Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is

Real-world K-Anonymity applications: The KGen approach and its evaluation in fraudulent transactions

Daniel De Pascale [a,*], Giuseppe Cascavilla [b], Damian A. Tamburri [b], Willem-Jan Van Den Heuvel [a]

[a] *Tilburg University - Jheronimus Academy of Data Science, 's-Hertogenbosch, The Netherlands*
[b] *Eindhoven University of Technology - Jheronimus Academy of Data Science, 's-Hertogenbosch, The Netherlands*

A R T I C L E   I N F O

Article history:
Received 15 July 2021
Received in revised form 30 May 2022
Accepted 31 January 2023
Available online 8 February 2023
Recommended by Matthias Weidlich

Keywords:
K-Anonymity
Privacy-by design
Data-intensive applications design & operations
Big data
Scalability

A B S T R A C T

K-Anonymity is a property for the measurement, management, and governance of the data anonymization. Many implementations of k-anonymity have been described in state of the art, but most of them are not practically usable over a large number of attributes in a "Big" dataset, i.e., a dataset drawing from Big Data. To address this significant shortcoming, we introduce and evaluate KGen, an approach to K-anonymity featuring meta-heuristics, specifically, Genetic Algorithms to compute a permutation of the dataset which is both K-anonymized and still usable for further processing, e.g., for private-by-design analytics. KGen promotes such a meta-heuristic approach since it can solve the problem by finding a pseudo-optimal solution in a reasonable time over a considerable load of input. KGen allows the data manager to guarantee a high anonymity level while preserving the usability and preventing loss of information entropy over the data. Differently from other approaches that provide optimal global solutions compatible with smaller datasets, KGen works properly also over Big datasets while still providing a good-enough K-anonymized but still processable dataset. Evaluation results show how our approach can still work efficiently on a real world dataset, provided by Dutch Tax Authority, with 47 attributes (i.e., the columns of the dataset to be anonymized) and over 1.5K+ observations (i.e., the rows of that dataset), as well as on a dataset with 97 attributes and over 3942 observations.

1. Introduction

The amount of data being produced and processed, both online and offline, is exponentially increasing, and so is the costly consumption of resources to carry such processing to fruition. On the one hand, maintaining data anonymity is a must-have, especially in sight of the severe sanctions connected to potential violations of the General Data Protection Regulation [1]. On the other hand, many agencies want or need to exploit such data for commercial purposes or public safety and security, implying that data should be usable.

It is, hence, fundamental to provide fast and reliable techniques to the stakeholders that guarantee the privacy and anonymity of the data and, at the same time, maintain the data's usefulness. This paper introduces and evaluates KGen, an approach to state-of-the-art privacy-preserving technologies implemented using a metaheuristic-based approach.

The process starts with a dataset, and, through an anonymization process, it provides a dataset anonymized. At the core of KGen is the most widely known k-anonymity approach to anonymization [2]. K-anonymity is defined as the condition wherefore, for each record in that dataset, there are at least other k-1 records indistinguishable from it.

The K-anonymity property is classified as an NP-Hard problem, as proved by Meyerson et al. [3]. Aggarwal [4] shows the problem raised by any K-anonymity algorithms applied with large datasets. When a dataset contains many attributes to anonymize, it becomes difficult to anonymize them without an unacceptably high amount of information loss.

Though it is not possible to anonymize a large dataset without loss of information, with KGen we aim to provide an anonymized dataset on the K-Anonymity property. In the scope of KGen, K-anonymity needs to be traded-off against the usefulness of data. At the same time, several algorithms address this problem, providing an optimal solution [2,5–8], all known approaches merely work on a relatively small number of attributes with a reduced level of generalization for each attribute. While the number of attributes that need to be anonymized grows, complexity increases to obtain a usable dataset.

To account for the trade-off mentioned above, KGen features an approach based on Genetic Algorithms [9] providing a pseudo-optimal solution in a time useful for practical usage We

* Corresponding author.
E-mail address: d.de.pascale@tue.nl (D.D. Pascale).

compared KGen with other approaches from the state-of-the-art in order to validate its results.

The main goal of this work is to provide an approach useful in an industrial context. To this end, we defined the following research question:

***Main RQ*$_1$**: *Is the performance of the proposed approach useful for its intended stakeholders?*

To answer the main research question, we outlined three subsequent research questions:

RQ1 *Does* KGen *perform when compared to state-of-the-art approaches?* To address this RQ We first compared our approach to existing ones by means of execution time to generate the best-anonymized dataset.

RQ2 *How accurate are* KGen *solutions compared to state-of-the-art approaches?* To answer this question, we proposed a measure of accuracy to measure how the pseudo-optimal solution is far from the optimal solution.

RQ3 *What is the quality of* KGen *solution?* We measured the quality of a solution using generalization and suppression metrics defined in the state-of-the-art and discussed in Section 2.3.

Moreover, to evaluate the applicability in a large context scenario, we outlined a followup main research question:

***Main RQ*$_2$**: *To what extent can the case-specific evaluation generalize to much larger datasets?*

Therefore, in order to evaluate KGen in an industrial context, the approach was used a real-world sample dataset provided by the Dutch Tax Authority for fraudulent transactions. The evaluation aims at accounting for KGen's real-life applicability. Moreover, we led a second experimentation, using the "c2k_data_comma.csv" dataset [10] to prove the applicability of the approach using a large dataset. The experimentation has been done using OLA [7], a state-of-the-art approach for the dataset k-anonymization, a brute force approach and a meta-heuristic random approach to evaluate the goodness of KGen. The experimentation reveals promising results and shows that KGen is an approach capable of providing a good-enough solution in less than 5 h:05 m:40 s (the worst case recorded with the "c2k_data_comma.csv" dataset and 25 quasi-identifiers attributes to anonymize). KGen showed to be able to find results up to 25 attributes to anonymize, under the limited-time set of 15 h differently from other approaches that provided results up to 7 attributes in much more time. Moreover, KGen demonstrates to preserve the quality of data correctly, a critical feature in order to keep the dataset qualitatively usable.

From a software and information systems engineering perspective the concrete usage of our proposed method KGen is twofold: (a) privacy-aware data-intensive applications [11,12] could be designed using KGen as a middleware to anonymize datasets before processing automatically; (b) compliance officers can use KGen to experiment with processed and non-processed data to quantify the extent of privacy "damage" carried out by data processors.

The remaining part of the paper is organized as follows. Section 2 introduces the state of the art of the anonymization process and the main works related to anonymization. Section 3 introduces KGen, explaining all its components. Section 4 outlines the research design of the work. It describes the dataset used for the experimentation, the metrics used to evaluate the RQs illustrated above and the algorithms used for the comparison study. The results o this work are shown in Section 5. Section 6 contains the discussion above the results obtained in Section 5. In Section 7 are discussed the threats to validity found in KGen. Lastly, Section 8 summarizes the main contributions of KGen and sketches future research directions.

2. Background and related work

This section is organized in three main subsections: the first one describes the anonymization process to allow a better understanding of the purposes behind this work; the second subsection explains what a genetic algorithm is — hence laying the technical foundations behind the metaheuristic underlying KGen. Third, finally, we showcase the known k-anonymity implementations in the state of the art to which KGen can be compared.

2.1. Anonymization

The anonymization process starts from a given dataset and generates an anonymous dataset. A dataset is composed of multiple observations with several different attributes. From a privacy perspective, there are two different kinds of attributes in any dataset [2]:

- **Identifiers.** An Identifier attribute can uniquely identify a row in the dataset. In the anonymization process, these are suppressed (this process is explained more in-depth in the next section).
- **Quasi Identifiers.** Are the set of attributes that can be superimposed with external information to reveal an individual's identity [13]. Examples of common quasi-identifiers are [14–17]: dates (such as birth, death, admission, discharge, visit, and specimen collection), locations (such as postal codes, hospital names, and regions), race, ethnicity, languages spoken, aboriginal status, and gender.

During the anonymization process, the data is changed by either removing or suppressing all identifiers [2]. This is essential to prevent reverting to the original dataset. Thus, nullifying the anonymization process. Stemming from this assumption, the only data that needs to be (partially)-anonymized while simultaneously ensuring the highest amount of information usability as possible are the quasi-identifiers.

Therefore, the central part of the anonymization process revolves around two main factors (1) the anonymization of those attributes, quasi-identifiers, and (2) finding the optimal trade-off between them. Hence, making it hard to uniquely identify rows in a dataset by removing information and maximizing the usefulness of the data, keeping as much as possible intact. In turn, the usability of the dataset can be measured using the loss of information metrics [7]. Metrics that are used to evaluate the goodness of a possible k-anonymous are explained below.

