonymity* and 6 times lower than BHA-based k-anonymity, on average. *GCP* is about 3.6 times lower than k-anonymity and 10.5 times lower than BHA-based k-anonymity, on average. Furthermore, unlike compared algorithms, C_{AVG} of KAB algorithm corresponds to the ideal case, which contributed to reducing the information loss introduced anonymization. With regard to classification errors (CM) rate, KAB algorithm brings a slight improvement compared to comparative approaches. We can conclude the *Classification accuracy* of our algorithm which is between 82% and 89%. Concerning execution time, KAB algorithm runs in a reasonable time. Although KAB algorithm is slower than OKA, Mondrian and BHA-based k-anonymity, we think that the execution time is acceptable considering its better performances with respect to other compared utility metrics. To highlight the general behavior of our algorithm, we have summarized all utility metrics, used in our evaluation, with respect to different privacy level and data size. The simulation results are depicted in Fig. 14 and Fig. 15.

Fig. 14 and Fig. 15 report summary of data utility of our algorithm, based on *GenTotal_IL*, *GCP*, *CM* and C_{AVG}, according to privacy level and data size, respectively. We can observe that KAB-based k-anonymization has a constant C_{AVG} and introduces an almost constant Classification Error (CM), regardless of privacy level or data size. The information loss measured with *GCP* is smaller than the one with *GenTotal-IL* because KAB algorithm is based on a clustering algorithm, using NCP metric as a distance and cost functions to generate initial population of stars.

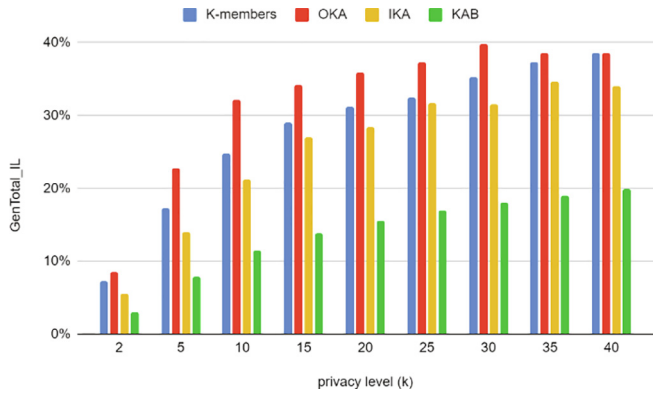

Fig. 12. Information Loss (*GenTotal_IL*) vs. privacy level.

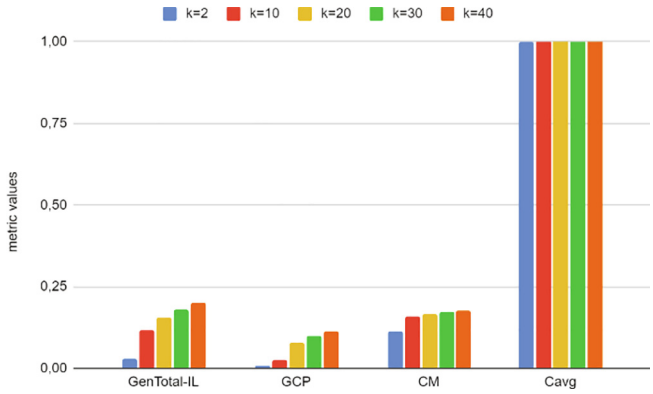

Fig. 14. Data utility across all metrics vs privacy level.

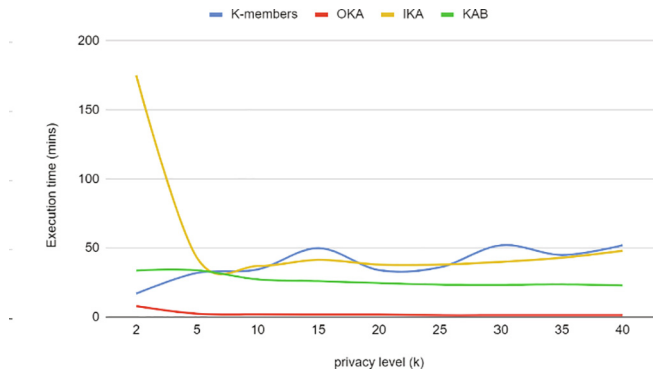

Fig. 13. Execution time vs. privacy level.

