Lightning: Utility-Driven Anonymization of High-Dimensional Data

Fabian Prasser, Raffael Bild, Johanna Eicher, Helmut Spengler, Florian Kohlmayer, Klaus A. Kuhn

Chair of Biomedical Informatics, Department of Medicine, Technical University of Munich (TUM), Germany.

E-mail: `firstname.lastname@tum.de`

Received 15 July 2015; received in revised form 18 April 2016; accepted 18 April 2016

Abstract. The ARX Data Anonymization Tool is a software for privacy-preserving microdata publishing. It implements methods of statistical disclosure control and supports a wide variety of privacy models, which are used to specify disclosure risk thresholds. Data is mainly transformed with a combination of two methods: *(1)* global recoding with full-domain generalization of attribute values followed by *(2)* local recoding with record suppression. Within this transformation model, given a dataset with low dimensionality, it is feasible to compute an optimal solution with minimal loss of data quality. However, combinatorial complexity renders this approach impracticable for high-dimensional data. In this article, we describe the *Lightning* algorithm, a simple, yet effective, utility-driven heuristic search strategy which we have implemented in ARX for anonymizing high-dimensional datasets. Our work improves upon existing methods because it is not tailored towards specific models for measuring disclosure risks and data utility. We have performed an extensive experimental evaluation in which we have compared our approach to state-of-the-art heuristic algorithms and a globally-optimal search algorithm. In this process, we have used several real-world datasets, different models for measuring data utility and a wide variety of privacy models. The results show that our method outperforms previous approaches in terms output quality, even when using k-anonymity, which is the model for which previous work has been designed.

Keywords. data privacy, anonymization, de-identification, statistical disclosure control, high-dimensional data, k-anonymity, ℓ-diversity, t-closeness, δ-presence, super-population models

1 Introduction

The ARX Data Anonymization Tool is an open source software for privacy-preserving microdata publishing [34]. A typical use case is the de-identification of individual-level research data prior to sharing it with others. The tool focuses on *a priori disclosure risk control*, where data is sanitized in such a way that predefined thresholds on privacy risks are met while loss of information is minimized. Risk thresholds are specified in terms of *privacy models*, many of which have been developed by the computer science community. ARX also supports *a posteriori disclosure risk control*, which is prevalent in the statistics community and where privacy risks are balanced with data quality. For this purpose, the tool enables users to browse a space of possible data transformations while offering methods for automatically and semi-automatically analyzing data utility and privacy risks. Also, privacy and utility can be balanced by using different methods and parameters for measuring both aspects. To achieve this flexibility, the tool utilizes an intuitive transformation model which is implemented in a highly scalable manner. ARX exposes all functionality via a comprehensive graphical user interface which provides wizards and visualizations that guide users through the different phases of a data anonymization process [32].

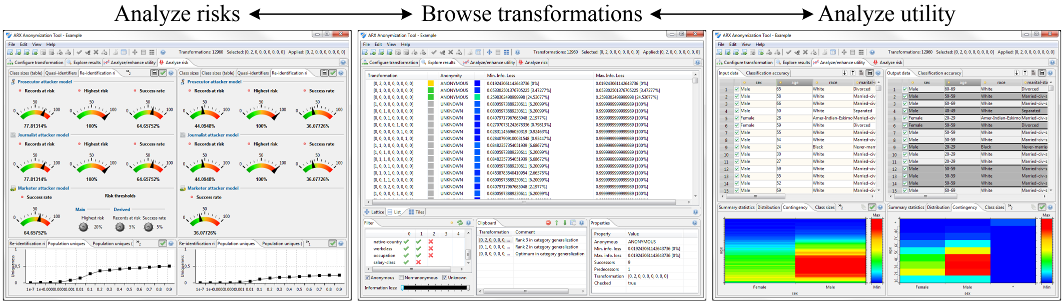


Figure 1: Screenshots of ARX's perspectives for (1) analyzing re-identification risks, (2) browsing the solution space and (3) analyzing data utility

As a basis for data anonymization, the tool constructs a solution space of possible data transformations, which is then characterized using models for disclosure risks and models for data quality (more details will be given in Section 2). Figure 1 shows three screenshots of perspectives provided by the tool. In the first perspective, re-identification risks can be analyzed and compared between input and output data. In the second perspective, data transformations are visualized as a list which can be filtered and sorted according to privacy properties and data utility. The aim of the third perspective is to enable users to manually analyze the utility of output datasets resulting from the application of a data transformation. For this purpose, results of univariate and bivariate methods of descriptive statistics are presented. While the methods implemented by ARX have been selected with applications to biomedical data in mind, the tool is domain-agnostic and suited for anonymizing a wide variety of datasets.

2 Background

A balancing of privacy and utility can be performed with ARX by choosing different privacy models, risk models, transformation methods and utility measures as well as by varying provided parameters. In this section, based on a comprehensive overview presented in [32], we will address important methods implemented by our tool with a specific focus on design aspects which motivated this work.

2.1 Privacy Models

In ARX, thresholds on privacy risks are represented by means of privacy models, which require assumptions to be made about the (likely) background knowledge and goals of potential attackers. The general attack vector assumed is *linkage* of a sensitive dataset with an identified dataset (or similar background knowledge about individuals). The attributes which could potentially be used for linkage are termed *quasi-identifiers* (or indirect identifiers, or keys). Such attributes are not directly identifying but they may in combination be used for linkage. Moreover, it is assumed that they cannot simply be removed from the dataset as they may be required for analyses and that they are likely to be available to an attacker. Three types of privacy threats are commonly considered [26]:

- **Membership disclosure** means that linkage allows to determine whether or not data about an individual is contained in a dataset [29]. While this does not directly disclose any information from the dataset itself, it may allow an attacker to infer meta-information. If, for example, the data is from a cancer registry, it can be inferred that an individual has or has had cancer.

- **Attribute disclosure** means that an attacker can infer *sensitive attribute values* about an individual, i.e. information with which individuals are not willing to be linked with, without necessarily relating the individual to a specific record in a dataset [27]. As an example, linkage to a set of records can allow inferring information if all records share a certain sensitive attribute value.

- **Identity disclosure** (or *re-identification*) means that an individual is linked to a specific data record [38]. This type of attack is addressed by many laws and regulations worldwide and it is therefore often related to severe consequences for data owners. From the definition of this type of disclosure it follows that an attacker can learn all sensitive information contained about the individual.

Figure 2 shows an example dataset in which the attributes *age* and *gender* are considered quasi-identifying, *state* is considered insensitive and *diagnosis* is considered sensitive. It further shows a privacy preserving transformation of the dataset, which will be explained in more detail in the following paragraph.

Quasi-identifying		Insensitive	Sensitive
Age	Gender	State	Diagnosis
34	Male	NY	Pneumonia
45	Female	MS	Gastritis
66	Male	NY	Gastritis
70	Male	TX	Pneumonia
35	Female	AL	Pneumonia
21	Male	AL	Gastritis
18	Female	TX	Pneumonia
19	Female	MS	Gastritis

Quasi-identifying		Insensitive	Sensitive
Age	Gender	State	Diagnosis
20-60	Male	NY	Pneumonia
20-60	Female	MS	Gastritis
≥61	Male	NY	Gastritis
≥61	Male	TX	Pneumonia
20-60	Female	AL	Pneumonia
20-60	Male	AL	Gastritis
⋆	⋆	⋆	⋆
⋆	⋆	⋆	⋆

Figure 2: Example dataset and a privacy-preserving transformation

To counter membership, attribute and identity disclosure, ARX supports arbitrary combinations of several privacy models:

- **k-Anonymity:** This model aims at protecting datasets from identity disclosure [38]. A dataset is *k-anonymous* if, regarding the attributes modeled as quasi-identifiers, each data item cannot be distinguished from at least $k-1$ other data items. This property is also used to define *equivalence classes* of indistinguishable entries [35]. The output dataset from Figure 2 fulfills 2-anonymity.

- **ℓ-Diversity, t-closeness and δ-disclosure privacy:** These models aim at protecting datasets against attribute disclosure [27, 25, 6]. Different variants exist of ℓ-diversity and t-closeness, which offer different degrees of protection. The weakest model, *distinct-ℓ-diversity*, requires that ℓ different sensitive attribute values are contained in each equivalence class [25]. The output dataset from Figure 2 fulfills *distinct*-2-*diversity*. ARX also implements two stricter variants of this model: *recursive-(c, ℓ)-diversity* and *entropy-ℓ-diversity* [27]. *t-closeness* requires that the distance between the distribution of sensitive values in each equivalence class and their overall distribution in the dataset must be lower than a given threshold [25]. ARX implements two variants of this model, one of which uses generalization hierarchies to calculate the required distances. *δ-disclosure privacy* also enforces a restriction on the distances between the distributions of sensitive values but it uses a multiplicative definition [6].

- **δ-Presence:** This model aims at protecting datasets against membership disclosure [29]. It requires that the disclosed dataset is explicitly modeled as a subset of a larger dataset which represents the attacker's background knowledge. The model enforces restrictions on the probabilities with which it can be determined whether or not an individual from the global dataset is contained in the research subset. Bounds for these probabilities are calculated based on the sizes of equivalence classes [29].