2.2. K-Anonymity

To guarantee anonymity KGen harnesses the concept of k-anonymity [2]. A dataset is called *k-anonymous* if a single row is indistinguishable from, at least, other k-1 rows in the dataset.

Definition. Let $T(A_1, \dots, A_n)$ be a table and $QI_T(A_1, \dots, A_j)$ be all the quasi-identifiers of that table. T is said k-anonymous if, for each row of T, there are at least k-1 rows equals to that row (for a total of k indistinguishable rows).

Table 2 shows an example of anonymization of the dataset in Table 1. The quasi-identifiers have been anonymized in order to guarantee the anonymization. Applying different levels of generalization for all quasi-identifier attributes, it is possible to guarantee the anonymization with a certain degree of remaining usability of the same dataset. Table 2, for example, shows a k-anonymous dataset with a level of k = 2.

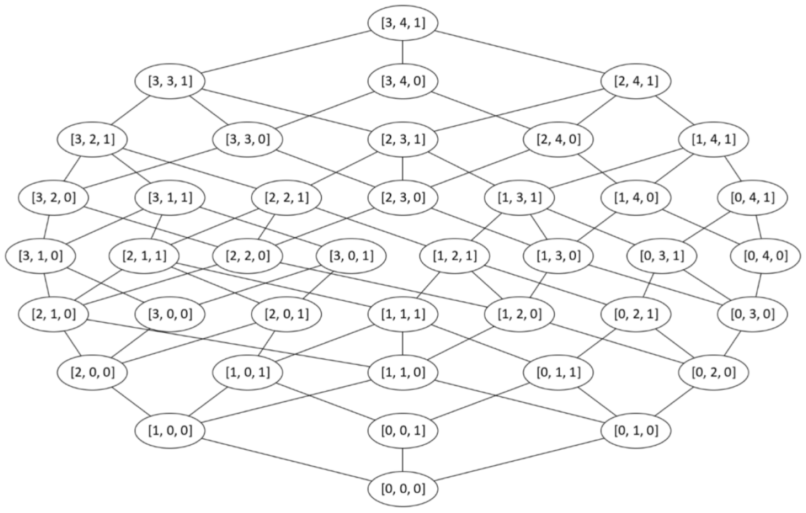


Fig. 1. Example of lattice (Age–Postcode–Gender). Each node contain a possible level of generalization, for each attribute, and is connected to other nodes that can be reached increasing or decreasing by one a single level of generalization of a given node.

Table 1
Original dataset. The attribute Name is an Identifier. Instead age, Gender and Postcode are Quasi-Identifiers.

Name	Age	Gender	Postcode	Crime
Alice	24	F	80015	Assault
Max	28	M	80019	Kidnapping
Laurel	42	F	85073	Homicide
Frank	49	M	85071	Rape

Table 2
Dataset k-anonymized. Considering the QI, the number of indistinguishable rows are two. So, the dataset is k-anonymized (k = 2).

Name	Age	Gender	Postcode	Crime
*****	20–30	P	8001*	Assault
*****	20–30	P	8001*	Kidnapping
*****	40–50	P	8507*	Homicide
*****	40–50	P	8507*	Rape

2.3. K-Anonymity operators

As mentioned before, the anonymization process revolves around the anonymization of attributes. State of the art offers several approaches, mainly around four different anonymization techniques, namely, generalization, suppression, anatomization and perturbation [2,18].

- **Generalization.** Given an attribute, its level of anonymity can be represented as a hierarchy (Fig. 2). The higher the level of generalization of an attribute, the more the dataset is generalized, ensuring a high level of anonymization and a correspondingly low level of usability.
- **Suppression.** If a dataset is not k-anonymized because there is only a single row that does not allow to satisfy the k-anonymity conditions, it is possible to suppress that single row to have a k-anonymized dataset.
- **Anatomization.** Unlike generalization and suppression, the anatomization operator does not work on QI and sensitive data, but it works on the relationship between them. The operator splits the QI and the sensitive data into two different tables. To preserve the relationship between the two groups, each table have a common attribute, groupID, All rows in the same group have the same groupID [18].
- **Perturbation.** The perturbation replaces the original values with synthetic data. The new record generated does not correspond to a real-world record. In this way, for the attacker is not possible to recover sensitive data, starting from the data published.

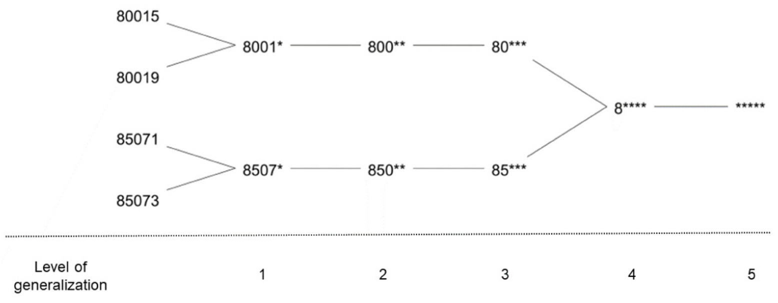
KGen uses only generalization and suppression operators because, in the comparison study done in this work, the state-of-the-art approach chosen uses only the two operators mentioned above.

Generalization works on the generalization of all values of a single attribute. Thus, no information is lost, but the entire dataset is modified. Conversely, suppression works at a local level, its approach revolving around the removal of entire rows, with the remaining data left unchanged [2].

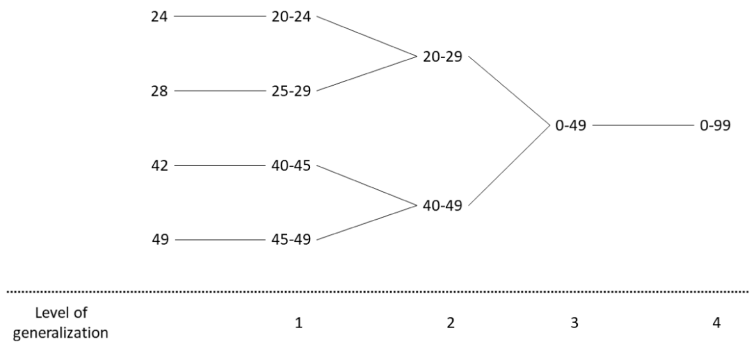
In both cases, however, it is always possible to compute the generalization hierarchy of all the attributes as represented by a lattice (i.e., repeating arrangement of points, see Fig. 1) [7]. Thus, a node of the lattice represents a possible anonymized dataset containing the level of generalization of each quasi-identifier attribute. The lattice shown in Fig. 1 is the representation of all possible configurations of the dataset in Table 1. The minimum node in a lattice is the representation of a dataset with all quasi-identifier attributes not anonymized (node (000) of Fig. 1); the maximum node, instead, is the representation of a dataset completely anonymized because contains the maximum level of generalization of each quasi-identifier attribute (node (341) of Fig. 1). Each arrow represents a possible generalization path taken through the lattice. Thus, the height of a lattice is equaled to the number of steps that, from the minimum node, are necessary to reach the maximum node, increasing one by one the level of generalization of a quasi-identifier attribute. Climb up the lattice allows to have a higher level of anonymization of a dataset but a lower utility (this concept is explained in Section 2.4).

Every path starting from the minimum node to the maximum node is called strategy path. For example, in Fig. 1 the path [(000), (001), (011), (021), (031), (041), (141), (241), (341)] is a strategy path.

All strategy paths share the same starting node (the minimum node of the lattice) and final node (the maximum node of the lattice). As explained before, since the maximum node represents a dataset completely anonymized, all strategy paths ensure the existence of at least one k-anonymized node. 3 In the lattice, every node could represent a k-anonymized dataset and, among these, only one represents the optimal global solution. So, the goal of k-anonymity is to find it in a reasonable time.



(a) Generalization: POSTCODE



(b) Generalization: AGE

Fig. 2. Generalization hierarchy of two quasi-identifiers attributes.

2.4. Measuring loss of information

Using generalization and suppression, all possible datasets in the lattice can be possible solutions. The way of preferring a dataset to another for KGen is to select the dataset whose information is most useful in generalization. A dataset with more generalization or more suppression has less information and, hence, lower usability. KGen uses metrics to measure the usability of an input dataset using different metrics of information loss. The significant metrics for information loss are outlined below. Subsequently, a selection is made and illustrated for KGen.

