A Practice-oriented Framework for Measuring Privacy and Utility in Data Sanitization Systems

Michal Sramka*
Department of Computer Engineering and Maths, Universitat Rovira i Virgili
Av. Paisos Catalans, 26, 43007 Tarragona, Spain
michal.sramka@urv.cat

Reihaneh Safavi-Naini, Jörg Denzinger, and Mina Askari
Department of Computer Science, University of Calgary
2500 University Drive NW, Calgary, Alberta T2N 1N4, Canada
{rei,denzinge,maskari}@ucalgary.ca

ABSTRACT

Published data is prone to privacy attacks. Sanitization methods aim to prevent these attacks while maintaining usefulness of the data for legitimate users. Quantifying the trade-off between usefulness and privacy of published data has been the subject of much research in recent years. We propose a pragmatic framework for evaluating sanitization systems in real-life and use data mining utility as a universal measure of usefulness and privacy. We propose a definition for data mining utility that can be tuned to capture the needs of data users and the adversaries' intentions in a setting that is specified by a database, a candidate sanitization method, and privacy and utility concerns of data owner. We use this framework to evaluate and compare privacy and utility offered by two well-known sanitization methods, namely k-anonymity and ϵ-differential privacy, when UCI's "Adult" dataset and the Weka data mining package is used, and utility and privacy measures are defined for users and adversaries. In the case of k-anonymity, we compare our results with the recent work of Brickell and Shmatikov (KDD 2008), and show that using data mining algorithms increases their proposed adversarial gains.

1. INTRODUCTION

Many databases contain data about individuals that are valuable for research, marketing, and decision making. The main concern in releasing this data is the loss of individuals' privacy. *Data sanitization* methods aim at making data publishable while providing protection guarantees against disclosures and at the same time maintaining the usefulness of the data. Two required properties of sanitized data are the *usefulness* of the data for data users, sometimes referred to as utility, and protection of individuals' information from malicious users, referred to as *privacy*. There is a trade-off between these two properties and sanitization methods need to balance them.

Quantifying privacy and usefulness has been a long standing problem. One challenge is that privacy and utility are scenario dependent. For example, utility in one scenario can be exact predictions of a particular attribute for the whole population, while in another scenario only approximate predictions for a specific group of users may be required. Similarly, privacy requirement is scenario dependent: in one case the same level of privacy is required for all users, while in another a selected group of users need higher protection for particular attributes.

Many well-founded schemes, such as ϵ-differential privacy [7, 8] and k-anonymity [23, 28], provide guarantees only for privacy and not for usefulness, although the need for usefulness, to "keep the data truthful", has always been recognized.

In practice, a data owner needs to choose a suitable sanitization method among a number of possible alternatives for a particular database such that some intended data users find the data useful, while the privacy loss is below a defined level. Here the notions of privacy and usefulness of data are defined by the data owner and the data owner needs a pragmatic and systematic way of comparing candidate sanitization methods by quantifying their effect on privacy and usefulness of data.

Here we provide a framework that can be specialized to specific scenarios - modeling privacy and usefulness notions and quantifying their levels for the given database - and so providing a decision support mechanism for the data owner to select the appropriate algorithm. It is worth emphasizing that the power of the framework is in its adaptability to capture various notions of privacy, utility and adversarial power for comparing sanitization systems for a particular setting. The privacy and utility levels measured by the framework must be interpreted in this context.

1.1 Our Contribution

We propose a pragmatic framework to evaluate and compare sanitization methods. The centerpiece of this framework is a set of data mining algorithms that learn patterns and properties of data. This knowledge can then be used by data users for legitimate purposes and by privacy adversaries for disclosing private and sensitive information about individuals in the database.

We propose a definition of utility for data mining algorithms and use it to quantify both privacy and usefulness of the sanitized data. We define utility of a mining algorithm when applied to sanitized data in terms of the predictions

*This work was done while the first author was with the Department of Computer Science, University of Calgary, and was supported by Informatics Circle of Research Excellence, Alberta.

that it can make about the original data. We model the utility of data users using weights for records in the database to show the relative importance of the record, and an appropriate error function to show the "cost" of making errors in predictions. This is called "good utility". The utility function can also be used to express utility of an adversary by measuring correct predictions of sensitive attributes for individuals in the database. This is called "bad utility". The proposed definition of utility is powerful in the sense that it provides a flexible way of modeling, (i) the utility of legitimate data users, and (ii) the gain of real-world privacy adversaries in attacking published data sets.

Evaluation of sanitization methods will be with respect to, (i) a target database, (ii) a specific set of miners, (iii) the requirements of data users captured by the "good utility", and (iv) a specific privacy concern of the data owner captured by the "bad utility". In other words, the evaluation will be with respect to *scenarios* that consist of the above four components. The set of data miners used by the users and the adversary can be different. This means that the result of the evaluation could change when the scenario changes. This is natural given the variety and complexity of privacy and utility notions in practice, and the fact that data owners will be mainly concerned about their own database.

We focus on a single database. In practice the adversary may have access to other databases which would increase his/her prediction capabilities. We leave the challenging question of considering the effect of (auxiliary) information from other sources on privacy of data for future work.

We show the power and effectiveness of our framework through a number of experiments. We first consider a comparison of k-anonymity [23, 28] and ϵ-differential privacy [7, 8] when used as a privacy enhancing mechanism for publishing the data from a database. We present the usefulness of the data in terms of classification accuracy that we express using our proposed utility measure. Similarly, we present privacy in terms of better predictions about the identifying attributes given the sanitized data, again expressed using our proposed utility measure.

The results of our experiments show how the performance of the two sanitization methods depends on the scenario at hand; that is, on the given database, set of miners, and on the definition of good and bad utilities. The results of the experiments confirm the difficulty of making general statements about comparing sanitization methods.

Our approach to evaluate sanitization methods is pragmatic and can be seen as providing a decision support mechanism for data owners who are faced with the question of which sanitization method is more suitable for their "purpose". Using the framework proposed in this paper, data owners will be able to estimate the risks involved with each choice. Our framework allows these differences to be systematically taken into account.

Using data mining as the primary tool of the adversaries and data users is a good approximation of how sanitized data is used by the legitimate users as well as how it is compromised by the adversaries. Data mining adversaries capture a large class of automated attacks on published data sets and hence our framework provides a baseline evaluation method for evaluating privacy guarantees afforded by sanitization methods. By expanding the set of data miners one may consider a wider class of attacks and thus a privacy guarantee against stronger attackers.

Privacy evaluation using data mining considers the sanitized data as a whole, and not individual records, and so allows discovery of hidden patterns and trends in the data. This is a strategy that is available to the adversary. We note that although not all adversarial strategies can be captured using a finite set of miners but the approach provides a first step towards a framework for systematic comparison of privacy and usefulness of different sanitization methods.

1.2 Related Work

The trade-off between the data mining utility and privacy for anonymization methods is considered in [4, 21]. These works however do not consider privacy against attackers with access to data mining algorithms.

In particular, Brickell and Shmatikov [4] recently studied the trade-off between privacy and utility of k-anonymization and some of its derivatives. A natural question is how their results compare with the framework proposed in this paper when applied to k-anonymity. (We note that our framework is general and applicable to any sanitization system and the comparison is only for the special case that the framework is applied to k-anonymity.) To answer this question, we express Brickell and Shmatikov's measures in our framework– Section 3. The utility measure in [4] can be expressed in terms of our proposed utility measure. The privacy measure in [4] is the "gain of the adversary" measured in the terms of reduction in their uncertainty, after seeing the sanitized database. By applying their definition of adversarial gain to the "predicted database", that is, the approximation of the original database reconstructed from the sanitized one by applying a data mining algorithm, we show that the real adversarial gain is in fact higher than what is predicted by their work. This indicates the importance of considering adversaries using data miners in the evaluation of privacy.

