DT-Anon: Decision Tree Target-Driven Anonymization

Sabrina De Capitani di Vimercati[0000−0003−0793−3551], Sara Foresti[0000−0002−1658−6734], Valerio Ghirimoldi, and Pierangela Samarati[0000−0001−7395−4620]

Computer Science Department, Università degli Studi di Milano
{sabrina.decapitani,sara.foresti,pierangela.samarati}@unimi.it;
valerio.ghirimoldi@studenti.unimi.it

Abstract. More and more scenarios rely today on data analysis of massive amount of data, possibly contributed from multiple parties (data controllers). Data may, however, contain information that is sensitive or that should be protected (e.g., since it exposes identities of the data subjects) and cannot simply be freely shared and used for analysis. Business rules, restrictions from individuals (data subjects to which data refer), as well as privacy regulations demand data to be sanitized before being released or shared with others. Unfortunately, such protection typically comes with a loss of utility of the released data, impacting the performance of the analytics tasks to be executed.
In this paper, we present DT-Anon, a target-driven anonymization approach that aims at protecting (anonymizing) data while preserving as much as possible the capability of a classification task operating downstream to learn from the anonymized data. The basic idea of our approach is to perform the anonymization process on partitions produced by a decision tree driven by the target of the classification task. Each partition is then independently anonymized, to limit the impact of anonymization on the attributes and values that work as predictors for the target of the classification task. Our experimental evaluation confirms the effectiveness of the approach.

Keywords: data anonymization · machine learning classifier · target-driven anonymization · decision tree

1 Introduction

Today's society is highly dependent on data, with huge (and ever increasing) amount of data generated, collected, and processed. Concepts such as big data, data analytics, and machine learning are today common terms for the layperson, witnessing their pervasiveness in every context of our daily life. As a matter of fact, the availability of massive amount of data, together with powerful and efficient computational infrastructures and services, and hence ability to extract knowledge from data, is at the heart of our smart society, bringing great benefits in different domains, from business to leisure.

More and more scenarios rely today on different parties contributing to the collection, sharing, and analysis of data. However, often data collections include information that cannot be freely shared (e.g., [2]). This is, for example, the case of data referred to individuals, whose privacy needs to be protected, as demanded by data regulations. The EU General Data Protection Regulation (GDPR), the California Consumer Protection Act (CCPA), and other similar regulations worldwide, demand protection for information referred to individuals (data subjects) whose identity and sensitive information should be properly protected by data anonymization solutions (with anonymized data being exempt from the obligations set out in the regulations).

Notwithstanding the great benefit of operating on data, it is therefore of utmost importance to ensure that the privacy of data subjects (to whom the data refer) be properly protected in such data sharing and analytics process. When data analysis is performed by external parties (other than the data controller responsible for the data), this implies that data should be properly anonymized before being shared with these parties. Unfortunately, anonymization, which by design causes information loss (for a privacy gain) in the data, can have a significant negative impact on the performance of the downstream data analytics task, with the known tension between privacy and utility.

In this paper, we consider a scenario characterized by multiple parties (data controllers) contributing data for a data analytics task. Our goal is to design a *target-driven anonymization* approach, that is, a data anonymization approach aware of the data analytics task operating downstream, and driven by it. We consider classification as data analytics task, and present a technique that both protects the privacy of data subjects and maintains the utility of the anonymized data with respect to the classification task to which data are fed. Figure 1 shows our reference scenario, with data controllers anonymizing their data before providing them for the global knowledge base on which the classification task operates. Our data anonymization approach, called DT-Anon, anonymizes data aware of the data classification task to which data are contributed, with the goal of minimizing the effect of protection on the data analysis to be executed.

The remainder of the paper is organized as follows. Section 2 illustrates the basic concepts of our approach. Section 3 describes the problem addressed and the rationale of our approach. Section 4 describes our target-driven anonymization. Section 5 illustrates our experimental evaluation. Section 6 discusses related work. Finally, Section 7 presents our conclusions. Appendixes A and B report a theorem on the approach and the DT-Anon pseudocode, respectively.