One metric for the level of information loss was proposed by Samarati [2]. The idea of the proposed approach is to take the k-anonymity node with a minimum height level in the lattice. So, for example, if in the lattice showed in Fig. 1 nodes (100) and (001) are both k-anonymized, using this metric, they have the same level of loss of information because they have the same height level in the lattice. However, the height lattice is not a helpful metric since it does not consider each attribute's maximum level of generalization. In the previous example, there are two nodes: the first one has only the first attribute generalized at level 1 of a maximum of 4 levels. Instead, the second one has the last attribute that, in this case, is completely anonymized. Moreover, with the first metric presented, they have the same level of loss of information. Sweeney in [6,19] takes into consideration as information metric also the level of generalization of each attribute. The aim is to evaluate, for each attribute, its level of generalization, called "LOG", using this formula:

$$LOG_i = \frac{log_i}{Hlog_i} \quad \forall i = 1, \ldots, N \tag{1}$$

where log is the actual level of generalization of the ith quasi-identifier, Hlog is the height of the generalization hierarchy of the ith quasi-identifier and N is the total number of quasi-identifier attributes in the dataset. Hence, the level of generalization of a single node is given by the average of all LOG values calculated.

$$LOG = \frac{\sum_{i=1}^{N} LOG_i}{N} \tag{2}$$

For example, the node [1, 0, 0], representation of the attributes Age/Postcode/Gender with a generalization hierarchy's height of, respectively, 4, 5 and 1, has a LOG level of ($\frac{1}{(4)}$ + $\frac{0}{(5)}$ + $\frac{0}{(1)}$) / 3 = 0.083. Instead, the node [0, 0, 1] has a LOG level of ($\frac{0}{(4)}$ + $\frac{0}{(5)}$ + $\frac{1}{(1)}$) / 3 = 0.33. With this metric, the node position in the lattice and the level of generalization of each attribute are taken into account. KGen uses this decaying information metric to find the dataset with the most information and the highest anonymization concurrently.

2.5. K-Anonymity complexity

Different works prove that an optimal k-anonymization algorithm is an NP-Hard problem. Meyerson et al. [3] provide a demonstration on the complexity classification of the problem, finding that not only the k-anonymity algorithm is NP-Hard, but also the k-anonymization with suppression of different attributes is NP-Hard.

Aggarwal [4] shows that the k-anonymity complexity is highly dependent on the size of the problem and that it is impossible to apply the k-anonymization property on a dataset with lots of quasi-identifier attributes with an acceptable level of information loss.

Sun et al. [20] introduce two variants of the k-anonymization problem, the Restricted K-anonymity problem and the Restricted

K-anonymity problem on attributes. They proved that both of them are *NP*-Hard for $k \geq 3$, but, on the positive side, they developed a polynomial solution for the k-anonymization problem with $k = 2$.

2.6. Genetic algorithms: An overview

Genetic algorithms are simulations of natural selection, used to solve optimization problems [21] such as the one reflected by KGen. The natural selection process inspires genetic algorithms, and their workings and architecture reflect the natural process of reproduction, proliferation, and selection. More specifically, starting from an initial population, the algorithm selects, with a function used to measure the goodness of an individual, the best individuals and, from them, produces new individuals. Then, the old and the new population are re-evaluated to see which of them survives to the next generation. This process goes on until a stop condition is satisfied. In order to better explain this process, it is essential to describe the main components of a genetic algorithm:

Solution encoding: a good solution representation plays a key role in a genetic algorithm because all future evaluations are applied to all solutions. So, if a solution is easy to evaluate, then the entire algorithm's complexity is low. A solution typically consists in an array of values. As a first step, a random population is generated. Then the algorithm tries to improve its solutions in order to find the best solution.

Fitness function: in implementing a genetic algorithm, a key role is played by the complexity of the fitness function. A fitness function is a good representation of the objective to achieve. If it has low complexity, then the entire algorithm has a lower complexity. The choice of the proper fitness function should be made together with the choice on the solution encoding because they are highly correlated. The fitness function is directly applied to the solution, so if they are incompatible, then the evaluation process is more complicated.

Genetic operator: Genetic operators are functions that automatically allow the generation of new chromosomes, starting from the previous population. There are three different types of operators: selection, an operator used to find the best chromosome in the population; crossover, a "mating process" applied to two chromosomes to generate two new chromosomes; mutation, operator used to mutate a single chromosome to avoid the genetic algorithm convergence into a local optimal solution [21].

2.7. Related work

There are many works on k-anonymization and its practical implementation. Samarati et al. [2] provide a k-minimal generalization algorithm to apply a binary search to find all k-anonymous node, selecting all nodes with the least steps as solutions. If there is more than one node as a solution, the algorithm selects one randomly or using other criteria, as the information loss. However, the node with the lowest distance vector is not guaranteed the optimal solution because they could be other nodes with a higher distance value but with a lower level of information loss. For this reason, the algorithm does not provide the optimal global solution.

Similarly, the Datafly algorithm adopts a heuristic based on the attribute [5,6]. The most distinct attribute is taken into account as how next generalized attribute. The process continues with new distinct attributes that do not satisfy k-anonymous until the k-anonymous criteria are satisfied. This approach does not guarantee the minimum k-anonymous solution, however, the found solution is always k-anonymous.

Kirsten et al. Incognito exploits a bottom-up approach with a breadth-first strategy to navigate the lattice to find all k-minimal distance vectors [8]. After detecting all vectors, the algorithm calculates their information loss to select the solution with the least information loss as the optimal solution. This algorithm can find, in this way, a global optimum.

Besides, the Optimal Lattice Anonymization (OLA) The OLA algorithm is an improvement of Incognito and Datafly algorithms [7]. All the anonymization processes, as shown in Fig. 1, may be represented as a lattice. The goal of the OLA algorithm is to find the optimal node in the lattice that must be k-anonymous and with minimum loss of information. The approach embraces a binary search algorithm for each strategy path. When the optimal node in a strategy path is reached, the algorithm commences to analyze the next strategy hub, and so on. In the end, the algorithm holds a list with all k-minimal nodes for each strategy path. At this point, it is chosen only the node with the minimum information loss. Thus, OLA, as Incognito, can provide a globally optimal solution.

Bayardo et al. [22] present a new approach to explore the space of possible combinations developing data-management strategies to reduce reliance on expensive operations. They can find an optimal solution under two representative cost measures and a wide range of k. Moreover, they can provide good anonymizations where the input data or input parameters preclude finding an optimal solution in a reasonable time.

Lyengar shows an example of a Genetic algorithm applied on the k-anonymity problem [23]. It seems to generate good results, as we can see from the experimentation done in their work. Nevertheless, they considered only a dataset with eight quasi-identifier attributes, lacking more considerable experimentation.

Among all of these k-anonymization algorithms, only OLA and Bayardo's algorithm proved that their results are better than the others (Datafly, Samarati's algorithm) [7,22]. For this work, we realized a comparison only with OLA because we found different implementations of it, differently from Bayardo's approach. Furthermore, we did not realize a comparison with Lyengar's GA because of lacking a pseudo-code of the algorithm or a repository with their work.

3. Scalable K-Anonymization: KGen explanation

This section describes KGen from a technical perspective, elaborating (1) the general KGen architecture; (2) the KGen lattice preprocessing; (3) solution encoding; (4) solution fitness; (5) genetic operators.

3.1. KGen architecture

An overview of KGen architecture is shown in Fig. 3. Processing of data starts with an input phase in which KGen receives a dataset to anonymize along with configuration parameters such: (a) the generalization strategy to be adopted; (b) attributes' information type, that is, whether they are Identifiers or Quasi-Identifiers. As explained by Samarati et al. [2], there are different generalization strategies, assuming the existence of different domains, including generalized values and mapping between each domain and domains generalization of it. Thus, for example, the postcode can be generalized, dropping, from the right, the least significant value (as shown in Fig. 2(a)).

The subsequent processing phase is the core of the KGen approach. An overview of this phase is provided in Algorithm 1. The first step of KGen processing phase is the preprocessing of the lattice for size reduction. The next step is an iteration of the KGen Genetic Algorithm (GA) implementation. In the KGen-GA step, KGen tries to converge to the optimal solution following the GA meta-heuristic approach recapped in Section 2. The output of the processing phase is the k-anonymized dataset using the best solution provided by KGen.