The need to consider semantic privacy which captures the shift in adversarial knowledge has been also recognized in randomized/perturbation models [9, 17]. The usefulness of sanitized data is evaluated using data mining utility in perturbation/randomized response methods [2]. All of the sanitization mechanisms that we know of are targeting a specific mining goal (in contrast to our work that allows scenario dependent goals). For anonymization methods, the usefulness of the data is mostly measured syntactically (as the number and amount of generalizations), but classification accuracy is also used as a measure in [10, 4, 25, 26]. We have also used data mining not to measure privacy, but to attack the privacy and make disclosures from sanitized data [24].

Data sanitization can be used to publish data that contains information about individuals. One approach, coming from statistical databases and interactive randomized response models, is to randomize (for example, add noise to) the values of individual records, and only release these records [1, 2, 9, 11, 30, 31]. The amount of noise can be limited using privacy notions, such as ϵ-differential privacy [7, 8], which we briefly discuss in Section 4.2. The other common approach, called anonymization, involves releasing records while making individuals hard to distinguish by masking (generalizing and suppressing) their identifying values: k-anonymity [22, 23, 28, 6] is a well-researched method [10, 13, 16, 18, 19, 20, 27, 12]. We again briefly discuss k-anonymity in Section 4.2. Other work in data anonymization includes ℓ-diversity [15], p-sensitive k-anonymity [29], (α, k)-anonymity [33], t-closeness [14], and

others.

2. FRAMEWORK

We concentrate on privacy of large amounts of data that are used for data mining. The general scenario is depicted in Fig. 1. We start by defining the key components of the framework, then model the real-world adversaries and end users using the same methodology, and finally propose practical and versatile measures that capture these ideas.

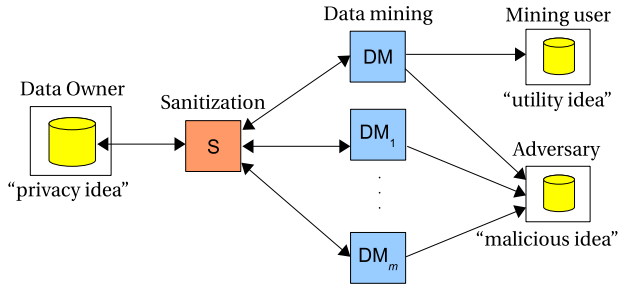


Figure 1: Scenario

The data from users can be horizontally or vertically partitioned and is collected by a trusted data collector in a single collection. The data collector becomes the single *data owner*, who is responsible for protecting the privacy of the data and for allowing the data to be used for legitimate purposes. A *mining user* (user of the data mining results) is an entity that wants to perform some data mining over the data collection. The mining user represents an end user of the data that wants to use the data for legitimate research or testing purposes. An *adversary* is just another mining user, differing in the mining task, goal, and idea. The adversary represents an end user whose aim is malicious.

The users supplying the data and the data owner have "privacy ideas", while the mining user has a "utility idea". A *privacy idea* captures the collective privacy wishes and expectations of the data owner and data supplying users about disclosing the data or parts of it. A *utility idea* refers to the usefulness of the results of data mining for the mining user. A *malicious idea* (malicious utility idea) refers to a utility idea that is in contradiction to any privacy idea of any data owner. In this sense, the adversary is a mining user whose utility idea is malicious, that is, it is in contradiction with the legitimate use of the data captured in the privacy and utility ideas. To achieve the malicious idea, the adversary may use some *auxiliary information*, which is some data that by itself does not violate the privacy ideas, that is, by itself it is not a disclosure. In contrast to others, we do not assume any auxiliary information. Even without any auxiliary information or background knowledge, we show that data mining can be used to measure privacy loss.

The data should be provided for data mining purposes to satisfy the utility idea of the mining user, yet it should be sanitized in order to satisfy the privacy ideas of the data owner and counter the malicious ideas of *any* adversary. A *sanitization mechanism* is a privacy-preserving algorithm that transforms and releases the original data that may contain sensitive values and relations in a way that preserves and protects the privacy of these sensitive values and relations. In this framework, the sanitization is performed by the data owner before the query results obtained from the data leave the data collection. The sanitized data may then be accessed in one of two models of interaction: *non-interactive* and *interactive* models [7]. In the first model, the sanitized data is published and becomes accessible by the public, while in the second model users can only access the data through a sanitization mechanism that transforms the responses to individual queries before delivering them to the user. We define a model that captures these ideas, and measures the privacy and usefulness using the same methodology but reflecting the choices, decisions, and purposes of the data defined by the data owner.

2.1 Modeling the Privacy and Utility Ideas

The core component of our framework consists of data mining algorithms. They are used for measuring and establishing both usefulness and privacy. We use the data mining utility to capture both the needs of mining users and intentions of adversaries.

We model the "privacy idea" of the data owner as a *bad* utility $\mathcal{U}_{\mathsf{bad}}$. We also refer to this bad utility as an *anti-utility*. The bad utility $\mathcal{U}_{\mathsf{bad}}$ reflects the possibility of obtaining protected information that the data owner is trying to keep private, and such a possibility is bad for the owner. This includes information such as identities or sensitive values and the relations among them. The information is theoretically obtained as a result of using *any* data mining algorithm over the data sanitized with a sanitization mechanism $\mathcal{S}$. In practice, we consider a reference set of data mining algorithms that are likely to be utilized by an adversary. For example, the already mentioned Weka data miner collection [32] can be used with very little knowledge about data mining and thus can be easily used by expert or non-expert adversaries, like reporters or noisy kids. For privacy reasons, we want the value of such information to be low, hence the bad utility $\mathcal{U}_{\mathsf{bad}}$ should be small.

We model the non-malicious "utility idea" of the mining user as a *good* utility $\mathcal{U}_{\mathsf{good}}$. It reflects the usefulness of the data mining results that are not in contradiction to the "privacy ideas", that is, they do not disclose protected information and relations. The results are obtained using a single fixed data mining algorithm operating over data that has been sanitized by a sanitization mechanism $\mathcal{S}$. Obviously, the mining user wants this utility value to be high, preferably as high as obtainable from the unsanitized original data.

The data owner classifies a sanitization mechanism $\mathcal{S}$ as *satisfying the privacy and utility ideas*, if the good utility $\mathcal{U}_{\mathsf{good}}$ for satisfying the utility idea of the mining user is acceptably high, and the anti-utility $\mathcal{U}_{\mathsf{bad}}$, representing the privacy idea of the data owner, is below a threshold of his/her choice. This notion for privacy and utility fits the concept of a relative privacy guarantee [7], and at the same time it conforms to the need of looking for useful triples (utility definition, sanitization mechanism, data miner) [25]. The measures of utility, acceptable level for good utility and the threshold for anti-utility are all choices of the data owner. Thus the statements about the usefulness or the privacy guarantees of the released sanitized data are dependent on these choices and the data, and general statements about a sanitization method may be contradictory and misleading.

2.2 Sanitization and Prediction

We model the database of the data owner as a universe of tuples $\mathbb{D} = D_1 \times \cdots \times D_n$, where the D_i's denote some domains. An element $x = (x_1, \ldots, x_n) \in \mathbb{D}$ is called a *tuple* and a coordinate x_i of x is referred to as a *field*. A *database* is a finite subset of $\mathbb{D}$, denoted as DB. The symbol $\perp$ represents a missing value and $\perp \in D_i$ for all $1 \leq i \leq n$, that is, we allow incomplete tuples or, in other words, databases that are missing some fields for some tuples. This all corresponds to the relational database concept.