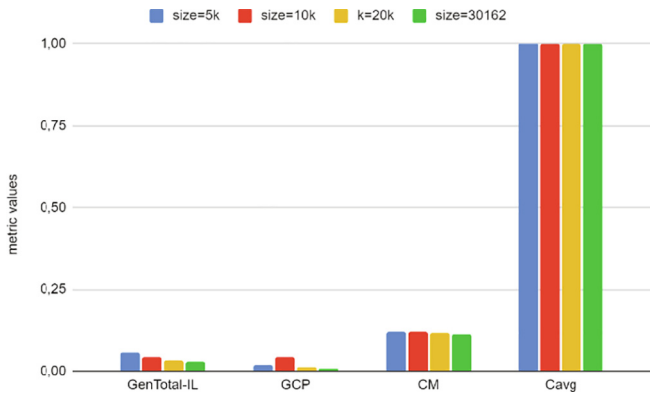

Fig. 15. Data utility across all metrics vs data size.

5. Conclusion

This work proposes a novel approach for k-anonymization based on BHA, called KAB. Its objective is to overcome the NP-hardness computational complexity of finding an optimal clustering-based k-anonymity. Our approach starts with a population of clustering-based k-anonymous candidate solutions, on which BHA is applied. To evaluate the quality of KAB algorithm, we compared it with k-anonymity, BHA-based k-anonymity and clustering-based k-anonymity techniques, in terms of data utility and scalability. The simulation results report that KAB algorithm outperforms all the compared techniques in term of data utility. However, even if the execution time of KAB is not the best, it remains acceptable in practice as anonymization is a process that, often, takes place offline. Data utility of our algorithm can be further improved by increasing the number of iterations and/or stars. However, the execution time can quickly increase, until becomes unacceptable. To manage this problem of scalability, we will propose, in a future work, a parallelized version of KAB algorithm based on MapReduce framework.

Our proposed algorithm has been applied to clustering and classification area, but it is not limited to this one. Some potential application areas of KAB are: Data mining and data analysis, machine learning, decision making and planning and engineering design optimization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Aggarwal, G., Panigrahy, R., Feder, T., Thomas, D., Kenthapadi, K., Khuller, S., Zhu, A., 2010. Achieving anonymity via clustering. ACM Trans. Algorithms (TALG) 6 (3), 1–19. https://doi.org/10.1145/1798596.1798602.

Aghdam, M.R.S., Sonehara, N., 2016. Achieving high data utility k-anonymization using similarity-based clustering model. IEICE Trans. Information Systems 99 (8). https://doi.org/10.1587/transinf.2015INP0019.

Arava, K., Lingamgunta, S., 2019. Adaptive k-anonymity approach for privacy preserving in cloud. Arabian J. Sci. Eng. 1–8. https://doi.org/10.1007/s13369-019-03999-0.

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

Bhaladhare, P.R., Jinwala, D.C., 2016a. Novel approaches for privacy preserving data mining in k-anonymity model. J. Inf. Sci. Eng., 32(1), 63-78.

Bhaladhare, P.R., Jinwala, D.C., 2016b. A clustering approach using fractional calculus-bacterial foraging optimization algorithm for k-anonymization in privacy preserving data mining. Int. J. Inf. Security Privacy (IJISP), 10(1), 45–65, doi: 0.4018/IJISP.2016010103.

Byun, J. W., Kamra, A., Bertino, E., & Li, N., 2007. Efficient k-anonymization using clustering techniques. In International Conference on Database Systems for Advanced Applications (pp. 188-200). Springer, Berlin, Heidelberg, doi: 10.1007/978-3-540-71703-4_18.

Chiu, C.C., Tsai, C.Y., 2007. A k-anonymity clustering method for effective data privacy preservation. In International Conference on Advanced Data Mining and Applications (pp. 89-99). Springer, Berlin, Heidelberg, doi: 10.1007/978-3-540-73871-8_10.