- **Sample- or population-based models**: The previously described privacy models enforce syntactic conditions on each individual equivalence class. ARX supports several more complex models, which calculate disclosure risks based on the entire dataset (*sample-based models*) or based on the relationship between the dataset and the underlying population from which it was sampled (*population-based models*). A typical example for the former type of models is *strict-average risk*, which enforces a threshold on the average size of equivalence classes [14]. A typical example of the latter type of methods are *super-population models*, which can be used to estimate *population uniqueness*, i.e., the fraction of records in the dataset which are unique within the overall population. Here, characteristics of the population are approximated with probability distributions, where parameters are estimated from the empirical distribution of the sizes of equivalence classes in a dataset. ARX currently implements the models by Hoshino [18] and by Chen and McNulty [7]. Moreover, the tool implements the model by Dankar et al. [9], which combines the above estimators with the model by Zayatz [42] and which has been validated with real-world clinical datasets [9]. We note that, to our knowledge, ARX is the only data anonymization tool which supports using such statistical models for a priori disclosure risk control.

2.2 Transformation Model, Solution Space and Utility Measures

Data transformation in ARX is primarily performed with user-defined generalization hierarchies, which can be created for categorical and continuous attributes.

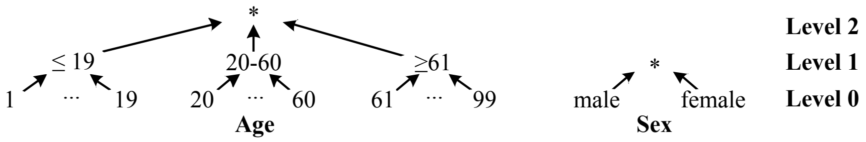


Figure 3: Examples of generalization hierarchies

Examples are shown in Figure 3. In these two simple hierarchies, values of the attribute *age* are first categorized by transforming them into age groups and then suppressed, while values of the attribute *sex* can only be suppressed. In the example from Figure 2, the attribute *age* has been generalized to the first level of the associated hierarchy. To support on-the-fly categorization of continuous variables, hierarchies in ARX can be represented in a functional manner, e.g. by means of intervals [32].

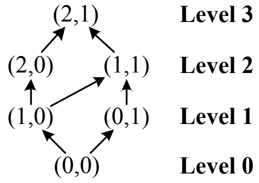


Figure 4: Generalization lattice for the example dataset and hierarchies

The backbone of ARX's data transformation functionality is global recoding with full-domain generalization [20]. Global recoding means that the same transformation rule is applied to identical values in different records of a dataset. Full-domain generalization means that all values of an attribute are generalized to the same level of the associated hierarchy. With this transformation model, it is possible to model the solution space as a *generalization lattice*, which is a partially ordered set of all possible combinations of generalization levels of each attribute. An example utilizing the hierarchies for *age* and *gender* from Figure 3 is shown in Figure 4. Each node represents a single transformation which defines generalization levels for all quasi-identifiers. An arrow indicates that a transformation is a direct generalization of a more specialized transformation which can be derived by incrementing one of the generalization levels defined by its predecessor. The original dataset is at the bottom $(0, 0)$, whereas the transformation with maximal generalization $(2, 1)$ is at the top. The output dataset from Figure 2 is the result of applying the transformation $(1, 0)$ to the input dataset.

Global recoding with full-domain generalization is often not flexible enough to produce datasets of high quality [15]. For this reason, the tool combines this basic transformation model with additional methods of local recoding. First, with *microaggregation* sets of values of an attribute can be made indistinguishable by replacing them with aggregates [15]. Second, *record suppression* can be used to automatically remove outliers from a dataset. As a result of record suppression, less generalization is required to ensure that the remaining records fulfill a given privacy model [22]. In the output from Figure 2 the last two records have been suppressed. Third, as an alternative to simply removing outliers, ARX is also able to *apply different generalization schemes* to different parts of a dataset. For the sake of clarity, we will in the remainder of this article focus on global recoding via full-domain generalization combined with local recoding via record suppression.

For assessing the quality of output data, ARX implements several general-purpose utility measures. They are meant to support users with finding a transformation which provides an adequate balance between privacy and data quality. Moreover, they can be used to perform automated a priori disclosure risk control, where the tool tries to find a solution to a given privacy problem which maximizes data utility. ARX distinguishes between two different types of measures. The first type of methods is based on the sizes of equivalence classes. Important examples are *Average Equivalence Class Size* (AECS) [23], *Discernibility* [4], *Ambiguity* [30] and *KL-Divergence* [27]. The second type of methods is characterized by calculating independent values for each attribute which are then compiled into a global value. In this case, the tool allows to assign weights to attributes which model their importance. Examples of methods from this class are *Height* [24], *Loss* [19], *Precision* [39] and *Non-Uniform Entropy* [13].

3 Objectives and Outline

Automatically finding data transformations which adequately balance privacy and data quality is a complex problem. Even for the simple k-anonymity privacy model, the problem of finding a solution which maximizes data quality is NP-hard for any $k \geq 3$ [28]. When data is transformed with full-domain generalization, which is a very restricted way of using generalization that results in relatively small search spaces, optimal solutions for small problem instances can be computed by using globally-optimal search algorithms [20]. Compared to other approaches, the algorithm implemented by ARX achieves excellent performance by utilizing sophisticated data compression schemes as well as various pruning strategies [21, 20, 33]. However, it is easy to see that even these types of solution spaces can become too large to be searched exhaustively: the size of a generalization lattice equals the product of the heights of the generalization hierarchies. This means that the size of the search space grows exponentially with the number of quasi-identifiers n for which such a hierarchy has been defined ($2^{\mathcal{O}(n)}$). As a rule of thumb, an optimal solution can be computed for datasets with up to about 15 quasi-identifiers, but the exact limit depends on the size of the generalization hierarchies utilized.

To cope with the challenge of anonymizing high-dimensional datasets, various heuristic search strategies have been proposed. However, these strategies focus on different transformation models or they have been developed with a specific privacy model in mind. Previous approaches are thus not well suited for ARX, which supports a broad range of privacy models and measures for data utility. As a consequence, we have developed *Lightning*, a novel heuristic for anonymizing high-dimensional datasets which uses the given mathematical model for data utility to guide the search process.

We have used the broad spectrum of methods supported by our tool to perform an extensive evaluation in which we have compared our approach with state-of-the-art heuristic search algorithms and with a globally-optimal algorithm. In this process, we have used several real-world datasets, different measures for data utility as well as multiple privacy models. The results of the experiments show that our approach outperforms previous solutions in terms of output quality, even when using k-anonymity, which is the model for which previous work has been designed. Moreover, our heuristic method constitutes a valuable complement to globally-optimal algorithms.

The remainder of this article is structured as follows. In Section 4 we will present an overview of previous approaches. In Section 5 we will describe our novel approach. In Section 6 we will describe our experimental setup and in Section 7 we will present the results. Finally, in Section 8, we will discuss our approach and conclude this paper.

4 Related Work

To our knowledge, two heuristic anonymization algorithms which use global recoding with full-domain generalization have been proposed in the literature: *DataFly* [37] and the *Improved Greedy Heuristic* (IGreedy) [3]. As the solution spaces considered are too large to be searched exhaustively, the algorithms need a termination condition. In this context, both methods utilize the concept of *minimal anonymization* [15]. This means that they perform a bottom-up search which terminates as soon as a transformation has been found which satisfies the given privacy model. Both approaches focus on the k-anonymity model.

The DataFly algorithm starts with the transformation which preserves the input dataset, i.e. the bottom node in the generalization lattice, and iteratively increases the generalization level of the attribute with the highest number of distinct values. IGreedy implements an extension of this strategy. It also starts with the bottom node and in each iteration it selects one attribute for generalization. This decision is made in such a way that the resulting dataset has the smallest possible minimal equivalence class size. If the same minimal class size may be achieved by generalizing different attributes it falls back on DataFly's strategy. Both algorithms terminate when they have found a transformation which results in a dataset that fulfills all disclosure risk thresholds. Both strategies are not well suited for ARX for three main reasons.

The first problem is related to the way in which record suppression is typically implemented by data anonymization algorithms and also by ARX. Here, users are allowed to specify a limit on the maximal number of records which may be removed from a dataset. This is called the *suppression limit* [13]. Also, the set of records which need to be removed from a dataset after a specific generalization scheme has been applied is identified by looking at the equivalence classes: when class-based privacy models are used all records are removed which are part of an equivalence class that does not conform to the specified disclosure risk thresholds. Both DataFly and IGreedy assume that on any path from the bottom node to the top node of the lattice there is a point at which disclosure risks fall below the given threshold and that all further generalizations of this transformation are also valid solutions to the given anonymization problem. This principle is called *monotonicity*. While it is true that most privacy models are monotonic when data is transformed only with global recoding via generalization, this is not true when the transformation model is combined with record suppression as described above [22]. Consider a transformation which is a valid solution to a given anonymization problem and in which at least one equivalence class has been suppressed because it does not fulfill the specified disclosure risk requirements. When the dataset is further generalized, the suppressed equivalence class may be merged with another equivalence class which did previously conform to the disclosure risk requirements. However, the resulting class, now containing records from both classes, may not conform to the privacy requirements. If this class contains too many records to be suppressed without violating the suppression limit, the resulting dataset is not a valid solution. Formal proofs can be found in Appendix A.

Secondly, with a transformation model which involves generalization followed by record suppression, minimality does not imply any guarantees in terms of data quality. The reason is that, in this transformation model, utility measures from all common models do not decrease monotonically with increasing generalization. This is easy to see, as increasing the amount of generalization may reduce the required amount of suppression, thereby increasing overall data utility [22]. Moreover, in the general case there may be many different minimal solutions to an anonymization problem, which are likely to have different properties in terms of data quality.