A *sanitization mechanism* $\mathcal{S}$ is a randomized algorithm that for a database DB and query f returns a transformation of the query result $f(DB)$ that conforms to the privacy guarantee of $\mathcal{S}$. A *privacy guarantee* of $\mathcal{S}$ is a notion, a description of what is meant and provided by the sanitization mechanism and how sensitive data, identities, and sensitive relations are protected by $\mathcal{S}$. Allowed queries for $\mathcal{S}$ depend on $\mathcal{S}$ itself. For example, k-anonymity can be modeled as using only "select all" queries, while sanitization mechanisms protecting statistical databases can be seen as allowing only statistical queries. ϵ-differential privacy allows queries that map the database to real values. We assume that the sanitization algorithm and its parameters are known to the adversary, while the database DB and the randomness used during sanitization is private to the data owner. For simplicity, in the rest of the paper, we concentrate on the non-interactive scenario, one where the query preserves the structure of DB after sanitization. Furthermore, we assume that no records are suppressed during sanitization, and so there is a one-to-one correspondence between DB and its sanitized version. Then by $\mathcal{S}(DB)$, $\mathcal{S}(x)$, and $\mathcal{S}(x)_i$, we denote the database DB, tuple x, and field x_i, respectively, after sanitization.

A *data miner* or simply a *miner* $\mathcal{M}^i$ is an algorithm. Its input is a database DB and a tuple x. The miner $\mathcal{M}^i$ determines trends and patterns in the database DB and based on them it outputs the prediction of the i-th field for the tuple x, denoted as $\hat{x}_i$. The true value of the i-th field is denoted as $\bar{x}_i$. In addition, a data miner $\mathcal{M}^i$ can operate over a sanitized database $\mathcal{S}(DB)$ and predict the i-th field for a sanitized tuple $\mathcal{S}(x)$, that is, $\mathcal{M}^i(\mathcal{S}(DB), \mathcal{S}(x))$ would predict the i-th field value of $\mathcal{S}(x)$, denoted as $\widehat{\mathcal{S}(x)_i}$.

2.3 Measuring Utility and Utility Preservation After Sanitization

We first present definition and instantiation of the utility of a particular data mining process, then we describe the flexibility and versatility of the instantiation. We measure good utility $\mathcal{U}_{\mathsf{good}}$ for only one field, denoted by index i. This measure can be extended to multiple fields and their combinations.

An *error implication function* $E_i(\hat{x}_i, \bar{x}_i)$ weights the seriousness of any occurring error for the utility. For example, $E_i(\hat{x}_i, \bar{x}_i)$ can be some value based on the predicted $\hat{x}_i$ and the true $\bar{x}_i$ that accounts for the preferences of a user who measures the utility. Note that we want the function E_i to have a high value if there is no error, while it should be near 0 or even negative for serious errors. Such a choice reflects the fact that higher utility values mean higher usefulness. In addition, the user may be interested in tuples with a certain interest factor, therefore the user decides on a *weight function* $w(x)$ for each tuple $x \in DB$, where higher weights for tuples show that the user is more interested in those tuples.

A *utility of prediction* can be measured for the original unsanitized data and for the sanitized data:

$$\mathcal{U}^{(\mathsf{orig})}_{\mathsf{good}}(DB, \mathcal{M}^i) = \sum_{x \in DB} w(x) E_i\Big(\mathcal{M}^i(DB, x),\ \bar{x}_i\Big) \text{ and,}$$

$$\mathcal{U}^{(\mathsf{san})}_{\mathsf{good}}(DB, \mathcal{S}, \mathcal{M}^i) = \sum_{x \in DB} w(x) E_i\Big(\mathcal{M}^i(\mathcal{S}(DB), \mathcal{S}(x)),\ \bar{x}_i\Big).$$

The error implication function $E_i(\hat{x}_i, \bar{x}_i)$ and the weight function $w(x)$ allow the data owner to capture and effectively model the needs of a large class of the mining users. For example, classification accuracy can be captured using $E_i(\hat{x}_i, \bar{x}_i) = 1$ if the two inputs are equal and 0 otherwise, and a constant weight $w(x) = 1$. The utility for a medical researcher interested in an early detection of a disease mainly in a population of women above 55 can be captured using $E_i(\hat{x}_i, \bar{x}_i) = 1$ if $\hat{x}_i - \bar{x}_i \geq 0$ and 0 otherwise, that is, she is accepting false-positive predictions, meaning she is willing to test extra people for the disease, as testing and prevention is cheaper than treatment. Further, she sets the weight $w(x) = 10$ for all women above 55, and $w(x) = 1$ or $w(x) = 0$ for others, which represents her interest in the population. Also, missing a correct prediction can be penalized by selecting negative values for the proper cases in the error implication function.

We now introduce the definition of a $(1 - \delta, L)$-utility-preserving sanitization mechanism $\mathcal{S}$. Our definition guarantees that the sanitized data would be useful for research, analysis, mining, and testing purposes. In the definition, the *utility decline* $\delta \in [0, 1]$ can be seen as the decline of the utility value in using any of the miners in the set L over the privacy-preserved sanitized data compared to mining with the same miners over the original unsanitized data. $(1 - \delta)$ is then the fraction that represents the utility preservation.

Since the data owner usually does not know in advance what the mining users want to obtain, the owner measures the usefulness using a few data mining algorithms represented in the set L that cover a wide range of possible users' desires. The usefulness provided by these miners should provide usefulness of the data in general.

A sanitization mechanism $\mathcal{S}$ operating on a database DB is said to be $(1 - \delta, L)$-*utility-preserving* if and only if

$$\mathcal{U}^{(\mathsf{san})}_{\mathsf{good}}(DB, \mathcal{S}, \mathcal{M}^i) \geq (1 - \delta) \cdot \mathcal{U}^{(\mathsf{orig})}_{\mathsf{good}}(DB, \mathcal{M}^i)\ ,$$

for every miner $\mathcal{M}^i \in L$ and a user selected error implication function $E_i(\hat{x}_i, \bar{x}_i)$ and weight function $w(x)$. In the case when $\mathcal{U}^{(\mathsf{san})}_{\mathsf{good}}(DB, \mathcal{S}, \mathcal{M}^i) \geq \mathcal{U}^{(\mathsf{orig})}_{\mathsf{good}}(DB, \mathcal{M}^i)$, we say that the decline is $\delta = 0$.

For the same miner or a set L of miners, the definition of a $(1 - \delta, L)$-utility-preserving sanitization mechanism allows the data owner to compare sanitization methods and their output on his/her database from the usefulness point of view. Clearly, the smaller the utility decline δ, the higher the utility preservation $1 - \delta$, and the higher the value of the released sanitized data for the mining users.

2.4 Measuring Privacy Breaches in Sanitized Data Using Utility

Based on a reference set of miners M, we define the concept of anti-utility. We describe the flexibility and versatility of the definition and provide an instantiation. We measure the anti-utility for only one field, denoted by index j to indicate that the privacy is evaluated over a different field than

in the case of the usefulness evaluation. As in the previous case, this measure can be extended to multiple fields.

An adversary may be interested in attacking the privacy of the same field that is of interest to legitimate users. But to achieve its goal, the adversary usually tries to reconstruct the relations between the attacked field and other fields in order to identify and "link" the individual. The results of mining may not be what the adversary is looking for, but the mining will help the adversary in achieving its malicious goal. In case the adversary mines over the same field, it necessarily learns the same amount as the legitimate user. Therefore we consider what happens when the adversary mines over a different field.