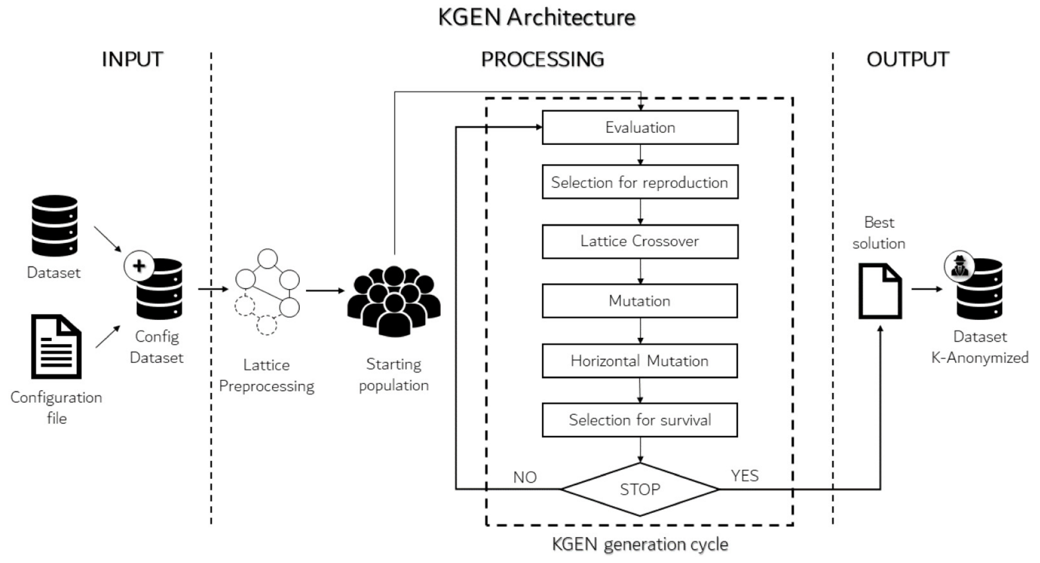


Fig. 3. KGEN Pipeline. It is divided into three steps (separated bu dotted vertical lines), input, processing, and output; the KGEN-GA architecture is described in the processing step.

Algorithm 1 KGEN Algorithm

```
    Input: Dataset
    Output: Dataset anonymized
 1: procedure KGEN ALGORITHM
 2:     $bounds \leftarrow LatticePreprocessing(dataset)$
 3:     $t \leftarrow 0$
 4:     $P_t \leftarrow initRandomPopulation(bounds)$
 5:     $evaluate(P_t)$
 6:     while $evaluation < maxEvaluations$ do
 7:         $Offsprings \leftarrow$ empty offspring list
 8:         for $(i = 0; i < populationSize; i+ = 2)$ do
 9:             $parents \leftarrow selection(Population)$
10:             $tmpOffsprings \leftarrow crossover(parents)$
11:             $mutation(tmpOffsprings)$
12:             $horizontalMutation(tmpOffsprings)$
13:             $Offsprings.add(tmpOffsprings)$
14:             $evaluation = evaluation + 2$
15:         $evaluate(Offsprings)$
16:         $P_t \leftarrow P_t \cup Offsprings$
17:         $P_{t+1} \leftarrow selection(P_t)$
18:         $t \leftarrow t + 1$
19:     $S \leftarrow minLOGSolution(P_t)$
20:     $newDataset \leftarrow anonymize(dataset, S)$
21: return $newDataset$
```

Table 3
Example of the entire lattice reduction process.

Age	Postcode	Gender
24	80015	F
28	80019	M
42	85073	F
49	85071	M

(a) Original dataset not anonymized. The attributes Age, Gender and Postcode are Quasi-Identifiers.

Age	Postcode	Gender
24	80015	F
28	80019	M
42	85073	F
49	85071	M

(b) First step of the reduction process. The original dataset is split into n dataset, where n is the number of quasi-identifiers in the original dataset. Each of these new datasets contain only one of these quasi-identifiers.

Age	Postcode	Gender
20–29	8001*	F
20–29	8001*	M
40–49	8507*	F
40–49	8507*	M
(c) Age: log 2.	(d) PC: log 1.	(e) Gender: log 0.

(f) Second step of the reduction process. Each of the datasets generates previously has been anonymized up to reach the minimum level of anonymization. The level of generalization of each of these datasets represent the new minimum level of generalization of the lattice.

3.2. Lattice preprocessing

The lattice reduction is the first step of KGen execution. It is based on the lattice pruning technique used in [8]. This step aims at removing the complexity given by the generation of a lattice at the expense of introducing an acceptable permutation computational cost. It reduces the lattice size, thus the complexity of the k-anonymization algorithm. The size-reduction process exemplified in Fig. 1 shows an example of a non-reduced lattice. In this example, the minimum node is <0, 0, 0> and the maximum node is <3, 4, 1>. The reduction technique is recapped in Table 3, parts from (a) to (f); KGen slices the dataset into N vectors, one per quasi-identifier (Table 3b), and validates the k-anonymity property iteratively on each vector thus obtained, until a new minimum level of generalization is found (Table 3f). The idea is that if at least one quasi-identifier attribute is not k-anonymized, then the entire dataset cannot be anonymized too. Hence, the computational cost for the execution of KGen on nodes containing quasi-identifiers not anonymized is meaningless. Although this approach poses limitations when anonymizing by suppression, such limitations are addressed in the Threats to Validity section, see Section 7.

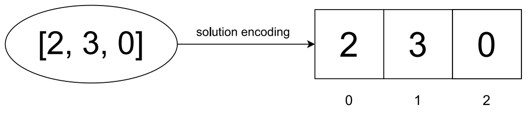


Fig. 4. Solution encoding of the lattice node ⟨2, 3, 0⟩.

3.3. Solution encoding

A genetic algorithm aims to find the best pseudo-optimal solution in a reasonable time. In this case, a solution is the representation of a node in the lattice (see Fig. 1) that represents its level of generalization. In KGen, a solution is represented as an array of numbers, where in the ith position of the array contains the value of the ith attribute in the lattice node. Fig. 4 shows the solution encoding of the lattice node Age/Postcode/Gender <2, 3, 0>. In the solution encoding process, the level of generalization values of Age, Postcode and Gender are respectively put in positions 0, 1 and 2. In a Genetic algorithm, the initial population is initialized randomly.

3.4. Fitness functions

Every Genetic Algorithm needs to define its fitness function. This function allows evaluating, for each iteration, all generated solutions. As discussed in Section 2, there are two metrics for the evaluation of a single node, namely, (a) k-anonymity and (b) loss of information. In KGen, the loss of information is the only metric used to evaluate the fitness of a solution. For every fitting solution, k-anonymity is evaluated to see if a solution is feasible or not. Thus, the goal of the KGen fitness function is to find the lowest value of loss of information of a node while ensuring, at the same time, the k-anonymity property.

3.4.1. Implementing K-Anonymity in KGen

We implemented KGen using the improved algorithm for k-anonymity presented by Zhang et al. [24]. They propose a technique for improving the k-anonymity implementation by providing a new structure for the generalization hierarchy, namely, a support map. A support map provides a structure in which each indistinguishable value is associated with its level of generalization, and all the rows contain an equal value. Table 4 shows an example of a support map, applied on two quasi-identifier attributes, Age and Postcode. With the support map technique, for each attribute, there is a related support map. This support map contains all values referred to that attribute, including all their generalization versions, and, for each value, they memorize its level of generalization and all rows that contain that value. In Table 4a, the value 24 has a level of generalization 0 and is included only in the first row. Differently, its generalization 20–29 has a level of generalization 2 and can be found in rows 1 and 2. In this way, to see if a dataset is k-anonymized, the algorithm intersects all value rows of a given level of generalization to see if there are no rows less than k. In Table 4, for example, with the intersection of log 2 of age and log 1 of Postcode, we have two groups of rows: the first one containing rows 1 and 2, that contain values 20–29 and 8001*; the last one, that contains rows 3 and 4 with values 40–49 and 8507*.

3.4.2. Implementing loss of information in KGen

As discussed in Section 2.4, KGen implements the precision criterion, as information loss metric. Each possible solution is evaluated with the precision Formula (1). The goal of KGen's genetic algorithm is to minimize the precision of a solution to find the best k-anonymized solution with the least precision.

Table 4
Example of support map for the quasi-identifiers AGE and Postcode.

Value	log	Rows
24	0	[1]
28	0	[2]
42	0	[3]
49	0	[4]
20–24	1	[1]
25–29	1	[2]
40–44	1	[3]
45–49	1	[4]
20–29	2	[1, 2]
40–49	2	[3, 4]
0–49	3	[1, 2, 3, 4]
0–99	4	[1, 2, 3, 4]

(a) Age support map.