Thirdly, ARX is able to automatically balance the application of generalization and record suppression to achieve optimal data utility [32]. It is thus reasonable to use a suppression limit of 100%. In this case, all transformations in the search space are valid solutions to the anonymization problem. DataFly and IGreedy will therefore return the transformation with minimal generalization, which is always the bottom node of the generalization lattice. This means that only record suppression will be used to anonymize a dataset, which is likely to result in output data with sub-optimal quality.

5 Utility-Driven Anonymization of High-Dimensional Data

In this section, we will first describe the basic ideas behind our approach. We will then introduce our notion and describe the building blocks of the *Lightning* algorithm. Finally, we will present a detailed description of our method.

5.1 Basic Idea and Notion

Data anonymization is a non-trivial optimization problem with two conflicting objectives: minimizing disclosure risks while maximizing data utility. With a priori disclosure risk control this contradiction is resolved by letting a human decision maker define a subjective preference on one of the optimization goals, i.e. specific disclosure risk thresholds. What remains is a simpler optimization problem in which the objective is to make sure that risk thresholds are met while data utility is maximized. A priori anonymization methods can still be used as a building block for balancing risks and utility, for example by repeatedly executing them with different risk thresholds to construct a risk-utility frontier [8].

Globally-optimal anonymization algorithms are clearly designed to optimize data utility [20]. However, related heuristic search strategies, i.e. DataFly and IGreedy, use objective functions (maximizing distinct values per attribute and minimizing equivalence class sizes) which are not directly related to either of both objectives but which are placed somewhere in between measures for disclosure risks and data utility. Also, only the risk of identity disclosure is considered while we aim at supporting a wide variety of privacy models.

The basic idea of the *Lightning* algorithm is to use the remaining objective function, i.e. the maximization of data utility, to guide a heuristic search. Starting from the bottom node of a given generalization lattice, our algorithm performs a best-first search by always selecting the successor which results in an output dataset with highest utility. The search terminates after a user-specified amount of time instead of when the first solution is discovered. This is motivated by the fact that neither disclosure risk nor data utility is guaranteed to be monotonic in our setup, as we have explained in the previous section.

We will use an object-oriented formalism to describe our approach. An object of type *Lattice* represents the search space and *lattice.bottom* returns its bottom node while *lattice.height* returns the level of its top node. An object of type *Transformation* represents a transformation and *transformation.utility* returns its utility as a decimal number, *transformation.successors* returns a list containing all of its direct successors and *transformation.expanded* returns whether or not the transformation has already been processed by the algorithm. We will explain this operation in the next paragraph.

Algorithm 1: Function EXPAND

```
Input: Transformation transformation, Transformation optimum, Priority queue queue
begin
    $transformation.expanded \leftarrow true$
    for ($successor \in transformation.successors$) do
        if ($\neg successor.expanded$) then
            CHECK($successor, optimum$)
            $queue.add(successor)$
```

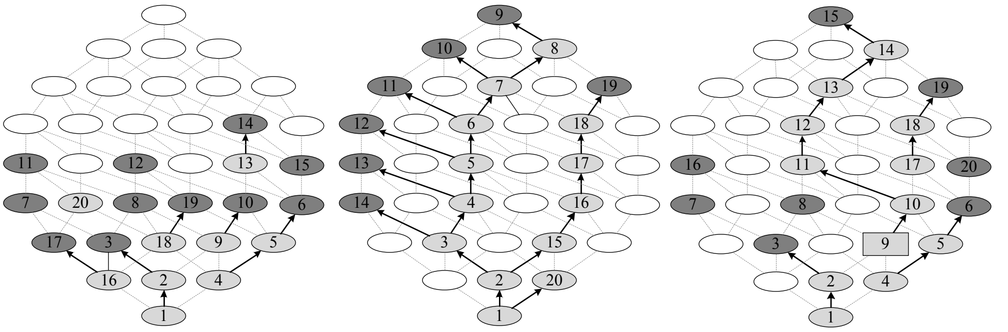
Our algorithm uses two basic building blocks.

- CHECK*(transformation, optimum)* takes two transformations as arguments. It applies the first transformation to the dataset and determines whether the resulting dataset meets the defined risk thresholds. Moreover, it computes the resulting utility (*transformation.utility*). If the thresholds are met and the utility is higher than that of the current optimum, *optimum* is updated accordingly. We will not explain the internals of this function in more detail, but instead assume that it is provided by the runtime environment, which in our case is ARX.

- EXPAND*(transformation, optimum, queue)* takes three arguments. It marks the *transformation* as *expanded,* iterates through *transformation.successors* and calls CHECK*(successor, optimum)* for all successors which have not already been expanded previously. Moreover, it adds the successors to the given priority queue. In the *queue*, transformations are ordered by utility from highest to lowest. Pseudocode for this function is provided in Algorithm 1. It is assumed that the given transformation has already been checked.

5.2 The Lightning Algorithm

In ARX our new algorithm complements our globally-optimal search strategy *Flash* [20]. We therefore call it *Lightning*. The algorithm starts by calling CHECK*(lattice.bottom, optimum)* and then adding *lattice.bottom* to the global priority queue. During the remaining execution time, the algorithm tries to find a solution which maximizes data utility by performing a heuristic search. An important aspect of our algorithm is how it handles local optima. We will explain this in a step-wise manner.



Best-first search Greedy search Hybrid search

Figure 5: Comparison of the first iterations of the three variants of our search strategy

The first variant of our algorithm performs a *best-first search*. In every step, it expands the top element from the global queue. An example is shown in Figure 5. Here, a transformation is marked with n, if it was expanded in the n-th iteration. An arrow between transformations indicates that they have been expanded in order. Transformations colored in dark gray denote the end of such a path, meaning that the search location has changed at this point. As can be seen in the figure, the best-first strategy will often resemble a breadth-first search, which in our context means that the lattice is traversed level by level [34]. The reason is that, although not guaranteed, data utility is likely to decrease with generalization. For high-dimensional datasets, which result in generalization lattices that are very large and very wide, the consequence is that the best-first algorithm will mainly expand transformations near the bottom. However, there are anonymization problems which require one or more attributes to be generalized to higher levels in order to meet disclosure risk thresholds.

The second variant of our algorithm will for the remaining execution time perform a *greedy search*. When a transformation is reached which does not have any unprocessed successors, the algorithm performs backtracking. As is visualized in Figure 5, this variant will also check transformations which are close to the top of the lattice. However, as lattices for high-dimensional datasets become equally wide near the bottom and the top, the algorithm will spend most of its execution time expanding transformations which define high levels of generalization and thereby miss the opportunity to find solutions with better data utility.

Algorithm 2: Main loop of the LIGHTNING algorithm

```
Input: Lattice lattice, Suppression limit suppression
begin
    queue ← new priority-queue
    optimum ← null
    CHECK(lattice.bottom, optimum)
    queue.add(lattice.bottom)
    step ← 0
    while (next ← queue.poll() ≠ null) do
        if (suppression ≠ 0 ∧ (optimum ≠ null ∨ node.utility ≥ optimum.utility)) then
            step ← step + 1
            if (step mod lattice.height = 0) then
                GREEDY(next, optimum, queue)
            else
                EXPAND(next, optimum, queue)
```

The final variant of our method, which is explained in Algorithm 2, implements a *hybrid strategy*. It starts by performing a best-first search for n steps. Each step is one *expand* operation. After n operations, it switches to a greedy search without backtracking. In this phase, as is explained in Algorithm 3, it iteratively expands the transformation with highest data utility until it reaches a transformation which does not have any unprocessed successors. All transformations which have been checked during this process are added to the global priority queue. The algorithm then returns to the best-first strategy for another n steps. Lightning also implements a very simple pruning strategy. If the suppression limit is 0% data utility will decrease monotonically with generalization for all common utility measures [4, 23, 13, 19, 24, 39]. In this case, it will therefore exclude transformations from the search process which result in datasets with a utility which is already lower than the current optimum.

Algorithm 3: Function GREEDY

```
Input: Transformation transformation, Transformation optimum, Priority queue queue
begin
    local-queue ← new priority-queue
    EXPAND(transformation, optimum, local-queue)
    if (next ← local-queue.poll() ≠ null) then
        GREEDY(next, optimum, queue)
    while (next ← local-queue.poll() ≠ null) do
        queue.add(next)
```

As a generic solution we chose to set the parameter n to *lattice.height*. This leads to a balanced use of both search strategies, because each greedy phase will result in not more than *lattice.height* expand operations. In Figure 5, a rectangular node indicates a transformation at which a greedy search has been started. The example shows that Lightning considers transformations with a small amount of generalization, transformations with a large amount of generalization as well as transformations in-between these two extremes.

6 Experimental Setup

In this section, we will describe the setup of our experiments. We emphasize that all datasets and generalization hierarchies used in our evaluation are publicly available [34]. Moreover, our implementation of all algorithms evaluated in this work is available online as open-source software [5].