An *error reduction function* $E_j^*(\hat{x}_j, \mathcal{S}(x)_j, x_j)$ measures the reduction of the given sanitized value $\mathcal{S}(x)_j$ toward the original value x_j by considering the prediction $\hat{x}_j = \mathcal{M}^j(\mathcal{S}(DB), \mathcal{S}(x))$. The value of E_j^* represents the significance of reducing the uncertainty about the original value x_j by having a prediction $\hat{x}_j$. Higher utility values mean higher usefulness, and so we want the function E_j^* to have a high value if there is no or only a small error in prediction, while it should be near 0 or even negative for serious errors. Furthermore, an adversary may be interested in tuples with a certain interest factor, and we model this by a *weight function* $v(x)$ for each tuple $x \in DB$. Note that this weight function $v(x)$ may be rather different from the weight function $w(x)$ used for establishing the usefulness of the sanitized data. The functions E_j^* and v can be seen on one hand as describing and modeling the interests of adversaries, but on the other hand also as modeling what the data owner wants to protect.

A *utility of privacy-impacting prediction* obtained by a miner $\mathcal{M}^j$ is then measured as

$$\mathcal{U}_{\mathsf{bad}}(DB, \mathcal{S}, \mathcal{M}^j) = \sum_{x \in DB} v(x) E_j^*\Big(\mathcal{M}^j(\mathcal{S}(DB), \mathcal{S}(x)),\, \mathcal{S}(x)_j,\, x_j\Big) \ ,$$

which may be seen as $\mathcal{U}_{\mathsf{good}}^{(\mathsf{san})}(DB, \mathcal{S}, \mathcal{M}^j)$ computed with a different miner, over a different field, and using different error and interest functions. These differences allow us to model the adversary and distinguish it from the legitimate mining user. For example, an adversary may be interested in breaching the privacy for a selected few people (say, famous people or celebrities). The data owner can model this behavior by increasing the interest weight $v(x)$ for these particular people, which would indicate the data owner's interest in providing higher protection for these people. The data owner can further model the adversary's intentions by tuning the error reduction function E_j^*, to capture whether an adversary is interested in exact disclosures or partial, and what kind of partial, disclosures. It should be noted that it can be possible that an adversary's interests and the interests of a mining user are rather similar, which naturally results in the anti-utility being similar to the good utility. We avoid this case by assuming that the field i that is of interest to a mining user is different than the field j that is of interest to the adversary. We now introduce the definition of a $(\mathcal{U}_{\mathsf{bad}}^{(\mathsf{avg})}, \mathcal{U}_{\mathsf{bad}}^{(\mathsf{worst})}, \mathcal{M}^{j(\mathsf{worst})})$-privacy-loosing sanitization mechanism, again for the field j.

The reference set of miners $M = \{\mathcal{M}_1^j, \ldots, \mathcal{M}_r^j\}$, consists of data miners that predict the field indexed by j and can be used to obtain privacy-implicating predictions. In practice, the set M consists of a collection of data mining algorithms (under different parameters and including specific pre-processing methods, such as sub-sampling or discretization) that an adversary is likely to use to breach privacy. The design and maintenance of this reference set is a challenging task. We discuss it in Sect. 4.4. Nevertheless, the set M should contain optimal algorithms that cover large classes of attacks.

The definition of $\mathcal{U}_{\mathsf{bad}}$ and the reference set of miners M allows us to define privacy through two anti-utility measures. Naturally, an adversary does not know which miner's results have the highest utility, and so we are performing a worst-case analysis. The value $\mathcal{U}_{\mathsf{bad}}^{(\mathsf{worst})}$ represents the maximum anti-utility that an adversary can obtain. Since an adversary may not know the miner $\mathcal{M}^{j(\mathsf{worst})}$ that attains this maximum utility, the adversary may use all the miners, which results in an average-case situation, and the average utility that can be obtained using all the miners in M is then denoted by $\mathcal{U}_{\mathsf{bad}}^{(\mathsf{avg})}$.

A sanitization mechanism $\mathcal{S}$ operating on a database DB is said to be $(\mathcal{U}_{\mathsf{bad}}^{(\mathsf{avg})}, \mathcal{U}_{\mathsf{bad}}^{(\mathsf{worst})}, \mathcal{M}^{j(\mathsf{worst})})$-*privacy-loosing*, where

$$\mathcal{U}_{\mathsf{bad}}^{(\mathsf{avg})}(DB, \mathcal{S}, M) = \frac{1}{r}\sum_{\ell=1}^{r} \mathcal{U}_{\mathsf{bad}}(DB, \mathcal{S}, \mathcal{M}_\ell^j) \text{ and}$$

$$\mathcal{U}_{\mathsf{bad}}^{(\mathsf{worst})}(DB, \mathcal{S}, M) = \max_{\ell=1,\ldots,r} \mathcal{U}_{\mathsf{bad}}(DB, \mathcal{S}, \mathcal{M}_\ell^j) \ ,$$

and where $\mathcal{M}^{j(\mathsf{worst})}$ is the miner that attains the highest anti-utility $\mathcal{U}_{\mathsf{bad}}^{(\mathsf{worst})}$.

Note that our privacy definition does not improve on the privacy guarantees of a given sanitization mechanism, but rather it captures the privacy ideas and guarantees of a sanitization mechanism in a uniform and quantitative way. In particular, we are measuring the utility of successful predictions that are leading to disclosures and breaches of privacy guarantees. Using this uniform way in measuring privacy allows us (and the data owners) to compare different sanitization methods using the same metrics and methodology.

One natural instantiation of the error reduction function is using a "nearness" of the prediction toward the value in the database as follows: A prediction $\widehat{\mathcal{S}(x)}_j = \mathcal{M}_j(\mathcal{S}(DB), \mathcal{S}(x))$ is c%-*nearer* to the original value x_j than the sanitized value $\mathcal{S}(x)_j$ if

$$\Delta_j(\widehat{\mathcal{S}(x)}_j, x_j) \leq \frac{100 - c}{100} \cdot \Delta_j(\mathcal{S}(x)_j, x_j) \ ,$$

where Δ_j is a distance function and $c \in (0, 100)$. We define a 100%-nearer prediction as the *exact* prediction, that is, $\widehat{\mathcal{S}(x)}_j = x_j$. A 0%-nearer prediction, or just a *nearer prediction* is like c%-nearer prediction with $c = 0$, but with strict inequality in the definition (not less than or equal). Based on this, the error reduction function E_j^* can simply have a high value, e.g. 1, if a prediction is c%-nearer and low value, e.g. 0, otherwise.

Using the reference set of miners M, the data owner can perform the mining and obtain c%-nearer predictions for a choice of c and interest weights $v(x)$. The definition of a $(\mathcal{U}_{\mathsf{bad}}^{(\mathsf{avg})}, \mathcal{U}_{\mathsf{bad}}^{(\mathsf{worst})}, \mathcal{M}^{j(\mathsf{worst})})$-privacy-loosing sanitization mechanism then allows the data owner to compare sanitization methods and their output on the database from the privacy point of view. Clearly, the higher the anti-utility

values $\mathcal{U}_{\text{bad}}^{(\text{avg})}$ and $\mathcal{U}_{\text{bad}}^{(\text{worst})}$, the higher the risk of disclosures.

3. RELATING ADVERSARIAL GAINS TO OUR FRAMEWORK

Considering and measuring the shift in adversarial knowledge before and after the adversary sees the sanitized data has been recognized in random-perturbation sanitization methods [9]. For anonymization methods, the privacy has been traditionally measured syntactically (as the number and amount of generalizations/sanitizations). Measuring privacy semantically as an adversarial gain has been first considered for ℓ-diversity [15] and again recently by Brickell and Shmatikov [4]. We relate the adversarial model of the latter work to our framework, because it also considers the balance between utility and privacy. We note that the model is only usable for k-anonymity-like anonymization methods, while our framework allows for any sanitization method.