Ciriani, V., di Vimercati, S.D.C., Foresti, S., Samarati, P., Yu, T., 2006. k-Anonymity. Security in Decentralized Data Management. jajodia S., Yu T., Springer.

Das, S., Biswas, A., Dasgupta, S., Abraham, A. 2009. Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. In Foundations of computational intelligence volume 3 (pp. 23-55). Springer, Berlin, Heidelberg, doi: 10.1007/978-3-642-01085-9_2.

De Montjoye, Y.A., Radaelli, L., Singh, V.K., 2015. Unique in the shopping mall: On the reidentifiability of credit card metadata. Science 347 (6221), 536–539. https://doi.org/10.1126/science.1256297.

Fung, Benjamin C.M., Wang, Ke, Philip, S.Yu., 2007. Anonymizing classification data for privacy preservation. IEEE Trans. Knowl. Data Eng. 19 (5), 711–725. https://doi.org/10.1109/TKDE.2007.1015. 9415662, In press.

Guo, N., Yang, M., Gong, Q., Chen, Z., & Luo, J. 2019. Data anonymization based on natural equivalent class. In 2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD) (pp. 22-27). IEEE.

Hatamlou, A., 2013. Black hole: A new heuristic optimization approach for data clustering. Inf. Sci. 222, 175–184. https://doi.org/10.1016/j.ins.2012.08.023.

He, X., Chen, H., Chen, Y., Dong, Y., Wang, P., Huang, Z., 2012. Clustering-Based k-anonymity. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Berlin, Heidelberg, pp. 405–417. https://doi.org/10.1007/978-3-642-30217-6_34.

IBM Research. http://www.almaden.ibm.com/cs/quest/.

Ipums. http://ipums.org.

Iyengar, V.S., 2002. Transforming data to satisfy privacy constraints. In: In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 279–288. https://doi.org/10.1145/775047.775089.

Jagannathan, G., & Wright, R. N. 2005. Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining (pp. 593-599), doi: 10.1145/1081870.1081942.

Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data clustering: a review. ACM Computing Surveys (CSUR) 31 (3), 264–323. https://doi.org/10.1145/331499.331504.

Kabir, M.E., Wang, H., Bertino, E., 2011. Efficient systematic clustering method for k-anonymization. Acta Informatica 48 (1), 51–66. https://doi.org/10.1007/s00236-010-0131-6.

LeFevre, K., DeWitt, D.J., Ramakrishnan, R., 2005. Incognito: Efficient full-domain k-anonymity. In: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, pp. 49–60. https://doi.org/10.1145/1066157.1066164.

LeFevre, K., DeWitt, D.J., Ramakrishnan, R., 2006. Workload-aware anonymization. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 277–286. https://doi.org/10.1145/1150402.1150435.

Li, J., Wong, R. C. W., Fu, A. W. C., & Pei, J., 2006. Achieving k-anonymity by clustering in attribute hierarchical structures. In International Conference on Data Warehousing and Knowledge Discovery (pp. 405-416). Springer, Berlin, Heidelberg, doi: 10.1007/11823728_39.

Lin, J.L., Wei, M.C., 2008. An efficient clustering method for k-anonymization. In Proceedings of the 2008 international workshop on Privacy and anonymity in information society (pp. 46-50), doi: 10.1145/1379287.1379297.

Lin, J.L., Wei, M.C., 2009. Genetic algorithm-based clustering approach for k-anonymization. Expert Syst. Appl. 36 (6), 9784–9792. https://doi.org/10.1016/j.eswa.2009.02.009.

Lin, J. L., Wei, M. C., Li, C. W., & Hsieh, K. C., 2008. A hybrid method for k-anonymization. In 2008 IEEE Asia-Pacific Services Computing Conference (pp. 385-390). IEEE, https://doi.org/10.1109/APSCC.2008.65.

Loukides, G., Shao, J., 2007. Capturing data usefulness and privacy protection in k-anonymization. In: Proceedings of the 2007 ACM symposium on Applied computing, pp. 370–374. https://doi.org/10.1145/1244002.1244091.