6.1 Privacy Models, Utility Measures and Parameters

We performed three kinds of experiments to evaluate different aspects of our solution:

- Firstly, we compared our algorithm with related work in terms of data utility. We used suppression limits of 0% and 10% as well as a variant of our algorithm which terminates when a minimally anonymous transformation has been found. This follows the experiments in previous work [37, 3] and thus provides comparability.
- Secondly, we compared our heuristic search strategy with a globally-optimal algorithm. We investigated the quality of the solution found by the heuristic strategy when terminating it after the amount of time required by the optimal algorithm to classify the complete solution space. The results provide insights into potential benefits which may be offered by heuristic strategies even when anonymizing low-dimensional data. We used suppression limits of 0% and 100%, basically testing two different transformation models: one with generalization only, and one with generalization and suppression. A suppression limit of 100% is a reasonable parameter in the ARX system, as the implemented utility measures are able to automatically balance the application of generalization and suppression. We note that we also experimented with other suppression limits and obtained comparable results.
- Finally, we evaluated our approach with high-dimensional data with different numbers of quasi-identifiers. We investigated how the utility of the output of our algorithm improved over time. We also used suppression limits of 0% and 100% to cover two different transformation models.

In most experiments we focused on measures against identity disclosure, because it is widely accepted that these are relevant in practice [11]. In particular, we used the k-anonymity model with $k = 5$, which is a common parameter, e.g. in the biomedical domain [14]. Moreover, we enforced a threshold of 1% uniqueness within the US population estimated with the model by Dankar et al. [9]. When we compared our solution to previous algorithms, variability was rather high as a consequence of using minimal anonymity (cf. Section 4). Hence, we used additional privacy models and calculated workload averages. We chose the most well-known models against attribute and membership disclosure with risk thresholds which have been proposed in the literature [11, 27, 25, 29]. We used recursive-(c, ℓ)-diversity and t-closeness based on the earth mover's distance [25] using generalization hierarchies. As parameters, we chose $c = 4$, $\ell = 3$ and $t = 0.2$, which have been proposed in the literature [27, 25]. Moreover, we used δ-presence with $\delta_{min} = 0.05$ and $\delta_{max} = 0.15$ for a randomly selected research subset containing 10% of the entries of the respective dataset. These parameters have also been proposed in the literature [29].

We measured data utility with AECS, which is a model based on equivalence class sizes, and with Loss, which is a model that is evaluated independently for each attribute. We note that we have obtained similar results using other models [5].

The experiments were performed on a desktop machine with a quad-core 3.1 GHz Intel Core i5 CPU running a 64-bit Linux 3.2.0 kernel and a 64-bit Oracle JVM (1.7.0).

6.2 Datasets

In our evaluation we used six different datasets, most of which have already been utilized for evaluating previous work on data anonymization. For experiments with low-dimensional data, we used an excerpt of the 1994 US census database (ADULT), which is the de-facto standard dataset for the evaluation of anonymization algorithms, data from the 1998 KDD Cup (CUP), NHTSA crash statistics (FARS), data from the American Time Use Survey (ATUS) and data from the Integrated Health Interview Series (IHIS).

Dataset	Quasi-identifiers	Records	Transformations	Size [MB]
ADULT	8	30,162	4,320	2.52
CUP	7	63,441	9,000	7.11
FARS	7	100,937	5,184	7.19
ATUS	8	539,253	8,748	84.03
IHIS	8	1,193,504	12,960	107.56

Table 1: Overview of the datasets used for comparing algorithms

An overview of basic properties of the datasets is shown in Table 1. They feature between about 30k and 1.2M records (2.52 MB to 107.56 MB) with seven or eight quasi-identifiers and one sensitive attribute. The search spaces consisted of between 4,320 and 12,960 transformations.

Dataset	Quasi-identifiers (height of hierarchy)	Sensitive attribute (distinct values)
ADULT	sex (2), age (5), race (2), marital-status (3), education (4), native-country (3), workclass (3), salary-class (2)	occupation (14)
CUP	zip (6), age (5), gender (2), income (3), state (2), ngiftall (5), minramnt (5)	ramntall (814)
FARS	iage (6), irace (3), ideathmon (4), ideathday (4), isex (2), ihispanic (3), iinjury (3)	istatenum (51)
ATUS	region (3), age (6), sex (2), race (3), marital status (3), citizenship status (3), birthplace (3), labor force status (3)	highest level of school completed (18)
IHIS	year (6), quarter (3), region (3), pernum (4), age (5), marstat (3), sex (2), racea (2)	educ (26)

Table 2: Overview of attributes in the datasets used for comparing algorithms

Table 2 shows additional details about quasi-identifiers and sensitive attributes in the low-dimensional datasets. As can be seen, the generalization hierarchies used in our experiments featured between 2 and 6 generalization levels. The sensitive attributes in the datasets contained between 14 and 814 distinct values.

Dataset	QIs	Records	Transformations	Size [MB]
SS13ACS	15	68,725	51,018,336	2.06
SS13ACS	20	68,725	41,324,852,160	2.79
SS13ACS	25	68,725	17,852,336,133,120	3.36
SS13ACS	30	68,725	40,167,756,299,520,000	4.02

Table 3: Overview of the high-dimensional datasets used for evaluating scalability

For experiments with high-dimensional data we used a subset of responses to the American Community Survey (ACS), an ongoing survey conducted by the US Census Bureau on demographic, social and economic characteristics from randomly selected people living in the US [1]. It contains the data collected in the state of Massachusetts during the year 2013. Each of the 68,725 records represents the response of one particular person. We selected up to 30 of the 279 attributes of the dataset, focusing on typical quasi-identifiers, such as demographics (e.g. age, marital status, sex), information about insurance coverage, social parameters (e.g. education) and health parameters (e.g. weight, health problems).

An overview of the different instances used of the ACS dataset is provided in Table 3. Each representation contains 68,725 records. File sizes vary between about 2 MB and 4 MB. The datasets contain between 15 and 30 quasi-identifiers, which resulted in search spaces of between about $5 \cdot 10^7$ and $4 \cdot 10^{16}$ transformations.

7 Results

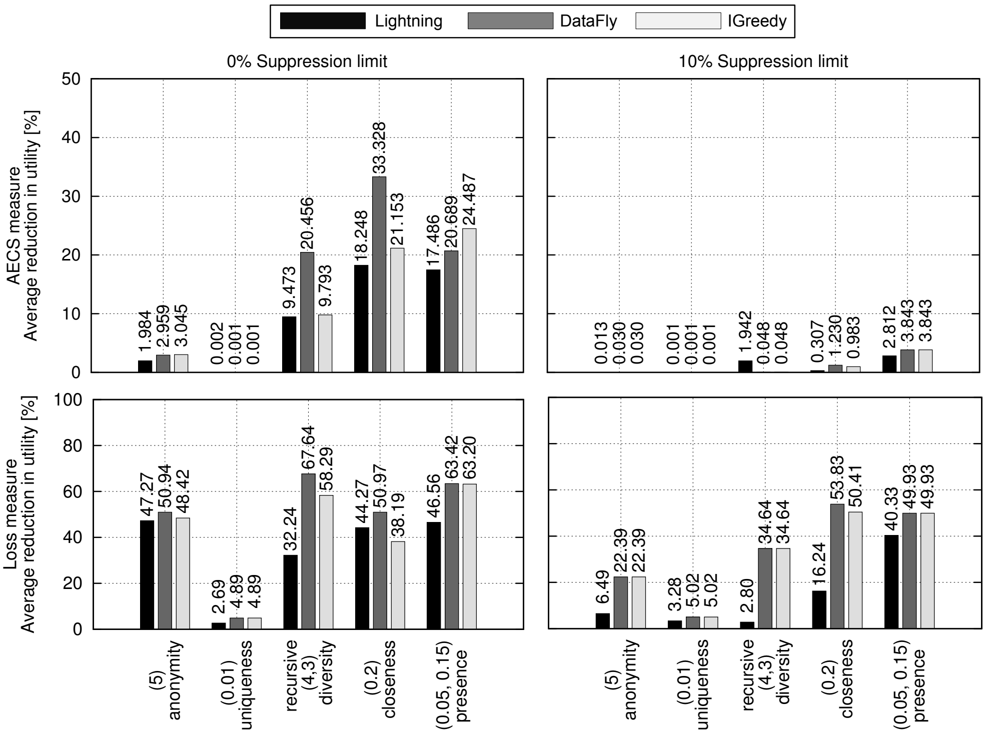


Figure 6: Overview of the average reduction in data utility as a result of performing minimal anonymization with Lightning, DataFly and IGreedy compared to an optimal solution (lower is better)

7.1 Comparison With Prior Work

A comparison of our approach with previous approaches is presented in Figure 6. It shows the reduction in data utility which resulted from performing minimal anonymization of the low-dimensional datasets with Lightning, DataFly and IGreedy compared to the optimal solution: 0% represents a solution with optimal data utility, 100% represents a solution in which all information has been removed from a dataset. The experiments were performed with five different privacy models, two different utility measures and two different suppression limits. We present workload averages which we defined as the geometric mean of the results for all five datasets. Detailed numbers, which emphasize the results presented in this section, can be found in Appendix B.