Following are the adversarial knowledge and accuracy gains proposed in [4] described using our notation and fitted into our framework: Suppose that a database DB contains only a single sensitive field and that its possible values are $s_1, \ldots, s_t$. Let $p(A, s)$ denote the fraction of a set of tuples A that contains the sensitive value s, that is, the probability that a randomly chosen tuple of A has the value s. By $\langle \mathcal{S}(x) \rangle$ we denote the set of tuples from $\mathcal{S}(DB)$ that have the same combination of "quasi-identifying" (QI) values. *Quasi-identifying attributes* are non-identifying attributes, but their combination can be used to identify an individual [23]. Let R denote the set of all representatives $\mathcal{S}(x)$ of these sets $\langle \mathcal{S}(x) \rangle$'s such that $\mathcal{S}(DB) = \bigcup_{\mathcal{S}(x) \in R} \langle \mathcal{S}(x) \rangle$. Finally, we denote by $\widehat{\mathcal{S}(DB)}$ the sanitized database that includes predictions obtained by $\mathcal{M}^{j\,(\text{worst})}$. Similarly, we then use the notation of $\langle \widehat{\mathcal{S}(x)} \rangle$ for the set with the same QI's and $\widehat{R}$ for the representatives.

The incremental gain in an adversary's knowledge before and after seeing the released sanitized database $\mathcal{S}(DB)$ is measured as the "adversarial knowledge gain" [4] $\mathcal{A}_{\text{know}}$ which can be represented in our framework by our anti-utility function $\mathcal{U}_{\text{bad}}$. $\mathcal{A}_{\text{know}} = \mathcal{U}_{\text{bad}}(DB, \mathcal{S}, \mathcal{M}_{\text{know}}^{sensitive})$ using the weight function $v(x) = 1$ for those x with $\mathcal{S}(x) \in R$, and $v(x) = 0$ otherwise, and using the error reduction function $E^*_{sensitive}(\hat{x}_j, \mathcal{S}(x)_j, x_j) = \hat{x}_j / |\mathcal{S}(DB)|$, where the prediction $\hat{x}_j$ is obtained using a theoretical miner $\mathcal{M}_{\text{know}}^{sensitive}(\mathcal{S}(DB), \mathcal{S}(x))$ that knows the distribution of the sensitive values in DB, and returns the value of $1/2 \sum_{i=1}^{t} |p(DB, s_i) - p(\langle \mathcal{S}(x) \rangle, s_i)|$.

We extend this model by measuring the adversarial knowledge gain over the sanitized data with predictions $\widehat{\mathcal{S}(DB)}$. In this case, $\mathcal{U}_{\text{bad}}(DB, \mathcal{S}, \mathcal{M}_{\text{know}-\text{new}}^{sensitive})$ is computed using $v(x) = 1$ for those x with $\widehat{\mathcal{S}(x)} \in \widehat{R}$ and $v(x) = 0$ otherwise, and using $E^*_{sensitive}(\hat{x}_j, \mathcal{S}(x)_j, x_j) = \hat{x}_j / |\widehat{\mathcal{S}(DB)}|$, where $\hat{x}_j = \mathcal{M}_{\text{know}-\text{new}}^{sensitive}(\mathcal{S}(DB), \mathcal{S}(x)) = 1/2 \sum_{i=1}^{t} |p(DB, s_i) - p(\langle \widehat{\mathcal{S}(x)} \rangle, s_i)|$. Similarly, it is possible to capture and extend the "adversarial accuracy gain" $\mathcal{A}_{\text{acc}}$, proposed in [4].

Our experimental results, presented in Sect. 4.4, show that the adversary always gains more after our privacy attack using data mining than before. In other words, our work can be seen as an extension of Brickell and Shmatikov's adversarial model, an extension which uses the same measures but now additionally takes data mining attacks on privacy into account. In this sense, the new privacy loss that needs to be considered is the adversarial gain computed over $\widehat{\mathcal{S}(DB)}$, also depicted in Fig. 2.

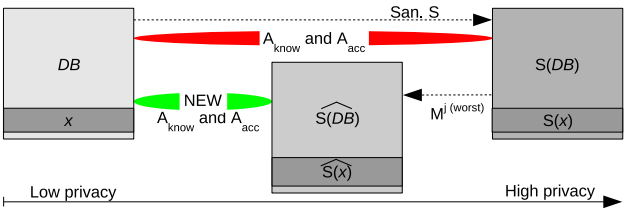


Figure 2: Extending the Brickell and Shmatikov's adversarial gain model by applying the measures to the sanitized data after including nearer predictions

In addition, we note that Brickell and Shmatikov's usefulness model is captured in our framework. Their measure of the usefulness of the sanitized data is classification accuracy, which can be easily modeled using our good utility concept and our definition of $(1-\delta, L)$-utility-preserving sanitization, as outlined in Sect. 2.3.

4. EXPERIMENTAL RESULTS

We simulate the problem of a data owner who needs to decide which sanitization method applied to his/her data provides better privacy and usefulness. We compare k-anonymity and ϵ-differential privacy. We use both methods in the non-interactive mode to produce sanitized versions of the UCI's Adult database for different values of k and ϵ. To perform an evaluation of the privacy and usefulness, we use data miners from the Weka package, which are readily available and can be used even by non-expert attackers.

Our experiments demonstrate that data mining can be successfully used to attack privacy and make privacy-impacting predictions. We consider several scenarios, where an attacker is interested in breaching privacy for (and hence the data owner is interested in protecting) selected individuals or all individuals, against partial or exact disclosures. In these scenarios, we demonstrate that it is impossible to make a general privacy statement about a sanitization method. This is because privacy depends on a database and many choices and decisions made by the data owner about the purpose of releasing the data and about the required level of protection.

We measure the usefulness of the released sanitized data in the terms of data mining utility rather than syntactically. As expected and previously shown [10, 13], we also confirm that the good utility of the sensitive attributes in the sanitized data can be preserved, that is, its decline is acceptable considering the privacy gains. Although the same data was used, this result is in contrast to [4], where the utility of non-sensitive attributes was measured.

In our experiments we do not assume any auxiliary or background-knowledge information that may be available to the adversaries. Rather, we evaluate the privacy based only on the available sanitized data. Although it does not allow us to make blanket statements about sanitization methods in general, this is a practical way of evaluating threats to privacy from a large portion of real-life adversaries.

4.1 Data

We performed our experiments on the *Adult data* from the UCI Machine Learning Repository [3] which is the de facto standard for experiments in sanitized data publishing. The data consists of 45,222 records. We prepared the data as proposed in [10]. That is, we removed the records with unknown values and kept the following attributes: age, work class, education, marital status, occupation, race, gender, native country, and salary. *Age* covers all the integers from 17 to 99. *Salary* is a binary attribute (over or below and equal to \$50K). For brevity we do not describe the other domains, but they can be found in [10].

4.2 Sanitization

We apply two distinct sanitization methods to the data: k-anonymity and ϵ-differential privacy. The former one is achieved by generalizations and/or suppressions, the later one by additive-noise perturbation. The former one's privacy guarantee is syntactic, the later one's semantic. The former one's utility is measured as the number and amount of generalizations and suppressions, but no data mining utility is considered, and the later one's utility idea is to approximate the original data distribution and use it for specific data mining tasks.

k-ANONYMIZATION. k-anonymity [23, 22, 28] is a notion that quantifies the privacy risk and specifies the requirements for publishing a database in order to limit the linking attack. The *linking attack* [23] is an attack where a priori knowledge from another source or database can be used to identify an entity in a released database.