Lunacek, M., Whitley, D., Ray, I., 2006. A crossover operator for the k-anonymity problem. In: Proceedings of the 8th annual conference on Genetic and evolutionary computation, pp. 1713–1720. https://doi.org/10.1145/1143997.1144277.

Madan, S., Goswami, P. 2019b. k-DDD measure and mapreduce based anonymity model for secured privacy-preserving big data publishing. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 27(02), 177-199, doi: 10.1142/S0218488519500089.

Madan, S., & Goswami, P., 2018. A privacy preserving scheme for big data publishing in the cloud using k-anonymization and hybridized optimization algorithm. In 2018 international conference on circuits and systems in digital enterprise technology (ICCSDET) (pp. 1-7). IEEE, doi: 10.1109/ICCSDET.2018.8821140.

Madan, S., Goswami, P., 2019a. A novel technique for privacy preservation using k-anonymization and nature inspired optimization algorithms. In Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur-India, doi: 10.2139/ssrn.3357276.

Meyerson, A., Williams, R., 2004. On the complexity of optimal k-anonymity. In Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (pp. 223-228), doi: 10.1145/1055558.1055591.

Moon, B., Jagadish, H.V., Faloutsos, C., Saltz, J.H., 2001. Analysis of the clustering properties of the hilbert space-filling curve. IEEE Trans. Knowl. Data Eng. 13 (1), 124–141. https://doi.org/10.1109/69.908985.

Ni, S., Xie, M., Qian, Q., 2017. Clustering based K-anonymity algorithm for privacy preservation. IJ Network Security 19 (6), 1062–1071.

Pramanik, M. I., Lau, R. Y., Zhang, W., 2016. K-anonymity through the enhanced clustering method. In 2016 IEEE 13th International Conference on e-Business Engineering (ICEBE) (pp. 85-91). IEEE, https://doi.org/10.1109/ICEBE.2016.024.

Run, C., Kim, H.J., Lee, D.H., Kim, C.G., Kim, K.J., 2012. Protecting privacy using k-anonymity with a hybrid search scheme. Int. J. Computer Commun. Eng. 1 (2), 155.

Sarju, S., 2015. Bigdata anonymization using one dimensional and multidimensional map reduce framework on cloud. Int. J. Database Theory Application 8 (6), 253–262 https://www.earticle.net/Article/A267622.

Sweeney, L. 1998. Datafly: A system for providing anonymity in medical data. In Database Security XI (pp. 356-381). Springer, Boston, MA, doi: 10.1007/978-0-387-35285-5_22.

Sweeney, L., 2002. k-anonymity: a model for protecting privacy. Int. J. Uncertainty Fuzziness Knowledge Based Syst. 10 (05), 557–570. https://doi.org/10.1142/S0218488502001648.

Talbi, E.G., 2009. Metaheuristics: From Design to Implementation, Vol. 74. John Wiley & Sons.

UCI machine learning repository, http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data.

Wai, E.N.C., Win, A.T., Tsai, P.W., Pan, J.S., 2017. Privacy preservation in big data by particle swarm optimization. University of Computer. Studies(Taunggyi).

Wong, W.K., Mamoulis, N., Cheung, D.W.L., 2010. Non-homogeneous generalization in privacy preserving data publishing. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, pp. 747–758. https://doi.org/10.1145/1807167.1807248.

Xu, J., Wang, W., Pei, J., Wang, X., Shi, B., Fu, A.W.C., 2006. Utility-based anonymization for privacy preservation with less information loss. ACM Sigkdd Explorations Newsletter 8 (2), 21–30. https://doi.org/10.1145/1233321.1233324.

Yan, Y., Herman, E.A., Mahmood, A., Feng, T., Xie, P., 2021. A weighted K-member clustering algorithm for K-anonymization. Computing 1–23. https://doi.org/10.1007/s00607-021-00922-0.

Zheng, W., Wang, Z., Lv, T., Ma, Y., & Jia, C., 2018. K-anonymity algorithm based on improved clustering. In International Conference on Algorithms and Architectures for Parallel Processing (pp. 462-476). Springer, Cham, doi: 10.1007/978-3-030-05054-2_36.