It can be seen that, on average, our approach outperformed previous solutions in terms of data quality in most cases. When our approach did not provide the best utility, the differences were almost negligible (e.g. (0.2)-closeness, 0% suppression limit, Loss and recursive-$(4,3)$-diversity, 10% suppression limit, AECS). This is an interesting finding, as Lightning was not designed to use minimality of the solution as a termination condition. The results strongly indicate that performing a heuristic search guided by the target utility measure is generally a good strategy. Our results also confirm the experiments by Babu et al. which show that, on average, IGreedy performs better than DataFly [3]. Moreover, the results show that heuristic algorithms which search for minimal solutions perform better when using a transformation model which includes record suppression. The reason is that with this transformation model, a smaller amount of generalization is required and the variability of the quality of different transformations is lower. Variability increases when more generalization is required, which means that heuristic search algorithms are less likely to discover a good solution. In the results of our experiments this is also reflected by a decrease of the algorithms' performance for privacy models which go beyond protection against identity disclosure and which therefore required more information to be removed from the datasets. Finally, the results also confirm our claim from Section 4 that different solutions which are all minimal in terms of the degree of generalization used are likely to have different properties in terms of data quality.

7.2 Comparison With a Globally-Optimal Algorithm

The aim of the experiments presented in this section was to compare our heuristic search, which does only use pruning mechanisms in configurations in which utility is monotonic, with a globally-optimal algorithm, which implements further pruning strategies. We used Flash for this purpose [20]. Interesting parameters include a comparison of the time required by Flash and Lightning to search the complete solution space (*Flash*, *Lightning*). Moreover, we investigated the quality of the result (*Utility*) found by the heuristic solution when executed with a time limit which equals the time required by the globally-optimal algorithm to characterize the complete solution space. Also, we report the time at which this solution was found (*Discovery*).

		0% Suppression limit				100% Suppression limit			
	Dataset	Flash [s]	Light. [s]	Discov. [s]	Utility [%]	Flash [s]	Light. [s]	Discov. [s]	Utility [%]
(5) anonymity	ADULT	0.033	1.394	–	–	0.847	1.375	0.048	100
	CUP	0.032	21.964	–	–	20.662	21.462	11.916	100
	FARS	0.061	2.669	–	–	2.219	2.766	1.752	100
	ATUS	0.217	19.097	–	–	11.109	18.421	0.424	100
	IHIS	2.168	144.978	–	–	51.573	170.599	1.082	100
(0.01) uniqueness	ADULT	0.184	0.110	0.076	100	9.359	10.028	0.074	100
	CUP	16.353	36.409	12.484	100	109.540	109.605	75.819	100
	FARS	7.554	20.947	5.237	∼100	40.145	40.775	16.419	100
	ATUS	0.067	0.058	0.058	100	273.027	285.699	0.058	100
	IHIS	0.195	0.165	0.165	100	689.753	791.150	0.164	100

Table 4: Comparison of Flash and Lightning for the AECS utility measure

Table 4 shows a comparison of Flash and Lightning for the AECS utility measure, the five low-dimensional datasets, suppression limits of 0% and 100% using (5)-anonymity and (0.01)-uniqueness. We measured significant differences in the time required to search the complete solution space. The differences were more significant when using (5)-anonymity, as Flash implements a dedicated pruning strategy for this privacy model [20]. Our heuristic strategy does not use this mechanism, because it is difficult to implement when the solution space is not explicitly represented in main memory. When using (0.01)-uniqueness both algorithms employed the same pruning methods. Here, the differences in execution times stem from the fact that the algorithms traverse the solution space in different ways. The ARX runtime environment implements several optimizations which can be exploited best when the solution space is traversed in a vertical manner [32]. On average, the depth-first strategy implemented by Flash was thus more efficient than the heuristic strategy, which also has a breadth-first component.

The fact that Flash implements more pruning strategies than Lightning is also the reason why the heuristic algorithm was not able to find a solution for (5)-anonymity with a suppression limit of 0% within Flash's execution time. In most other cases, the heuristic strategy found the optimal solution, or a solution which was very close to the optimum ((0.01)-uniqueness, 0% suppression limit, FARS datasets), in much less time than the globally-optimal algorithm needed to search the complete search space. The largest difference was measured when anonymizing the ATUS dataset with (0.01)-uniqueness and a suppression limit of 100%, where the time required by the heuristic strategy to find the optimum was only about 0.02% of the time required by the optimal algorithm to classify the solution space. However, the heuristic strategy would need to search the complete solution space as well to ensure that this is actually the optimum.

		0% Suppression limit				100% Suppression limit			
	Dataset	Flash [s]	Light. [s]	Discov. [s]	Utility [%]	Flash [s]	Light. [s]	Discov. [s]	Utility [%]
(5) anonymity	ADULT	0.027	1.579	–	–	1.172	1.582	0.042	100
	CUP	0.034	29.940	–	–	30.012	29.892	7.056	100
	FARS	0.062	3.674	–	–	2.907	3.703	0.097	100
	ATUS	0.219	32.625	–	–	13.84	33.304	0.204	100
	IHIS	2.203	127.193	–	–	62.067	147.184	1.438	100
(0.01) uniqueness	ADULT	0.628	0.280	0.090	100	21.543	22.745	0.090	100
	CUP	124.891	12.189	9.527	100	463.655	462.177	359.723	100
	FARS	15.808	4.783	3.138	100	66.313	68.251	2.778	100
	ATUS	0.157	0.148	0.148	100	373.064	380.183	0.149	100
	IHIS	0.966	0.939	0.939	100	968.625	1196.804	0.995	100

Table 5: Comparison of Flash and Lightning for the Loss utility measure

Table 5 shows the results of performing the same set of experiments with the Loss utility measure. It can be seen that with a 0% suppression limit and (0.01)-uniqueness, Lightning consistently outperformed Flash, even when searching the complete solution space. In these cases very little generalization was required and both algorithms used the same pruning strategies. Because Lightning starts its search at the bottom of the lattice, it quickly discovered the global optimum and was able to exclude the remaining transformations from the search process. Flash, however, starts its search in the center of the lattice and it therefore needed to check a few more transformations to discover the optimum. This resulted in significantly longer execution times, as evaluating this privacy model for a given transformation requires to repeatedly solve a non-linear equation system which is computationally complex. In the other experiments with the Loss utility measure we obtained results which are comparable to the results obtained when using AECS.

7.3 Evaluation With High-Dimensional Data

In this section we present the results of evaluating our algorithm with high-dimensional data. We anonymized the SS13ACS dataset with a selection of 15, 20, 25, and 30 quasi-identifiers as described in Section 6.2. This resulted in solution spaces consisting of between about $5 \cdot 10^7$ and $4 \cdot 10^{16}$ transformations. In each experiment Lightning was executed with a time limit of 600 seconds. We report the development of the utility of the anonymized dataset over time: 0% represents the result with lowest data utility and 100% represents the result with highest data utility found within the 600s time frame. The plots only show time ranges in which output quality did change.

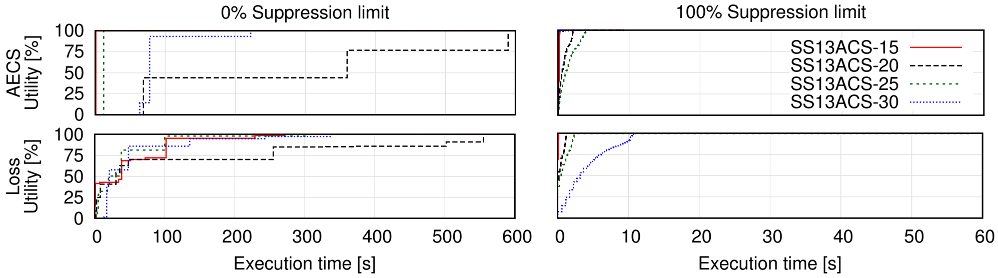


Figure 7: Development of output quality when using (5)-anonymity (higher is better).

Figure 7 shows the results obtained when using (5)-anonymity. It can be seen that the quality of the solution increased in finer steps and stabilized earlier when the transformation model involved generalization and suppression (no increase in utility after 58s compared to after 590s without suppression). Data quality improved in finer steps when using the Loss utility measure, especially with a 0% suppression limit. In no experiment the algorithm was able to search the complete solution space within the time limit of 600s. However, output quality often stabilized quickly. For example, with 15 quasi-identifiers the best solution was already found after 0.9s (0% suppression limit), and after 0.2s (100% suppression limit) respectively, when using the AECS measure.

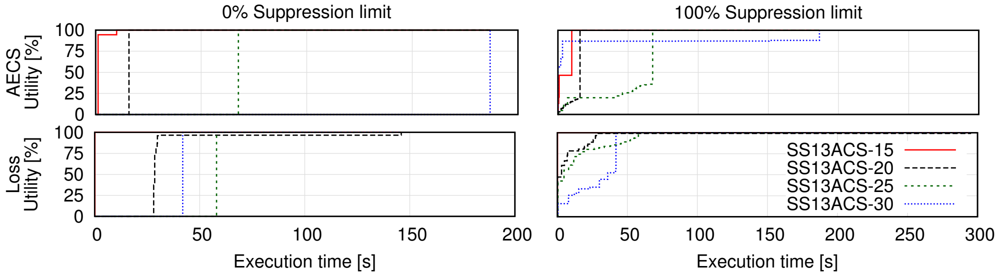


Figure 8: Development of output quality when using (0.01)-uniqueness (higher is better).

Figure 8 shows the results obtained when using (0.01)-uniqueness as a privacy model. Without record suppression the solutions stabilized more quickly than in the previous experiments. The reason for this is that (0.01)-uniqueness can be fulfilled with much less generalization than (5)-anonymity and the pruning strategy implemented by Lightning is therefore more effective. As a consequence, the algorithm was able to search the complete solution space in all experiments with the dataset with 15 quasi-identifiers and a 0% suppression limit. In all other experiments with a 0% suppression limit Lightning needed to search for a significant amount of time to find a solution. Moreover, no solution with better output quality could be found within the remaining time. With a record suppression limit of 100% utility improved more often during the 600s time frame than in the previous experiments.