A database to be released contains some sensitive attributes, identifying attributes, and so-called quasi-identifying attributes. *Quasi-identifying attributes* are non-identifying attributes, but their combination can be used to identify an individual in a linking attack.

A released database is said to satisfy *k-anonymity* (is *k-anonymized*) if for each existing combination of quasi-identifying attribute values in the database, there are at least $k-1$ other records in the database that contain such a combination. There are several methods to achieve k-anonymity [22, 27]. The basic techniques use hierarchical generalizations and cell suppressions. The released k-anonymized database has all the identifying attributes suppressed and contains unmodified sensitive attributes.

The privacy idea is syntactic, that is k-anonymity does not capture the shift or gain in the adversary's knowledge. The k-anonymity guarantee is that an identified individual in the database is indistinguishable from at least $k-1$ other individuals. However, k-anonymity lacks the guarantee that an identified individual can be linked to a sensitive value – the homogeneity and the background knowledge attacks [15] are still possible. There are many extensions of k-anonymity that overcome this and provide additional privacy measures, please see the Related Work section for references.

There is no explicit utility idea captured in the k-anonymity notion. In fact, suppressing all the identifying and quasi-identifying attribute values, that is, publishing just the sensitive values, satisfies the k-anonymity definition (for any $2 \leq k \leq n$, where n is the number of records). A recent result [4] shows that, in most cases, this trivial k-anonymization provides equivalent data-mining utility. Regardless, the released data is supposed to be a valuable source of information for research, statistical analysis, or data mining. The concept of *optimal* k-anonymity [20] is concerned with the achievement of a k-anonymous data release while minimizing the number of cell suppressions or number and amount of generalizations. Although these results were developed for showing the hardness of the k-anonymization process, they can be seen as sanitization mechanisms that preserve the usefulness of the released data.

To achieve k-anonymity, we decided to use a recent k-clustering algorithm [5], because it strives to minimize the number of necessary generalizations through creating clusters of at least k records while minimizing the sum of cluster diameters. Our sanitization algorithm $\mathcal{S}$ was the proposed greedy k-clustering algorithm [5, Fig. 5]. It does not suppress any cells nor records. Our quasi-identifying (QI) attributes were all the attributes except *salary*, which was considered to be a sensitive attribute.

Selecting all but the sensitive attribute as QI-attributes is common, because it provides the maximum protection against any linking attack with any subset of the QI-attributes. On the other hand, it requires more generalizations and/or suppressions during sanitization, thus possibly negatively affecting utility. Our experiments show that the utility decline is negligible.

During k-anonymization, the values of the attribute *age* were generalized into intervals of length 5 (0-4, 5-9, ...), then into length 10 (0-9, 10-19, ...), then into length 20 (0-19, 20-39, ...), then into length 50 (0-49, 50-99) and finally suppressed into length 100 (0-99). The taxonomy trees that were used for generalizations of the other attributes are omitted for brevity. We performed k-anonymization for $k = 2$, 10, 50, and 100.

ϵ-DIFFERENTIAL PRIVACY. Perturbation methods mask information and relations by adding noise to the released values. The question of how much noise is necessary to achieve privacy and utility was considered by Dwork et al. [8, 7] in their notion called "differential privacy": A randomized function $\mathcal{S}$ gives *ϵ-differential privacy* if for all data sets D_1 and D_2 differing on at most one element, and all $R \subseteq Range(\mathcal{S})$, $\Pr[\mathcal{S}(D_1) \in R] \leq \exp(\epsilon) \times \Pr[\mathcal{S}(D_2) \in R]$ ϵ-differential privacy is achieved by adding random exponential (Laplace) noise to released numeric values.

The privacy guarantee of the ϵ-differential privacy is semantic, that is, it considers the shifts and incremental gain in the adversary's knowledge. The ϵ-differential privacy notion limits the probability that the randomized function $\mathcal{S}$, a sanitization function, would leak information from a data set that is extended by at most one element.

The utility of the data released by an ϵ-differential privacy mechanism is not well-defined. For example, consider a randomized function $\mathcal{S}$ that on every input always outputs the same constant. Such a function satisfies the ϵ-differential privacy notion, as well as the similar ϵ-indistinguishability notion [8], for every leakage $\epsilon \geq 0$. Yet the output of this function $\mathcal{S}$ is useless for practical purposes, such as statistical analysis and knowledge discovery.

In our experiments, we used leakage $\epsilon = 0.5$, 0.1, 0.05, and 0.01. Then the ϵ-differential privacy for the numerical attributes was achieved by a sanitization mechanism $\mathcal{S}$ that added to the original numerical values noise chosen randomly from the Laplace distribution $\text{Lap}(0, \Delta f/\epsilon)$ with the probability density function $h(y) = (\epsilon/2) \cdot \exp(-\epsilon|y|/\Delta f)$, where Δf is the sensitivity [7] of the query function f, and was $\Delta f = 82$ for the attribute *age* (range 17-99) and

Miner $\mathcal{M}$	$\mathcal{U}_{\text{good}}^{(\text{orig})}$	$\mathcal{U}_{\text{good}}^{(\text{san})}$ of k-anonymized data				$\mathcal{U}_{\text{good}}^{(\text{san})}$ of ϵ-diff.privacy data			
		$k=2$	$k=10$	$k=50$	$k=100$	$\epsilon=0.5$	$\epsilon=0.1$	$\epsilon=0.05$	$\epsilon=0.01$
NaiveBayes	36,972	36,642	36,285	35,900	35,242	36,279	36,356	36,350	36,320
J48 (C4.5)	37,609	37,143	36,909	36,711	36,298	37,293	37,222	37,221	37,221
RandomForest	36,172	36,069	36,707	36,740	36,321	36,012	35,969	35,932	35,962
Utl. decline δ	N/A	1.24%	1.86%	2.90%	4.68%	1.87%	1.67%	1.68%	1.76%

Table 1: Good utility and the highest utility decline for the three Weka miners

$\Delta f = 16$ for the attribute *education* (range 1-16).

4.3 Mining for Good Utility

We performed a classification of the binary attribute *salary*. Our miners were the RandomForest, NaiveBayes, and J48 (C4.5 decision tree) classifiers with the default settings from the Weka 3.6.0 package [32]. This approach, data, classifiers, and k-anonymization are the same as in the previous works in the field [15, 14, 4]. In addition, we also considered another sanitization – the ϵ-differential privacy – and evaluated the perturbed data with the three mentioned classifiers. Although it is impossible to know in advance what the real mining users of the sanitized data want to obtain, we measure the usefulness of the sanitized data using these three data mining algorithms under the assumption that using miners based on similar concepts (Bayesian or tree methods) would provide a comparable usefulness.

For all three miners, we considered the classification accuracy. We used the Weka's default 10-fold cross-validation to evaluate the accuracy and compute the utility. Our utility functions were $\mathcal{U}_{\text{good}}^{(\text{orig})}$ and $\mathcal{U}_{\text{good}}^{(\text{san})}$ with weight function $w(x) = 1$ for all $x \in DB$ and the following error implication function that simply captured the classification accuracy: $E_{salary}(\hat{x}_{salary}, \bar{x}_{salary}) = 1$ if $\hat{x}_{salary} = \bar{x}_{salary}$ and 0 otherwise.

The utility obtained from mining over the original unsanitized data as well as over the sanitized data is in Table 1, together with the decline of the utility that was the highest among the three Weka miners.