Value	log	Rows
80015	0	[1]
80019	0	[2]
85073	0	[3]
85071	0	[4]
8001*	1	[1, 2]
8507*	1	[3, 4]
800**	2	[1, 2]
850**	2	[3, 4]
80***	3	[1, 2]
85***	3	[3, 4]
8****	4	[1, 2, 3, 4]
*****	5	[1, 2, 3, 4]

(b) Postcode support map.

3.5. Genetic operators

For the implementation of the KGen-GA approach, the following operators are provided.

Selection. For the selection operator, KGen uses the *Tournament Selection* operator [25] with penalty. The Tournament Selection is used to select the fittest candidate for the current generation. This operator assigns a probability to each solution based on two criteria: the fitness value and the penalty of a solution. The fitness value, in our case, is the loss of information metric. Instead, the penalty is calculated as follows: when a new solution is generated, its penalty value is 0. Suppose this solution survives going to the next generation, its penalty increases by 1. The maximum value reachable is 9. Otherwise, with a value of 10, the penalty decreases the probability to 0. The concept is that the more a solution survives, the more the probability to be chosen decreases. Therefore, the penalty is used as a weight for solution optimality. An example of this process is shown in Fig. 5 (in the figure, the data regarding the level of generalization (LOG) and the penalty are chosen randomly, just to explain the process behind the KGen selection operator). The probability of selection is calculated using this formula:

$$P(S_j) = \frac{log(S_j) * w_j}{\sum_{i=1}^{n}(log(S_i) * w_i)} \tag{3}$$

Crossover. KGen provides its own Crossover implementation, based on the double point crossover defined in [26]. Fig. 6(a) shows the first step: (i) the PARENTS selected with the selection operation, (ii) on top of them the crossover generates two new chromosomes, one with the highest value extracted from PARENTS and the second one with the lowest values extracted from PARENTS.

Subsequently, three possible scenarios manifest:

- **Case 1.** Both parents are k-anonymized. In this case, the maximum node is anonymized because, by definition of

LOG	Penalty	LOG weighted	Percentage
0.85	0	(1-0) * (0.85) = 0.85	0.51
0.44	3	(1-0.3) * (0.44) = 0.308	0.19
0.55	1	(1-0.1) * (0.55) = 0.495	0.3

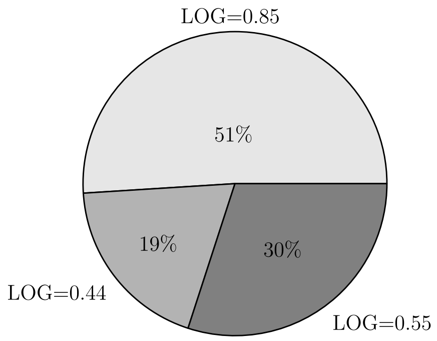
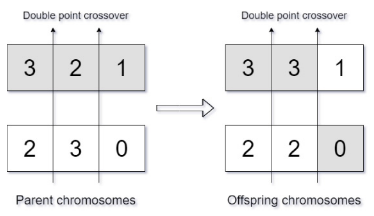
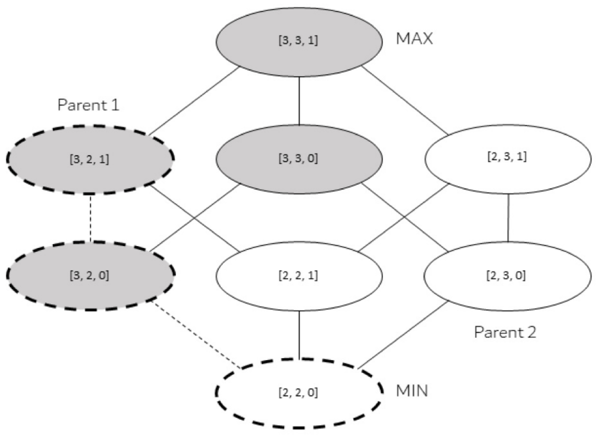


Fig. 5. Selection process, based on LOG metric as fitness function. Based on their LOG value and their penalty, the selection generates all probabilities. The pie chart shows the probability to choose a single solution.



(a) Offspring generation.



(b) Crossover lattice, an example with only one k-anonymized parent.

Fig. 6. Example of crossover operator with only one of the parent k-anonymized. In this case, all nodes with dashed lines represent a possible final offspring of the crossover.

strategy path, all nodes after a k-anonymized node are also k-anonymized. If also the minimum node is anonymized, add it to the final offspring. Otherwise, the algorithm adds a random node between the minimum node and the first parent node and another random node between the minimum node and the second parent node;

- **Case 2.** Both parents are not k-anonymized. In this case, the minimum node is not k-anonymized, and the final offspring is the maximum node;
- **Case 3.** Only one of parents is k-anonymized. The minimum node is not k-anonymized, and the maximum node is k-anonymized. In this case, the last offspring is a random node between the minimum node and the k-anonymized parent.

An example of case 3 is shown in Fig. 6, while Fig. 6(a) shows the generation of the minimum and maximum nodes. Finally, Fig. 6(b) shows the crossover lattice that contains all the possible crossover's offspring. In this case, only nodes with dashed lines are considered since they represent the random solution discussed previously.

Mutation. In this case, KGen uses two different Mutation techniques:

- **Standard mutation.** a classic mutation operator, inherited from the approach in [9]. This approach changes a single value of the chromosome and allows to change a possible solution with another one from the same strategy path. This operator needs to guarantee the principle of exploitation [27] since this principle allows a solution to move up or down its strategy path;
- **Horizontal mutation.** this operator allows the genetic algorithm to change a solution with another solution of a different strategy path. In this way, it is possible to guarantee the exploration criteria. In order to change the strategy path, it is necessary to change more than one value of the solution and, to avoid having a solution in the same strategy path, it is necessary, alternatively, increase and decrease the chosen value, with a value between the minimum value (or maximum value in case we need to increase the value) and the actual node. An example of Horizontal mutation is shown below:
  Example
  Minimum solution: 0 0 0 0 0
  Actual solution: 2 2 2 2 2
  Maximum solution: 4 4 4 4 4
  Percentage of values to mutate: 50%. In this case it means that we need to mutate 2 values
  Random indexes chosen: 2, 3
  Algorithm: The value in the index 2 can choose a random value between its value and its maximum (so, from 2 to 4). The value in the index 3 can choose, instead, a value between 2 and 0, its minimum.
  Possible mutate solution: 2 2 3 0 2
  This procedure of increasing and decreasing iteratively must keeps going on until all indexes chosen have been mutated.

Table 5
Generalization strategies applied on the K-Anonymity problem.

Generalization techniques	
NUMBER	Range generalization (3 –>0−5)
STRING	Star generalization (NL805 –>NL80*)
DATE	Date generalization (01/01/1970 ->01/1970 –>1970)
PLACE	Place generalization (Den Bosch –>Noord Brabant)

4. Research design

The main goal of this work is to provide an approach to the stakeholders that can be used in a real case scenario. To that end, we proposed KGen, a meta-heuristic approach based on a Genetic Algorithm, to build an infrastructure capable of anonymizing a dataset in a real case scenario. First, it means that the dataset specification cannot know a priori, so the approach should scale with the dataset provided. Secondly, we evaluated the algorithm proposed with experimentation, using a large dataset to validate the approach in a significant case context.

4.1. Dataset

To answer the first main research question, we build an experimentation on top of the dataset provided by the Financial Forensics (F^2) Taskforce West-Brabant-Zeeland. The task force needed a middleware capable of enabling forensic analysis without putting at risk the privacy of data owners and without any human intervention over the data; furthermore, this needed to be done in computational times which were consistent with the quantity of data available as opposed to the qualities of that data. The task force has many instances of data constrained around a reasonable set of 50+ features. Therefore, the key requirement was striking a balance between the computational complexity of the algorithms involved and the anonymization reliability of such algorithms. We were provided with an experimental dataset in the scope of our experimentation, which was completely spoofed at the source. Namely, the data was disguised as a communication from an unknown source but still reflecting the original structure and properties. The dataset in question contained 47 attributes and 1599 observations involving four different attribute types: Dates, Numbers, Strings, Places. The generalization techniques used to generalize them are showed in Table 5.

To validate KGen with a large dataset, we led a second experimentation using the "c2k_data_comma.csv" dataset [10], which is commonly considered big data (in terms of attributes, or columns of the dataset) for anonymization research, with its 97 attributes and 3942 observations. The attributes analyzed are all numeric, so the only generalization strategy applicable is the range generalization [2]. The more the range of possible values increases, the more a number is generalized (e.g., 23, at the level of generalization 1 can be generalized in 20–25).