8 Discussion and Conclusion

In this article, we have presented a heuristic search algorithm for anonymizing data with a broad spectrum of privacy models and models for measuring data utility. The key idea of our approach is to perform a best-first search which aims to optimize output quality. Our method combines this strategy with a greedy search to explore different areas of the solution space, including transformations with low degrees of generalization, transformations with high degrees of generalization and transformations in between these extremes.

In contrast to previous approaches the concept of minimal anonymity is not well suited in our context. The main reason is that we aim to support multiple privacy models, which means that neither disclosure risks nor data utility are guaranteed to be monotonic when a transformation model is used which combines full-domain generalization with class-based record suppression (see Section 4). In the results of our experiments this is reflected by the fact that different minimal solutions to most anonymization problems had very different properties in terms of data utility (see Section 7.1 and Appendix B). As a consequence, we designed our algorithm to terminate after a user-defined amount of time.

We have presented the results of an extensive evaluation in which we have compared our method to state-of-the-art algorithms for anonymizing data with the same transformation model. The experiments showed that our approach outperforms other heuristic algorithms in terms of output quality and that it is often able to discover a good solution quickly. It can therefore act as a valuable addition to globally-optimal algorithms, for example by first performing a heuristic search with a short execution time limit and using a globally-optimal algorithm only if the result is not satisfactory.

From a methodological perspective, we have presented a novel heuristic approach for solving a computationally complex problem. Of course, it is not guaranteed that our method performs as well as in our experiments when it is used in other setups. However, the same is true for previous approaches and we have put specific emphasis on evaluating our solution with a wide variety of different anonymization problems and datasets. The results indicate that our method performs well in practice. We have further put specific emphasis on evaluating our method with models which protect data from identity disclosure, because it is well-understood that these are of central relevance [11, 14]. We emphasize that all datasets and generalization hierarchies which we have used in the experiments are publicly available [34]. Moreover, our implementation of all methods which we have evaluated in our experiments is available as open source software [5].

In this work we have focused on heuristic anonymization algorithms which use global recoding with full-domain generalization. Other works have investigated heuristic methods for anonymizing data with different transformation models. For example, Fung et al. [16] and Xia et al. [41] have developed approaches using subtree generalization. The different heuristics use different objective functions. Fung et al. focus on finding a single solution with a good trade-off between privacy and utility [16], while Xia et al. aim to efficiently construct a risk-utility frontier [41]. Several algorithms have also been developed which transform data with microaggregation. Examples include the approach by Domingo-Ferrer and Torra for achieving k-anonymity of attributes with different scales of measure [10] and the approach by Soria et al. for combining k-anonymity with t-closeness [36]. An additional line of research involves methods which use local recoding with attribute generalization, for example, the approach by Goldberger and Tassa which supports k-anonymity and ℓ-diversity [17].

Different transformation models have different advantages and drawbacks. In ARX we have implemented the model described in this article because it can handle a wide variety of privacy models, it is intuitive and because it has been recommended for anonymizing data which is intended for use by humans (as opposed to, e.g., machine learning) [13]. On the other hand, with full-domain generalization more information may be removed than required [10]. With local recoding or subtree generalization this is not an issue, but the results are complicated to analyze [36]. Microaggregation has the potential to offer the best of both worlds, but in some domains, e.g. biomedical research, it has been argued that pertubative methods can not be used [12]. In ARX we have implemented methods for combining global recoding, local recoding and microaggregation [32]. However, these techniques are just a first step and in future work we plan to further investigate how the different types of data transformation can be integrated in a flexible and efficient manner.

In its current form our method is not well suited for anonymizing data with a very high number of quasi-identifying attributes (e.g. more than 50), as complex inter-attribute relationships will result in unacceptable reduction of data utility [2]. Approaches for anonymizing such data can be important for handling longitudinal information, for example in the biomedical domain where parameters are often collected repeatedly over a series of successive visits. One solution to this problem is to treat the data as transactional, i.e. set-valued, which is a way to remove inter-attribute relationships. Specific privacy models have been proposed for such data, for example k^m-anonymity [40] and (k, k^m)-anonymity [31]. In future work we plan to integrate similar methods into our tool as well.

One of the key features for facilitating user interaction in ARX is its visualization of the solution space. When using the heuristic algorithm, the tool only visualizes those parts which have been explored by the algorithm. However, it offers a method for dynamically expanding the solution space to transformations which have not yet been characterized. We note that by implementing the approach described in this paper, ARX is the first open source tool which supports automated anonymization of high-dimensional data.

Authors' contributions

FP, FK, RB, JE and HS designed and implemented the algorithm and the testbed for experiments. FP, RB, JE and HS performed the experiments. FP wrote the manuscript. RB, JE, HS, FK and KK helped to draft and revised the manuscript. KK contributed to the conception and design of the work at all stages. All authors have read and approved the final manuscript.

References

[1] U.S. Census Bureau - American Community Survey Main. http://www.census.gov/acs/www/. Accessed 03 Mar 2016.

[2] C. C. Aggarwal. On k-anonymity and the curse of dimensionality. In *Proceedings of the International Conference on Very Large Data Bases*, pages 901–909. Springer, 2005.

[3] K. S. Babu, N. Reddy, N. Kumar, M. Elliot, and S. K. Jena. Achieving k-anonymity using improved greedy heuristics for very large relational databases. *Transactions on Data Privacy*, 6(1):1–17, 2013.

[4] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *Proceedings of the International Conference on Data Engineering*, pages 217–228. IEEE, 2005.

[5] A benchmark of anonymization methods for high-dimensional data in ARX. https://github.com/arx-deidentifier/highdimensional-benchmark. Accessed 03 Mar 2016.

[6] J. Brickell and V. Shmatikov. The cost of privacy: Destruction of data-mining utility in anonymized data publishing. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 70–78. ACM, 2008.

[7] G. Chen and S. Keller-McNulty. Estimation of identification disclosure risk in microdata. *Journal of Official Statistics*, 14:79–95, 1998.

[8] L. H. Cox, A. F. Karr, and S. K. Kinney. Risk-utility paradigms for statistical disclosure limitation: How to think, but not how to act. *International Statistical Review*, 79(2):160–183, 2011.

[9] F. Dankar, K. El Emam, A. Neisa, and T. Roffey. Estimating the re-identification risk of clinical data sets. *BMC Medical Informatics and Decision Making*, 12(1):66, July 2012.

[10] J. Domingo-Ferrer and V. Torra. Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Mining and Knowledge Discovery*, 11(2):195–212, 2005.

[11] K. El Emam and C. Álvarez. A critical appraisal of the article 29 working party opinion 05/2014 on data anonymization techniques. *International Data Privacy Law*, 5:73–87, 2015.

[12] K. El Emam and L. Arbuckle. *Anonymizing Health Data: Case Studies and Methods to Get You Started*. O'Reilly and Associates, Sebastopol, 1 edition, 2014.

[13] K. El Emam, F. K. Dankar, R. Issa, E. Jonker, D. Amyot, E. Cogo, J. Corriveau, M. Walker, S. Chowdhury, R. Vaillancourt, T. Roffey, and J. Bottomley. A globally optimal k-anonymity method for the de-identification of health data. *Journal of the American Medical Informatics Association*, 16(5):670–682, 2009.

[14] K. El Emam and B. A. Malin. Appendix B: Concepts and methods for de-identifying clinical trial data. In Committee on Strategies for Responsible Sharing of Clinical Trial Data; Board on Health Sciences Policy; Institute of Medicine, editor, *Sharing Clinical Trial Data: Maximizing Benefits, Minimizing Risk*, pages 1–290. National Academies Press (US), Washington (DC), 2015.

[15] B. C. M. Fung, K. Wang, A. W.-C. Fu, and P. S. Yu. *Introduction to Privacy-Preserving Data Publishing: Concepts and Techniques*. CRC Press, 2010.

[16] B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *Proceedings of the International Conference on Data Engineering*, pages 205–216. IEEE, 2005.

[17] J. Goldberger and T. Tassa. Efficient anonymizations with enhanced utility. In *Proceedings of the International Conference on Data Mining Workshops*, pages 106–113. IEEE, 2009.

[18] N. Hoshino. Applying pitman's sampling formula to microdata disclosure risk assessment. *Journal of Official Statistics*, 17(4):499–520, 2001.

[19] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 279–288. ACM, 2002.

[20] F. Kohlmayer, F. Prasser, C. Eckert, A. Kemper, and K. A. Kuhn. Flash: Efficient, stable and optimal k-anonymity. In *Proceedings of the International Conference on Privacy, Security, Risk and Trust*, pages 708–717. IEEE, 2012.

[21] F. Kohlmayer, F. Prasser, C. Eckert, A. Kemper, and K. A. Kuhn. Highly efficient optimal k-anonymity for biomedical datasets. In *Proceedings of the International Symposium on Computer-Based Medical Systems*. IEEE, 2012.

[22] F. Kohlmayer, F. Prasser, and K. A. Kuhn. The cost of quality: Implementing generalization and suppression for anonymizing biomedical data with minimal information loss. *Journal of Biomedical Informatics*, 58:37–48, 2015.