Discussion. For privacy reasons, the QI-attribute set was selected to be maximal, which consequently resulted in more generalizations compared to the case when the set of the QI-attributes would be smaller. Even with the higher number of generalizations, the utility of k-anonymized data declined at most by 4.86%, which can be seen as an acceptable decline considering the privacy gains. The results also show that the higher the parameter k (that is, the stronger privacy guarantee is requested), the higher the utility decline. This supports the hypothesis that there is a trade-off between privacy and utility. In some cases, the utility of mining over the sanitized data is slightly higher than the utility of mining over the original data. This is likely due to the fact that k-anonymity's generalization smooths the data and hence removes some outliers. Overall, k-anonymity achieved through the greedy k-clustering algorithm on the Adult data is ($\delta = 4.68$, $L = \{\text{NaiveBayes, J48, RandomForest}\}$)-utility-preserving.

The utility decline for the ϵ-differential privacy perturbed data is at most 1.87%. The lower utility decline compared to k-anonymity is the result of the fact that only two attributes, *age* and *education*, are numerical and were perturbed. This is an inherent consequence of a comparison of different sanitization methods over the same data. There is no trend supporting the hypothesis of the trade-off between privacy and utility, as it was in the case of k-anonymity. Overall, ϵ-differential privacy achieved by noise-addition with noise from a Laplace distribution is ($\delta = 1.87$, $L = \{\text{NaiveBayes, J48, RandomForest}\}$)-utility-preserving.

From the utility point of view, when using the three Weka miners, the ϵ-differential privacy is a better alternative than k-anonymity, because for the same miners it has a lower utility decline. We have measured only the classification accuracy and have not exploited all the possibilities of our utility definition. In different scenarios, such as the medical scenario briefly described in Sect. 2.3, using different data, or using a different set of miners, it may turn out that the declines are different and that k-anonymity is better than ϵ-differential privacy.

4.4 Mining for Anti-Utility to Measure Privacy

We were interested to see if we can predict a nearer value of the attribute *age*, where the nearness depends on our scenarios.

Reference set of miners. For predicting categorical values, such as *age* that has been generalized into intervals, we selected some of the miners available in the Weka 3.6.0 package [32]. Our reference set $M_{categorical}$ of miners used to try to predict nearer categorical values consisted of the following 32 miners: bayes (AODE, AODEsr, BayesNet, HNB, NaiveBayes, NaiveBayesSimple, NaiveBayesUpdateable, WAODE), functions (RBFNetwork, SMO), lazy (IB1, IBk, KStar, LWL), rules (ConjunctiveRule, DTNB, DecisionTable, OneR, Ridor, ZeroR), trees (DecisionStump, Id3, J48, J48graft, REPTree, RandomForest, RandomTree, SimpleCart), misc (HyperPipes, MinMaxExtension, OLM, VFI). Our reference set $M_{numerical}$ of miners, used to try to predict nearer numerical values, consisted of the following 10 miners from the Weka 3.6.0 package [32]: functions (RBFNetwork), rules (ConjunctiveRule, DecisionTable, M5Rules, ZeroR), trees (DecisionStump, M5P, REPTree), functions (LeastMedSq, LinearRegression). All the miners that we selected were used with their default options, but the memory available to the Java virtual machine was increased to 2 GB. Although only default settings and no optimizations were used, the experiments clearly show higher adversarial gains than previously thought and therefore the importance of considering data mining as a tool for privacy evaluation.

In practice, the selection of miners is a choice of the data owner. We elected to choose miners from the Weka package because of their availability and ease of use, and therefore a likelihood that an adversary would do the same. However, the selection of the reference miners can be influenced by the type of data and the organization that collects them. In

	San. $\mathcal{S}$:	k-anonymity				ϵ-differential privacy			
		$k=2$	$k=10$	$k=50$	$k=100$	$\epsilon=0.5$	$\epsilon=0.1$	$\epsilon=0.05$	$\epsilon=0.01$
A:	$\mathcal{U}_{\mathsf{bad}}^{(\mathsf{avg})}$	316.6	135.7	22.2	4.4	43487.7	43832.0	43425.8	43430.8
	$\mathcal{U}_{\mathsf{bad}}^{(\mathsf{worst})}$	630	322	185	77	44383	44729	44741	44729
	$\mathcal{M}_{age}^{(\mathsf{worst})}$	LWL	OLM	VFI	OLM	ZeroR	ZeroR	RBNF.	ZeroR
B:	$\mathcal{U}_{\mathsf{bad}}^{(\mathsf{avg})}$	0	0	0	0	1046.5	598.67	293.83	104.5
	$\mathcal{U}_{\mathsf{bad}}^{(\mathsf{worst})}$	0	0	0	0	1341	943	780	608
	$\mathcal{M}_{age}^{(\mathsf{worst})}$	N/A	N/A	N/A	N/A	RBNF.	Dec.Stump	Dec.Stump	Dec.Table
C:	$\mathcal{U}_{\mathsf{bad}}^{(\mathsf{avg})}$	192.6	90.3	16.0	3.2	2200.8	2193.5	2159.2	2159.3
	$\mathcal{U}_{\mathsf{bad}}^{(\mathsf{worst})}$	389	199	121	57	2242	2261	2261	2261
	$\mathcal{M}_{age}^{(\mathsf{worst})}$	LWL	OLM	VFI	OLM	ZeroR	ZeroR	ZeroR	ZeroR

Table 2: Anti-utility and the miner that attains the maximum anti-utility for three scenarios A (all nearer predictions), B (exact predictions), and C (protection of selected 5% people)

any case, the reference set of miners should be updated to reflect the progress and advances in data mining technology, and to keep up with the possible adversaries. Of course, using more miners, refined parameters, pre-processing (such as sub-sampling and/or discretization), and predicting over more attributes than just *age* can lead to additional privacy loss. It is impossible to avoid all privacy attacks and adversaries, and in this sense, evaluating privacy using such a reference set of miners estimates the threat to privacy based on the adversaries that are likely to use these or similar miners. For instance, using available and easy-to-use miners such as the ones from Weka would measure the threat to privacy coming from the equivalent to so-called script-kiddies – adversaries that use ready-to-use tools and scripts.

Scenarios. We used the following three scenarios that an data owner may choose to model an adversary's intentions: The data owner is interested in protecting (A) all the individuals equally against any partial disclosure, that is, against any nearer prediction; (B) all the individuals equally against exact disclosures, that is, against 100%-nearer predictions; and (C) a selected 2261 (5%) individuals, that is, the owner chooses the weight $v(x)=1$ for them and $v(x)=0$ for the rest of the population. These scenarios are modeled using the interest weight function $v(x)$, using the nearness concept as the error reduction function E^*_{age}, and the parameter c for c%-nearer predictions. The distance function Δ_{age} used in the "nearness" concept was the Euclidean distance function and a function computing the length of intervals, respectively, for numbers and intervals. The error reduction function E^*_{age} was set to 1 if a prediction was c%-nearer and 0 otherwise.

Table 2 summarizes the obtained results of our data mining efforts to attack privacy of k-anonymized and ϵ-differential privacy perturbed data for these three scenarios. The table also represents the classification of k-anonymity achieved through the greedy k-clustering algorithm and ϵ-differential privacy achieved by noise-addition with noise from a Laplace distribution on the Adult data using our definition of a $(\mathcal{U}_{\mathsf{bad}}^{(\mathsf{avg})}, \mathcal{U}_{\mathsf{bad}}^{(\mathsf{worst})}, \mathcal{M}^{j(\mathsf{worst})})$-privacy-loosing sanitization mechanism.