4.2. Metrics

To find an answer to our minor RQ*s* outlined in Section 1, we defined the evaluation metrics below.

The RQ$_1$ compares the performance of the approach using execution time of the anonymization algorithm concerning the complexity of the dataset in input, as defined in related work [7]. K-Anonymity property is an NP-Hard problem [3]. For this reason, when the number of quasi-identifier attributes increases, the number of nodes in the lattice increases and, consequently, the execution time to analyze them. Hence, the execution time is a reliable indicator to compare approaches.

To answer to the RQ$_2$, we proposed a measure of accuracy, expressed as the distance between the optimal solution and pseudo-optimal solution. Each solution is part of a strategy path, and there is an optimal solution for each strategy path. Following this principle, the worst solution is the last node of this strategy path, with an accuracy value equal to 0. Instead, the optimal node has an accuracy value equal to 1. More in general, the accuracy of a solution is computed as follows:

$$acc_i = 1 - \frac{|H(S_i) - H(optS)|}{H(worstS) - H(optS)} \tag{4}$$

where H(x) is the height function of an x solution. The general accuracy, instead, is the weighted arithmetic mean of all accuracy values of our solutions, formally:

$$accuracy = \frac{\sum_{i=0}^{n}(\omega_i * acc_i)}{\sum_{i=0}^{n}\omega_i} \tag{5}$$

We choose the weighted arithmetic mean because of the 0 value problem [28]; in our case, accuracy could be 0, and it is not possible to use harmonic or geometric means with values less or equal to 0. The problem with these metrics is that we should always know the optimal solution to measure the accuracy level. So, the only way to determine the accuracy level is to compare an approach with another one that provides optimal solutions.

In the RQ$_3$, we measure the quality of a proposed solution. The quality is strongly related to the anonymization and usability of a dataset. As previously stated, the metrics used to evaluate these two aspects are the level of generalization and the percentage of a solution's suppression. With the former, we measure the level of generalization of a solution, and the latter is used as an indicator of the level of suppression of a dataset. All solutions provided by an approach are k-anonymized. Therefore, the lower is the level of generalization and the level of suppression of a solution, the better its quality. Since solutions could be more than one, the final level of generalization is the minimum of all levels of generalizations of solutions and the level of suppression is taken from the solution found.

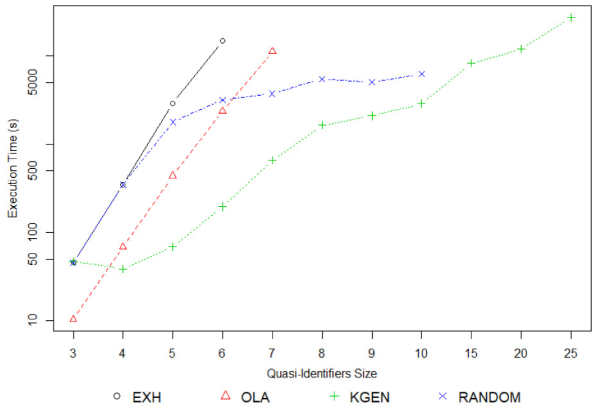
4.3. Evaluated algorithms

In the scope of our evaluation, we select four k-anonymization algorithms from state of the art, which use generalization and suppression techniques as well as an exhaustive algorithm featuring a brute-force approach by enumeration [29]. Below are listed the selected algorithms:

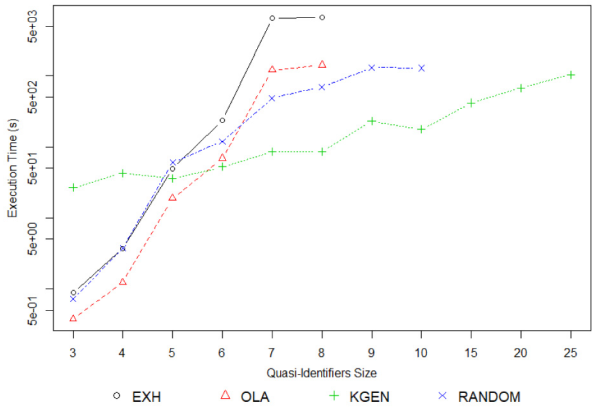
- **Exhaustive Approach.** This algorithm is an implementation of the k-anonymization property assessment algorithm as well as the generalization and suppression metrics on all nodes in the input lattice. After the analysis of the entire lattice, it is possible to find the minimum k-anonymization node. This approach provides the optimal solution;
- **OLA Approach.** As explained in the Related Work section (see Section 2.7), the OLA algorithm is an optimization of the k-anonymization algorithm. Also, this algorithm converges towards the optimal solution;
- **KGen Approach.** KGen is the approach that we want to test within this work, designed to cope with big datasets;
- **Random-Search Approach.** This algorithm is included as a validation baseline for KGen. The comparison with this algorithm is due to genetic algorithms' feature of introducing a certain degree of randomness in solution generation. Hence, by comparing KGen to a Random algorithm, we aim at establishing whether the KGen behavior is close or not to a Random approach.

Table 6
Metaheuristic parameters setup.

	KGEN	Random
maxEvaluations	5000	5000
populationSize	100	5000
crossoverRate	0.9	–
mutationRate	0.2	–
horizontalMutationRate	0.4	–

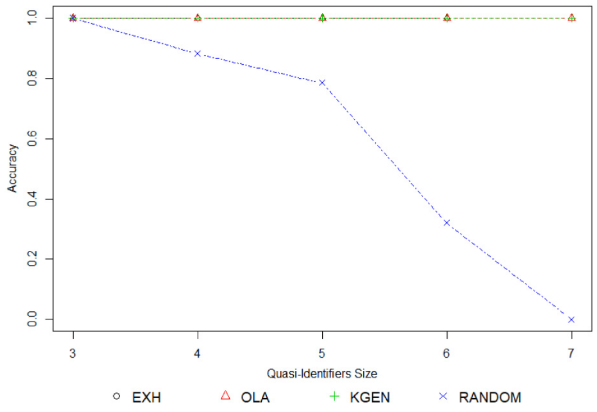


(a) Execution time c2k dataset.

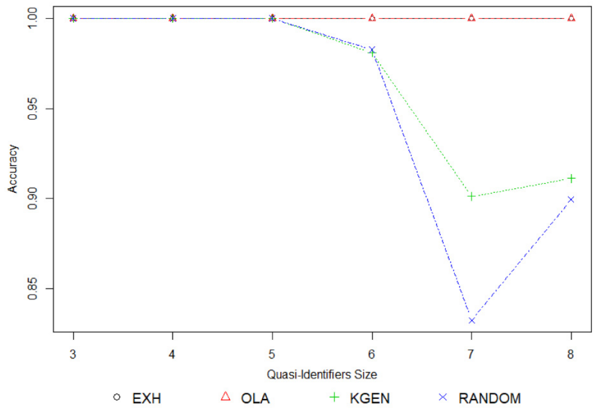


(b) Execution time F^2 dataset.

Fig. 7. Execution time evaluation results over the considered datasets.



(a) Accuracy on c2k dataset.



(b) Accuracy on F^2 dataset.

Fig. 8. Accuracy evaluation results over the considered datasets.

The remaining approaches from state of the art discussed in Section 2.7 were already compared in other previous works with the OLA approach [7]. For this reason, they have not been used in this evaluation study.

5. Results

For the comparative analysis, the experimentation was run on a CPU i7-7700HQ 2.8 GHz, 16 GB RAM DDR4, on Windows 10 64 bit. The maximum threshold allowed for the suppression technique required by the stakeholder is 0.5%. The computational time limitations were set to 15 h. Others metaheuristic parameters related to KGen and Random-Search Approach can be seen in Table 6.

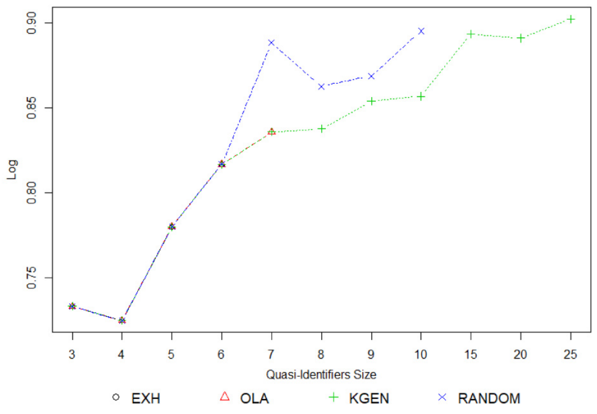
5.1. RQ1: KGen *performance*

Fig. 7 plots execution times in a logarithmic scale. The exact approach can give results for a maximum of 6 QID for the c2k dataset and 10 QID for F^2 dataset while its computation halts or crashes with the increase of QIDs. Conversely, KGen and random-search provide results until to 25 QID for c2k and 15 QID for F^2.