[23] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proceedings of the SIGMOD Conference on Management of Data*, pages 49–60. ACM, 2005.

[24] R. R. LeFevre K, DeWitt DJ. Multidimensional k-anonymity (TR-1521). Technical report, University of Wisconsin, 2005.

[25] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and ℓ-

diversity. In *Proceedings of the International Conference on Data Engineering*, pages 106–115. IEEE, 2007.

[26] T. Li, N. Li, J. Zhang, and I. Molloy. Slicing: A new approach for privacy preserving data publishing. *Transactions on Knowledge and Data Engineering*, 24(3):561–574, 2012.

[27] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. In *Proceedings of the International Conference on Data Engineering*, pages 106–115. IEEE, 2007.

[28] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *Proceedings of the ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 223–228. ACM, 2004.

[29] M. E. Nergiz, M. Atzori, and C. Clifton. Hiding the presence of individuals from shared databases. In *Proceedings of the SIGMOD Conference on Management of Data*, pages 665–676. ACM, 2007.

[30] M. E. Nergiz and C. Clifton. Thoughts on k-anonymization. *Data & Knowledge Engineering*, 63(3):622–645, 2007.

[31] G. Poulis, G. Loukides, A. Gkoulalas-Divanis, and S. Skiadopoulos. Anonymizing data with relational and transaction attributes. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 353–369. Springer, 2013.

[32] F. Prasser and F. Kohlmayer. Putting statistical disclosure control into practice: The ARX data anonymization tool. In G. Loukides and A. Gkoulalas-Divanis, editors, *Medical Data Privacy Handbook*, pages 111–148. Springer International Publishing, 2015.

[33] F. Prasser, F. Kohlmayer, and K. A. Kuhn. A benchmark of globally-optimal anonymization methods for biomedical data. In *Proceedings of the International Symposium on Computer-Based Medical Systems*, pages 66 – 71. IEEE, 2014.

[34] F. Prasser, F. Kohlmayer, R. Lautenschlaeger, and K. A. Kuhn. ARX - a comprehensive tool for anonymizing biomedical data. In *Proceedings of the AMIA Annual Symposium*, pages 984–993, 2014.

[35] P. Samarati and L. Sweeney. Generalizing data to provide anonymity when disclosing information. In *Proceedings of the Symposium on Principles of Database Systems*. ACM, 1998.

[36] J. Soria-Comas, J. Domingo-Ferrer, D. Sanchez, and S. Martinez. t-closeness through microaggregation: Strict privacy with enhanced utility preservation. *Transactions on Knowledge and Data Engineering*, 27(11):3098–3110, 2015.

[37] L. Sweeney. Datafly: A system for providing anonymity in medical data. In *Proceedings of the International Conference on Database Security XI: Status and Prospects*, pages 356–381. Chapman & Hall, 1997.

[38] L. Sweeney. *Computational disclosure control - a primer on data privacy protection*. PhD thesis, Massachusetts Institute of Technology, 2001.

[39] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):571–588, 2002.

[40] M. Terrovitis, N. Mamoulis, and P. Kalnis. Privacy-preserving anonymization of set-valued data. In *Proceedings of the International Conference on Very Large Data Bases*. Springer, 2008.

[41] W. Xia, R. Heatherly, X. Ding, J. Li, and B. Malin. Efficient discovery of de-identification policy options through a risk-utility frontier. In *Proceedings of the ACM Conference on Data and Application Security and Privacy*, pages 59–70. ACM, 2013.

[42] L. V. Zayatz. Estimation of the percent of unique population elements on a microdata file using the sample. *Statistical Research Division Report Number: Census/SRD/RR-91/08*, 1991.

Appendix A

In this appendix, we will proof the non-monotonicity of ℓ-diversity within a transformation model consisting of generalization and suppression. For proofs regarding further privacy models we refer the interested reader to [22].

A privacy model is monotonic, if the fact that a dataset fulfills the model implies that any generalization of the dataset fulfills the model as well [4, 23]. It follows that in the type of solution spaces investigated in this work, all (direct and indirect) generalizations of a transformation which fulfills a privacy model also fulfill the privacy model [13]. It is easy to see that this implies the reverse as well: all (direct and indirect) specializations of a transformation which does not fulfill a privacy model will not fulfill the model either.

Distinct-ℓ-diversity is monotonic within a transformation model consisting of generalization and suppression [27]. In this section, we will analyze the monotonicity of two additional variants of ℓ-diversity: entropy-ℓ-diversity and recursive-(c, ℓ)-diversity. The former is the strictest instance of the ℓ-diversity model but it may not be achievable for some datasets. The latter is a more relaxed variant which aims at providing a good trade-of between utility and privacy [27]. We will use a counterexample to proof that both models are not monotonic within the given transformation model.

Less Generalized				More Generalized		
ID	Age	Diagnosis	Anonymity	Age	Diagnosis	Anonymity
0	[20-39]	Colon cancer	Recursive-(3,2)-diversity	[20-79]	Colon cancer	None
1	[20-39]	Stroke	Entropy-1.8-diversity	[20-79]	Stroke	
2	[20-39]	Colon cancer		[20-79]	Colon cancer	
3	[40-59]	Colon cancer	Recursive-(3,2)-diversity	[20-79]	Colon cancer	
4	[40-59]	Stroke	Entropy-1.8-diversity	[20-79]	Stroke	
5	[60-79]	Stroke	None	[20-79]	Stroke	
6	[60-79]	Stroke		[20-79]	Stroke	
7	[60-79]	Stroke		[20-79]	Stroke	
8	[60-79]	Stroke		[20-79]	Stroke	
9	[60-79]	Stroke		[20-79]	Stroke	
10	[60-79]	Stroke		[20-79]	Stroke	
11	[60-79]	Stroke		[20-79]	Stroke	
12	[60-79]	Stroke		[20-79]	Stroke	
13	[60-79]	Stroke		[20-79]	Stroke	
14	[60-79]	Stroke		[20-79]	Stroke	

Figure 1: Non-monotonicity of recursive-(3,2)-diversity and entropy-1.8-diversity with a suppression limit of 10 records

Figure 1 shows two generalizations of a dataset in which the attribute *age* is a quasi-identifier and *diagnosis* is a sensitive attribute for which recursive-(3,2)-diversity and entropy-1.8-diversity are to be achieved. The suppression limit is assumed to be 10 records.

We will use the following formalism: E is the set of equivalence classes in a dataset. For any class $e \in E$, $v(e)$ is the set of sensitive attribute values in the class. Moreover, $p(e, s)$ returns the relative frequency of a sensitive value s in $v(e)$.

Non-Monotonicity of Entropy-ℓ-Diversity

We will first review the formal definition of entropy-ℓ-diversity. We will then prove its non-monotonicity.

Definition

The entropy of a sensitive attribute in an equivalence class $e \in E$ is defined as

$entropy(e) = -\sum_{s \in e(v)} p(e,s) \cdot \log_2 p(e,s)$.

A dataset fulfills entropy-ℓ-diversity, if for all equivalence classes $e \in E$

$entropy(e) \geq \log_2 \ell$.

Proof of Non-Monotonicity

We first show that the dataset on the left fulfills entropy-1.8-diversity. To this end, we check whether the classes l_1, l_2 and l_3 fulfill the privacy model.

$entropy(l_1) = -(\frac{2}{3} \cdot \log_2(\frac{2}{3}) + \frac{1}{3} \cdot \log_2(\frac{1}{3})) = 0.9183 \geq log_2(1.8) = 0.8480.$
$entropy(l_2) = -(\frac{1}{2} \cdot \log_2(\frac{1}{2}) + \frac{1}{2} \cdot \log_2(\frac{1}{2})) = 1.0 \geq log_2(1.8) = 0.8480.$
$entropy(l_3) = -(\frac{10}{10} \cdot \log_2(\frac{10}{10})) = 0.0 < log_2(1.8) = 0.8480.$

This shows, that l_1 and l_2 fulfill entropy-1.8-diversity, whereas l_3 does not. As l_3 contains exactly 10 records it can be suppressed. As a consequence, the dataset fulfills entropy-1.8-diversity.

Next, we show that the generalized dataset on the right does not fulfill entropy-1.8-diversity.

$entropy(m_1) = -(\frac{3}{15} \cdot \log_2(\frac{3}{15}) + \frac{12}{15} \cdot \log_2(\frac{12}{15})) = 0.7219 < log_2(1.8) = 0.8480.$

This shows, that m_1 does not fulfill entropy-1.8-diversity. As the class contains more than 10 entries it cannot be suppressed. As a consequence, the dataset on the right does not fulfill entropy-1.8-diversity and therefore is a non-anonymous generalization of an anonymous dataset. This shows that entropy-ℓ-diversity is not monotonic within the transformation model investigated in this article. □

Non-Monotonicity of Recursive-(c, ℓ)-Diversity

We will first review the formal definition of recursive-(c, ℓ)-diversity. We will then prove its non-monotonicity. We will use the following formalism: $r(e,i)$ with $1 \leq i \leq m$ returns the frequency of the i-th frequent sensitive value in a class $e \in E$.

Definition

An equivalence class $e \in E$ fulfills recursive-(c, ℓ)-diversity if

$r(e,1) < c \cdot (r(e,\ell) + r(e,\ell+1) + ... + r(e,m))$.

A dataset fulfills recursive-(c, ℓ)-diversity, if all equivalence classes fulfill recursive-(c, ℓ)-diversity.