Discussion. In all three scenarios, the k-anonymity is a better sanitization method for releasing the Adult database from the privacy point of view, because the anti-utility is lower than it is for ϵ-differential privacy. k-anonymity exhibits an expected trend – as k is increased (more privacy), the anti-utility decreases (less risk of privacy breach). The same trend is observable for ϵ-differential privacy in scenario B, proving that higher leakage ϵ (more privacy) results in fewer exact disclosures. This trend is not obvious for scenarios A and C in the case of ϵ-differential privacy – roughly the same anti-utility is obtained regardless of the leakage ϵ. This means that noise can be reduced (nearer prediction obtained) for approximately the same population independently of ϵ. Finally, the inability to obtain exact disclosures for k-anonymity in scenario B was expected, as the original age values were generalized into intervals, and miners were only able to learn and predict shorter intervals but not particular values.

In summary, k-anonymity is a better sanitization than ϵ-differential privacy for the Adult database, for all three scenarios, and the data owner selected parameters.

Comparison. Table 3 shows that the previously proposed adversarial gain model [4], introduced in Sect. 3, does not capture the attack on privacy using data mining – for all k's, the originally proposed adversarial accuracy gain as well as the adversarial knowledge gain computed over the released sanitized data $\mathcal{S}(DB)$ were lower than the gains computed over the sanitized data containing nearer predictions $\widehat{\mathcal{S}(DB)}$ that were obtained using $\mathcal{M}^{age(\mathsf{worst})}$. The original adversarial gains cannot be used for measurements over the ϵ-differential privacy perturbed data, only over k-anonymized data.

San. $\mathcal{S}$:	$k=2$	$k=10$	$k=50$	$k=100$
$\mathcal{A}_{\mathsf{acc}}$ of $\mathcal{S}(DB)$	0.1084	0.0861	0.0658	0.0541
$\mathcal{A}_{\mathsf{acc}}$ of $\widehat{\mathcal{S}(DB)}$	0.1124	0.0898	0.0667	0.0546
$\mathcal{A}_{\mathsf{know}}$ of $\mathcal{S}(DB)$	0.2763	0.2329	0.2053	0.1913
$\mathcal{A}_{\mathsf{know}}$ of $\widehat{\mathcal{S}(DB)}$	0.2775	0.2357	0.2063	0.1919

Table 3: Adversarial gains of the released k-anonymized data $\mathcal{S}(DB)$ and of the data $\widehat{\mathcal{S}(DB)}$ that in addition includes all the nearer predictions obtained by $\mathcal{M}^{age(\mathsf{worst})}$

5. CONCLUSIONS

We proposed a pragmatic framework with the ability to evaluate and compare sanitization methods using data mining utility. It provides decision support for a data owner

to help decide which sanitization method is the best for a given database based on the end user needs and an adversary's intentions (that the owner wants to defend against). Our experimental results demonstrated that the previously proposed adversarial gains over the sanitized data with predictions are higher than over the sanitized data without predictions.

We focused on motivating and defining the framework, and providing practical evaluation. Several extensions of our work are possible, as indicated throughout the paper.

An important question remains whether the proposed framework and measures can be used in designing a versatile sanitization method that considers both privacy and utility and further takes into account the needs of the end users and requirements of the data owners.

6. REFERENCES

[1] N. A. Adam and J. C. Wortman. Security-control methods for statistical databases. *ACM Comput Surv*, 21(4):515–556, 1989.

[2] R. Agrawal and R. Srikant. Privacy-Preserving Data Mining. In *SIGMOD*, pages 439–450, 2000.

[3] A. Asuncion and D. Newman. UCI Machine Learning Repository, 2007.

[4] J. Brickell and V. Shmatikov. The Cost of Privacy: Destruction of Data-Mining Utility in Anonymized Data Publishing. In *KDD*, pages 70–78, 2008.

[5] J.-W. Byun, A. Kamra, E. Bertino, and N. Li. Efficient k -Anonymization Using Clustering Techniques. In *DASFAA*, pages 188–200, 2007.

[6] V. Ciriani, S. D. C. di Vimercati, S. Foresti, and P. Samarati. k-Anonymity. In *Secure Data Management in Decentralized Systems*. Springer, 2007.

[7] C. Dwork. Differential Privacy. In *ICALP*, pages 1–12, 2006.

[8] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In *TCC*, pages 265–284, 2006.

[9] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *PODS*, pages 211–222, 2003.

[10] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *KDD*, pages 279–288, 2002.

[11] M. Kantarcioglu and C. Clifton. Privacy-preserving Distributed Mining of Association Rules on Horizontally Partitioned Data. *IEEE T Knowl Data En*, 16(9):1026–1037, 2004.

[12] D. Kifer and J. Gehrke. Injecting utility into anonymized datasets. In *SIGMOD*, pages 217–228, 2006.

[13] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Workload-aware anonymization. In *KDD*, pages 277–286, 2006.

[14] N. Li, T. Li, and S. Venkatasubramanian. t-Closeness: Privacy Beyond k-Anonymity and ℓ-Diversity. In *ICDE*, pages 106–115, 2007.

[15] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. In *ICDE*, pages 24–35, 2006.

[16] D. J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Y. Halpern. Worst-Case Background Knowledge for Privacy-Preserving Data Publishing. In *ICDE*, pages 126–135, 2007.

[17] G. Miklau and D. Suciu. A Formal Analysis of Information Disclosure in Data Exchange. In *SIGMOD*, pages 575–586, 2004.

[18] M. E. Nergiz and C. Clifton. Thoughts on k-Anonymization. In *PDM*, page 96, 2006.

[19] M. E. Nergiz, C. Clifton, and A. E. Nergiz. MultiRelational k-Anonymity. In *ICDE*, pages 1417–1421, 2007.

[20] H. Park and K. Shim. Approximate algorithms for K-anonymity. In *SIGMOD*, pages 67–78, 2007.

[21] V. Rastogi, S. Hong, and D. Suciu. The Boundary Between Privacy and Utility in Data Publishing. In *VLDB*, pages 531–542, 2007.

[22] P. Samarati. Protecting Respondents' Identities in Microdata Release. *IEEE T Knowl Data En*, 13(6):1010–1027, 2001.

[23] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical Report SRI-CSL-98-04, SRI Computer Science Laboratory, 1998.

[24] M. Sramka, R. Safavi-Naini, and J. Denzinger. An Attack on the Privacy of Sanitized Data That Fuses the Outputs of Multiple Data Miners. In *PADM*, pages 130–137, 2009.

[25] M. Sramka, R. Safavi-Naini, J. Denzinger, M. Askari, and J. Gao. Utility of Knowledge Discovered from Sanitized Data. Technical Report 2008-910-23, University of Calgary, 2008.

[26] M. Sramka, R. Safavi-Naini, J. Denzinger, M. Askari, and J. Gao. Utility of Knowledge Extracted from Unsanitized Data when Applied to Sanitized Data. In *PST*, pages 227–231, 2008.

[27] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *Int J Uncertainty, Fuzziness and Knowl-based Syst*, 10:571–588, 2002.

[28] L. Sweeney. k-anonymity: a model for protecting privacy. *Int J Uncertainty, Fuzziness and Knowl-based Syst*, 10(5):557–570, 2002.

[29] T. M. Truta and B. Vinay. Privacy Protection: p-Sensitive k-Anonymity Property. In *PDM*, pages 94–103, 2006.

[30] J. S. Vaidya and C. Clifton. Privacy preserving association rule mining in vertically partitioned data. In *KDD*, pages 639–644, 2002.

[31] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis. State-of-the-art in privacy preserving data mining. *SIGMOD Record*, 33(1), 2004.

[32] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[33] R. C.-W. Wong, J. Li, A. W.-C. Fu, and K. Wang. (α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing. In *KDD*, pages 754–759, 2006.