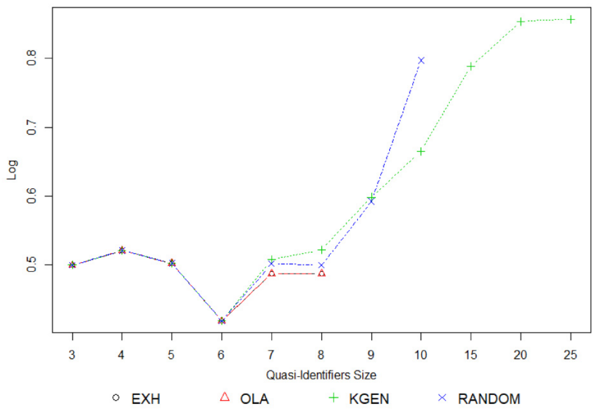
5.2. RQ2: KGen *accuracy*

Fig. 8 outlines results for accuracy. Given the limitation of the exact approaches to provide the optimal solution for several quasi-identifiers higher than 7 for the c2k dataset and 8 for the real dataset, the accuracy graph shows the accuracy level only up to 7 or 8 quasi-identifiers.

As Fig. 8(a) shows, most approaches, including KGen offer accurate results with the apparent exception of the random approach, which, by definition, is bound to be non-accurate. On Fig. 8(b), the real data dataset results show that the accuracy decreases from the seventh quasi-identifiers.

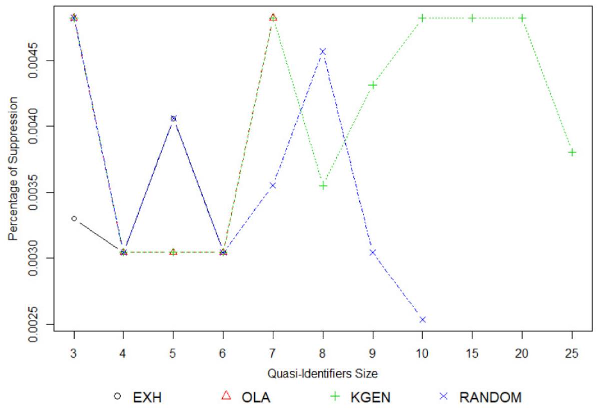


(a) Solution quality, Level of generalization (LOG) of c2k dataset.

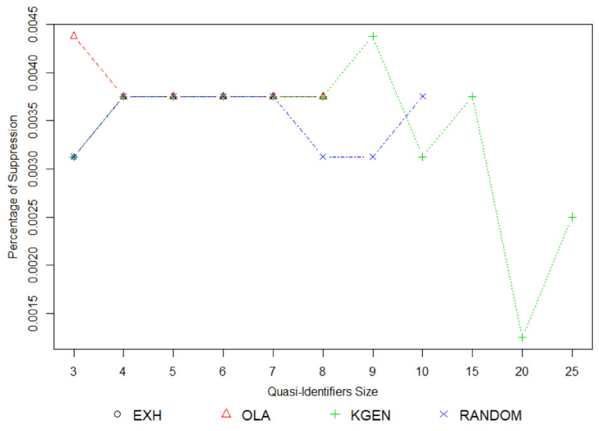


(b) Solution quality, Level of generalization (LOG) of F^2 dataset.

Fig. 9. Level Of Generalization on the dataset anonymized.



(a) Level of suppression (LOS) of c2k dataset.



(b) Level of suppression (LOS) of F^2 dataset.

Fig. 10. Level Of Suppression on the dataset anonymized.

5.3. RQ3: KGen solution quality

Figs. 9 and 10 show the level of generalization and suppression of all approaches compared. In the scope of the plot, to evaluate the extent of goodness for approaches different than exact ones (i.e., KGen and random), it is sufficient to evaluate how low are their curves.

Regarding the level of generalization, the KGen result, except with the F^2 dataset with 7/8 quasi-identifiers, is always equal or lower to the other approaches. Even when the other approaches cannot provide a solution, KGen provides better results than the Random approach.

The suppression criteria, instead, presents a different behavior depending on the dataset used. With the F^2 dataset, the behavior of KGen, for values less than 7, seems in line with the results of the exact approach. However, the suppression level for values higher than 7 is more chaotic, giving a problematic interpretation of the result achieved. The c2k dataset, instead, presents curves with unstable behavior for all the approaches considered, making it more challenging to analyze. Nonetheless, the KGen behavior is equal to the exact approaches. Considering that exact approaches provide the best results, it means that KGen provides the same good results as the exact approaches results. Considering the analysis for both levels of generalization and suppression, we can draw the following conclusions. First, the level of generalization analysis shows us how KGEN results align with the exact approach and are better than the random approach. Differently, the level of suppression analysis does not lead to any reliable conclusion since the behavior of all the approaches is a bit chaotic.

6. Discussion

As expected from our results on RQ1 the state-explosion problem [30] does not allow to have the exact solution in a reasonable time in all cases. More specifically, with more than 6 quasi-identifier attributes for the c2k dataset and 10 quasi-identifier attributes for the F^2 dataset, it is unfeasible to run exact approaches. Differently, with the usage of metaheuristics, we can provide solutions until 25 quasi-identifiers attributes and opportunistically continue if granted with the appropriate computational means. Clearly, from that point onwards, also for metaheuristic approaches is difficult to provide a solution. One factor strongly related to the increasing of the execution time on metaheuristics pertains to the maximum number of evaluations, based on metaheuristic configuration (e.g., see Table 6) since the number of nodes evaluated is directly related to these configurations.

Consequently, to decrease the execution time, operators and data processing agents can fine-tune the maxEvaluation parameter of KGen (or even the random approach) opportunistically and as needed. Another important aspect is that the slope of execution-time curves for the random approach is lower than KGen at the increase of QID. This is because a single evaluation run in KGen analyzes more than one single node, given that the crossover operator continuously generates new nodes. This limitation can be the object of future study by researchers and practitioners interested to address its impact.

Moreover, concerning RQ2, the accuracy level shows how KGen provides solutions that are identical ±.9% to the optimal approach. It means that KGen can (a) converge using its genetics operators to the optimal solution with small instances and (b) to be very close to the optimum as the number of instances increases. Conversely, the random approach initially provides a good level of accuracy due to the number of evaluations concerning the size of the problem. For example, if a lattice contains 300 nodes, with 5000 evaluations (setting of the random approach described in Table 6), the random approach analyzes all nodes in the lattice, providing a high level of accuracy. However, the opposite is true exponentially with the increase of lattice nodes.

Focusing on RQ3, we can observe the level of generalization and suppression of KGen as being very close to the level of generalization of optimal approaches. This is a good indicator of the power of our research solution. Most notably, our approach (just as the random one) can provide solutions that can deal with higher numbers of quasi-identifier attributes, a feature where most optimal approaches fail. Unlike the random approach, however, KGen provides excellent results in terms of generalization level, considering the suppression applied. If the random approach seems to have better results on large instances, considering only the generalization level, we can see how this is due to the high level of suppression applied by the random approach itself. Looking at both metrics, we can easily understand how KGen has the best results.

From the results of the three sub-research questions, we can assume that KGen performs well in real case contexts. Moreover, given the dataset provided by the Taskforce West-Brabant-Zeeland, we can anonymize their dataset with a good level of anonymization, having the same results of exact approaches.

Unlike heuristic approaches, meta-heuristic approaches can also perform well in a context where the dataset size, in terms of the number of quasi-identifiers, is more extensive. Hence, a stakeholder can use KGen in large contexts scenarios. Nonetheless, to ensure the applicability in a general context, the approach needs to be validated with more datasets.

Lastly, after providing the anonymized dataset, KGen provides also metadata regarding the information loss for each dataset attribute. Hence, the final user can estimate the damage entity by mean of information loss of each attribute.