Proof of Non-Monotonicity

We first show that the dataset on the left fulfills recursive-$(3, 2)$-diversity. To this end, we check whether the classes l_1, l_2 and l_3 fulfill the criterion.

$r(l_1, 1) = 2 < 3 \cdot r(c_1, 2) = 3 \cdot 1 = 3.$
$r(l_2, 1) = 1 < 3 \cdot r(c_2, 2) = 3 \cdot 1 = 3.$
$r(l_3, 1) = 10 > 3 \cdot r(c_3, 2) = 3 \cdot 0 = 0.$

This shows, that l_1 and l_2 fulfill recursive-$(3, 2)$-diversity, whereas l_3 does not. As l_3 contains exactly 10 records it can be suppressed. As a consequence, the dataset fulfills recursive-$(3, 2)$-diversity.

Next, we show that the generalized dataset on the right does not fulfill recursive-$(3, 2)$-diversity.

$r(m_1, 1) = 12 > 3 \cdot r(m_1, 2) = 3 \cdot 3 = 9.$

This shows, that m_1 does not fulfill recursive-$(3, 2)$-diversity. As the class contains more than 10 entries it cannot be suppressed. As a consequence, the dataset on the right does not fulfill entropy-1.8-diversity and therefore is a non-anonymous generalization of an anonymous dataset. This shows that recursive-(c, ℓ)-diversity is not monotonic within the given transformation model. □

Appendix B

In this appendix, we present the detailed results from which the workload averages in Section 7.1 were compiled. The tables show a comparison of the data utility which resulted from anonymizing the five low-dimensional datasets (ADULT, CUP, FARS, ATUS, IHIS) with our approach, DataFly and IGreedy. We report the reduction in utility compared to an optimal solution: 0% represents a solution with optimal data utility, 100% represents a solution with the lowest possible data utility. All algorithms were configured to use minimal anonymity as a termination condition. The experiments were performed for (5)-anonymity and (0.01)-uniqueness with suppression limits of 0% and 10%. Utility was measured with AECS and Loss.

		0% Suppression limit			10% Suppression limit		
Dataset	Privacy model	Lightning	DataFly	IGreedy	Lightning	DataFly	IGreedy
ADULT	(5)-anonymity	**3.656**	**3.656**	**3.656**	**0.036**	0.117	0.117
CUP	(5)-anonymity	**5.263**	**5.263**	**5.263**	**0.009**	0.013	0.013
FARS	(5)-anonymity	**0.748**	**0.748**	**0.748**	0.019	**0.013**	**0.013**
ATUS	(5)-anonymity	**0.346**	4.652	4.652	**0.003**	**0.003**	**0.003**
IHIS	(5)-anonymity	**0.013**	0.570	0.989	**0.001**	0.002	0.002
ADULT	(0.01)-uniqueness	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**	**0.001**
CUP	(0.01)-uniqueness	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
FARS	(0.01)-uniqueness	0.007	**0.002**	**0.002**	**0.002**	**0.002**	**0.002**
ATUS	(0.01)-uniqueness	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
IHIS	(0.01)-uniqueness	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
ADULT	recursive-(4,3)-diversity	**4.274**	**4.274**	**4.274**	**0.086**	0.110	0.110
CUP	recursive-(4,3)-diversity	**8.696**	**8.696**	**8.696**	**0.008**	0.017	0.017
FARS	recursive-(4,3)-diversity	**0.709**	**0.709**	**0.709**	0.015	**0.008**	**0.008**
ATUS	recursive-(4,3)-diversity	**24.000**	100	32.444	9.972	**0.098**	**0.098**
IHIS	recursive-(4,3)-diversity	11.087	11.087	**5.530**	**0.002**	0.007	0.007
ADULT	(0.2)-closeness	**100**	**100**	**100**	0.293	**0.000**	**0.000**
CUP	(0.2)-closeness	**0.000**	**0.000**	**0.000**	0.204	**0.051**	**0.051**
FARS	(0.2)-closeness	**0.000**	**0.000**	**0.000**	0.304	**0.095**	**0.095**
ATUS	(0.2)-closeness	**12.727**	100	27.273	**0.000**	3.230	3.230
IHIS	(0.2)-closeness	**2.544**	5.328	**2.544**	**0.737**	2.826	1.578
ADULT	(0.05,0.15)-presence	**7.895**	**7.895**	**7.895**	**3.556**	**3.556**	**3.556**
CUP	(0.05,0.15)-presence	**100**	**100**	**100**	**6.173**	**6.173**	**6.173**
FARS	(0.05,0.15)-presence	**3.538**	**3.538**	**3.538**	**4.395**	**4.395**	**4.395**
ATUS	(0.05,0.15)-presence	**0.160**	14.027	33.132	**0.082**	4.617	4.617
IHIS	(0.05,0.15)-presence	**0.024**	0.509	0.509	**0.002**	0.559	0.559

Table 1: Relative information loss [%] of Lightning, DataFly and ImprovedGreedy for the AECS utility measure (lower is better, best result has been highlighted).

Table 1 shows the results obtained with the AECS utility measure. It can be seen that the individual results support the conclusions drawn from workload averages in Section 7.1. Without record suppression, from a total of 25 experiments, our approach returned the best result of all three algorithms in 23 experiments. In contrast, IGreedy found the best solution in 19 and DataFly in 17 experiments. In 17 out of 25 experiments all three approaches returned the same result. With a 10% suppression limit, our approach returned the best results in 19 experiments. In contrast, IGreedy and DataFly returned the best solution in 15 experiments, respectively. In 8 out of 25 experiments, all three approaches returned the same result.

Table 2 shows the results obtained with the Loss utility measure. Without record suppression, from a total of 25 experiments, our approach returned the best result of all three algorithms in 20 experiments. In contrast, IGreedy found the best solution in 11 and DataFly in 9 experiments. In 6 out of 25 experiments, all three approaches returned the same result. With a 10% suppression limit, our approach returned the best results in 23 experiments. In contrast, IGreedy and DataFly returned the best solution in only 8 experiments. In 7 out of 25 experiments all three approaches returned the same result.

		0% Suppression limit			10% Suppression limit		
Dataset	Privacy model	Lightning	DataFly	IGreedy	Lightning	DataFly	IGreedy
ADULT	(5)-anonymity	**30.49**	65.729	65.729	**4.152**	26.724	26.724
CUP	(5)-anonymity	100	**35.024**	**35.024**	**14.879**	66.364	66.364
FARS	(5)-anonymity	**47.526**	**47.526**	**47.526**	**1.612**	15.668	15.668
ATUS	(5)-anonymity	**16.564**	53.744	53.744	**7.652**	**7.652**	**7.652**
IHIS	(5)-anonymity	54.373	54.373	**41.907**	**4.632**	**4.632**	**4.632**
ADULT	(0.01)-uniqueness	**8.597**	**8.597**	**8.597**	**8.597**	**8.597**	**8.597**
CUP	(0.01)-uniqueness	**3.93**	12.972	12.972	**3.93**	12.972	12.972
FARS	(0.01)-uniqueness	**1.155**	3.476	3.476	**4.128**	**4.128**	**4.128**
ATUS	(0.01)-uniqueness	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
IHIS	(0.01)-uniqueness	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
ADULT	recursive-(4,3)-diversity	**0.000**	65.729	65.729	**3.149**	28.518	28.518
CUP	recursive-(4,3)-diversity	**100**	51.395	51.395	**6.48**	65.669	65.669
FARS	recursive-(4,3)-diversity	**19.726**	47.526	47.526	**2.827**	14.231	14.231
ATUS	recursive-(4,3)-diversity	**9.79**	100	71.563	9.79	47.107	47.107
IHIS	recursive-(4,3)-diversity	**53.815**	78.819	56.484	**1.657**	23.657	23.657
ADULT	(0.2)-closeness	**33.736**	100	100	**12.02**	67.026	67.026
CUP	(0.2)-closeness	**13.169**	**13.169**	**13.169**	**46.284**	77.471	77.471
FARS	(0.2)-closeness	100	**0.000**	**0.000**	**13.985**	52.024	52.024
ATUS	(0.2)-closeness	**3.224**	100	51.351	**2.435**	19.82	19.82
IHIS	(0.2)-closeness	100	73.217	**47.104**	**10.896**	59.545	42.559
ADULT	(0.05,0.15)-presence	**19.616**	63.135	63.135	**0.000**	65.76	65.76
CUP	(0.05,0.15)-presence	**100**	**100**	**100**	100	**38.109**	**38.109**
FARS	(0.05,0.15)-presence	**30.439**	34.231	34.231	**40.621**	43.858	43.858
ATUS	(0.05,0.15)-presence	**23.331**	73.485	72.306	**25.729**	49.483	49.483
IHIS	(0.05,0.15)-presence	75.702	**53.422**	**53.422**	**53.876**	**53.87**	**53.87**

Table 2: Relative information loss [%] of Lightning, DataFly and ImprovedGreedy for the Loss utility measure (lower is better, best result has been highlighted).

The experiments clearly show that, on average, our approach outperforms previous solutions in terms of data quality. Moreover, the difference in quality of data output is much higher when utility is measured with Loss instead of AECS. As this measure captures fine-grained differences in the quality of anonymized datasets, our strategy seems to be of general relevance. Additionally, our results confirm the experiments by Babu et al. which show that IGreedy performs better than DataFly [3].