7. Limitations and threats to validity

This section outlines the major limitation we perceive in our work, which reflects one of the optimizations that KGen features in its processing and algorithms. As outlined in Section 3.2, KGen features a lattice size reduction technique that limits the approach applicability in specific cases. Nevertheless, the technique is essential since it can work on a smaller search space than the original one, whose size could be untractable without major software-defined infrastructure requirements. However, the described technique introduces a vulnerability when, during the anonymization process, also the suppression technique is introduced. Preprocessing without suppression ensures that all lattice nodes except for the new minimum node found in the process are not k-anonymized. With the suppression active, instead, this does not hold. Let us take into account the example in Table 7. If we apply the suppression criteria (with a maximum level of suppression set, by default, of 35%) on each dataset, in order that all datasets are k-anonymized, we have the suppression of the last row on the first and second dataset and the suppression of the second row in the last dataset (Table 7b–7c–7d). At this point, removing the second and third-row from the dataset, the remaining dataset is composed by only the first row, with a final level of generalization of <2, 1, 0>. Nevertheless, this dataset is k-anonymized also without any generalization (<0, 0, 0>). By applying the suppression criteria, it is possible to have one k-anonymized node with a level of generalization less than the minimum level of generalization provided by the preprocessing. We are aware of this limitation and plan to address it in future developments and iterations over this work.

Table 7
Lattice reduction process with suppression criteria.

Age	Postcode	Gender
24	80015	F
28	80019	M
42	85073	F

(a) Example of a dataset not k-anonymized.

Age	Postcode	Gender
20–29	8001*	F
20–29	8001*	~~M~~
~~40–49~~	~~8507*~~	F
(b) Age: LOG 2.	(c) PC: LOG 1.	(d) Gender: LOG 0

(e) Datasets k-anonymized, applying the suppression criteria (with a max level of suppression of 35% of the entire dataset). The final dataset contains only the first row.

8. Conclusion and future work

With the quickly increasing amount of digital data, there emerges a growing need to provide support for fast and scalable data-processing capable of offering anonymization guarantees. In this paper, we introduce KGen, a scalable approach to data-intensive k-anonymization featuring genetic algorithms.

The KGen approach focuses on the assessment of the balance between two critical, and opposite data quality attributes functional to data-processing, namely, data privacy versus usefulness of data. As aforementioned, KGen exploits genetic algorithms that allow organically increasing the level of privacy of the data while safeguarding that the data evidence, which is still usable e.g., in terms of financial evidence and audit trails part of governmental data-intensive processing.

KGen is a practical, scalable, data-intensive approach that can effectively anonymize datasets embracing the well-accepted k-anonymization measure. The approach is supported by a prototype coded in Java and tested through various experiments using benchmarks and real-life industrial datasets.

Initial results look very promising. We have shown empirically that the behavior of KGen level of generalization metric performance equally well as other optimization approaches, while KGen – in contrast to other approaches – can deal with a large number of quasi-identifiers, and thus Big datasets.

Future work will focus on building a more robust and user-friendly interface on top of the current prototype and more personalized privacy measures. Besides, we intend to work on a dynamic version of KGen, D-KGen, that can deal with streaming data that dynamically add/remove/alter the dataset on-the-fly and just-in-time, breaking the "closed-world assumption" underpinning most of the existing approaches.

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

[1] E. Arfelt, D.A. Basin, S. Debois, Monitoring the GDPR, in: K. Sako, S. Schneider, P.Y.A. Ryan (Eds.), ESORICS (1), in: Lecture Notes in Computer Science, vol. 11735, Springer, 2019, pp. 681–699.

[2] P. Samarati, Protecting respondents identities in microdata release, IEEE Trans. Knowl. Data Eng. 13 (6) (2001) 1010–1027.

[3] A. Meyerson, R. Williams, On the complexity of optimal k-anonymity, in: Proceedings of the Twenty-Third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, ACM, 2004, pp. 223–228.

[4] C.C. Aggarwal, On k-anonymity and the curse of dimensionality, in: VLDB, Vol. 5, 2005, pp. 901–909.

[5] L. Sweeney, Guaranteeing anonymity when sharing medical data, the Datafly system, in: Proceedings of the AMIA Annual Fall Symposium, American Medical Informatics Association, 1997, p. 51.

[6] L. Sweeney, Achieving k-anonymity privacy protection using generalization and suppression, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 10 (05) (2002) 571–588.

[7] K. El Emam, F.K. Dankar, R. Issa, E. Jonker, D. Amyot, E. Cogo, J.-P. Corriveau, M. Walker, S. Chowdhury, R. Vaillancourt, et al., A globally optimal k-anonymity method for the de-identification of health data, J. Am. Med. Inform. Assoc. 16 (5) (2009) 670–682.

[8] K. LeFevre, D.J. DeWitt, R. Ramakrishnan, Incognito: Efficient full-domain k-anonymity, in: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, ACM, 2005, pp. 49–60.

[9] D.E. Goldberg, J.H. Holland, Genetic algorithms and machine learning, Mach. Learn. 3 (2) (1988) 95–99.

[10] C. Crawford, Cargo 2000 dataset kaggle, 2017, URL https://www.kaggle.com/crawford/cargo-2000-dataset.

[11] M. Guerriero, D.A. Tamburri, Y. Ridene, F. Marconi, M.M. Bersani, M. Artac, Towards DevOps for privacy-by-design in data-intensive applications: A research roadmap, in: W. Binder, V. Cortellessa, A. Koziolek, E. Smirni, M. Poess (Eds.), ICPE Companion, ACM, 2017, pp. 139–144.

[12] M. Guerriero, D.A. Tamburri, E.D. Nitto, Defining, enforcing and checking privacy policies in data-intensive applications, in: J. Andersson, D. Weyns (Eds.), SEAMS@ICSE, ACM, 2018, pp. 172–182.

[13] T. Dalenius, Finding a needle in a haystack or identifying anonymous census records, J. Off. Stat. 2 (3) (1986) 329.

[14] K. El Emam, A. Brown, P. AbdelMalik, Evaluating predictors of geographic area population size cut-offs to manage re-identification risk, J. Am. Med. Inform. Assoc. 16 (2) (2009) 256–266.

[15] K. El Emam, S. Jabbouri, S. Sams, Y. Drouet, M. Power, Evaluating common de-identification heuristics for personal health information, J. Med. Int. Res. 8 (4) (2006) e28.

[16] K. El Emam, E. Jonker, S. Sams, E. Neri, A. Neisa, T. Gao, S. Chowdhury, Pan-Canadian De-Identification Guidelines for Personal Health Information, Privacy Commissioner of Canada, 2007.

[17] C.I. of Health Research, CIHR Best Practices for Protecting Privacy in Health Research, Canadian Institutes of Health Research, 2005.

[18] B.C. Fung, K. Wang, R. Chen, P.S. Yu, Privacy-preserving data publishing: A survey of recent developments, ACM Comput. Surv. (Csur) 42 (4) (2010) 1–53.

[19] L. Sweeney, Computational disclosure control: a primer on data privacy protection (Ph.D. thesis), Massachusetts Institute of Technology, 2001.

[20] X. Sun, H. Wang, J. Li, On the complexity of restricted k-anonymity problem, in: Asia-Pacific Web Conference, Springer, 2008, pp. 287–296.

[21] D. Simon, Evolutionary Optimization Algorithms, Wiley, 2013.

[22] R.J. Bayardo, R. Agrawal, Data privacy through optimal k-anonymization, in: 21st International Conference on Data Engineering (ICDE'05), IEEE, 2005, pp. 217–228.

[23] V.S. Iyengar, Transforming data to satisfy privacy constraints, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 279–288.

[24] J. Zhang, X. Gong, Z. Han, S. Feng, An improved algorithm for k-anonymity, in: International Conference on E-Business Technology and Strategy, Springer, 2012, pp. 352–360.

[25] T. Blickle, L. Thiele, A mathematical analysis of tournament selection, in: ICGA, Vol. 95, Citeseer, 1995, pp. 9–15.

[26] S. Mirjalili, Genetic algorithm, in: Evolutionary Algorithms and Neural Networks, Springer, 2019, pp. 43–55.

[27] M.W. Mkaouer, M. Kessentini, Model transformation using multiobjective optimization, in: Advances in Computers, Vol. 92, Elsevier, 2014, pp. 161–202.

[28] R. Wood, M. Lenzen, Zero-value problems of the logarithmic mean divisia index decomposition method, Energy Policy 34 (12) (2006) 1326–1331.

[29] J. Ullmann, An algorithm for subgraph detection, J. ACM 23 (1) (1976) 31–42.

[30] E. Clarke, Model checking – my 27-year quest to overcome the state explosion problem, 2008, p. 182, http://dx.doi.org/10.1007/978-3-540-89439-1_13.