2 Basic Concepts

We consider anonymization of datasets to be contributed to a data analytics task. Wishing to operate with truthful information for data analytics [1], we assume anonymization to be carried out according to k-anonymity [4, 12] enhanced with ℓ-diversity [11]. We assume datasets to be anonymized to be relational ta-

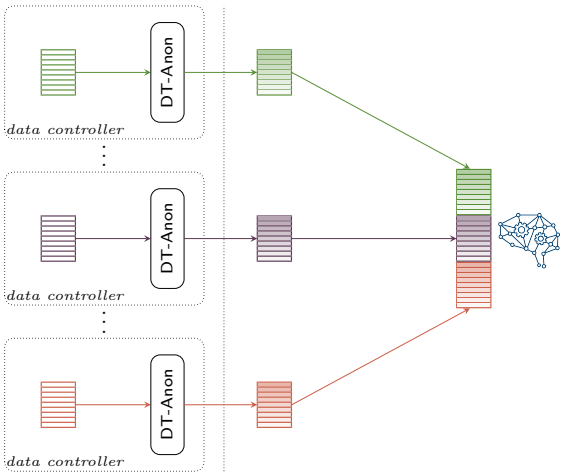


Fig. 1: Reference scenario

bles, where each relation R is characterized by a set $\{a_1, \ldots, a_n\}$ of attributes comprising:

- *identifiers*: attributes identifying data subjects, that is, entities to which data refer (we consider these attributes to be removed before release and therefore discard them from our treatment);
- *quasi-identifier*: set of attributes that jointly can, through linking with other sources, possibly reduce the uncertainty about identities of data subjects;
- *sensitive*: attribute whose values, in association with (the identity of) data subjects, are considered sensitive and should therefore be protected.

Data anonymization through k-anonymity and ℓ-diversity implies generalizing values of the quasi-identifier attributes to ensure each combination of (generalized) values of quasi-identifying attributes appearing in the table to occur at least k times (k-anonymity), and each group of tuples with the same generalized quasi-identifying values to have at least ℓ well-represented sensitive values (ℓ-diversity). We consider generalization applied at the level of cell (in contrast to the whole attribute column), hence operating at finest possible grain to limit information loss [5]. We represent the generalized value for a set of values as the interval between the minimum and maximum values in the set for continuous (i.e., numerical) attributes, and as a set comprising all the values for categorical attributes. Also, for simplicity, in the following examples, we assume the well-represented criterion of ℓ-diversity to be enforced by requiring each group to include at least ℓ different values ($\ell = 1$ implies requiring only k-anonymity to hold with no restriction on the occurrences of the sensitive values). We refer to a transformed (generalized) version of a relation satisfying k-anonymity

	Age	State	Job	Income
t_1	51	MN	Gov	150
t_2	30	CA	Non-gov	150
t_3	35	CA	Non-gov	120
t_4	35	TX	Gov	300
t_5	62	MN	Gov	200
t_6	40	TX	Gov	300
t_7	62	CA	Gov	300
t_8	37	MN	Non-gov	150
t_9	36	TX	Gov	180
t_{10}	24	MN	Gov	100
t_{11}	64	CA	Gov	250

(a)

	Age	State	Job	Income
t_2	[30-35]	CA	Non-gov	150
t_3	[30-35]	CA	Non-gov	120
t_7	[62-64]	CA	Gov	300
t_{11}	[62-64]	CA	Gov	250
t_1	[51-62]	MN	Gov	150
t_5	[51-62]	MN	Gov	200
t_8	[24-37]	MN	Non-gov	150
t_{10}	[24-37]	MN	Gov	100
t_4	[35-40]	TX	Gov	300
t_6	[35-40]	TX	Gov	300
t_9	[35-40]	TX	Gov	180

(b)

	Age	State	Job	Income
t_1	[37-62]	MN	Gov	150
t_5	[37-62]	MN	Gov	200
t_8	[37-62]	MN	Non-gov	150
t_2	[24-35]	{CA,MN}	Non-gov	150
t_3	[24-35]	{CA,MN}	Non-gov	120
t_{10}	[24-35]	{CA,MN}	Gov	100
t_4	[35-36]	TX	Gov	300
t_9	[35-36]	TX	Gov	180
t_6	[40-64]	{CA,TX}	Gov	300
t_7	[40-64]	{CA,TX}	Gov	300
t_{11}	[40-64]	{CA,TX}	Gov	250

(c)

Fig. 2: An example of original relation (a) and of two $(2, 2)$-anonymous versions of the original relation (b)-(c)

and ℓ-diversity as a (k, ℓ)-anonymous version of the relation. Figure 2 illustrates an example of original dataset and two possible $(2, 2)$-anonymous versions of it, considering `Age` and `State` to work as quasi-identifier and attribute `Income` to be sensitive.

3 Problem Definition and Sketch of the Approach

Our reference scenario is characterized by multiple *data controllers* contributing data for a data analytics task, operating on the collective information contributed by the different controllers. As data analytics task we consider *classification*. The goal of classification is to lear from classified data a model (classifier) able to predict the class (i.e., the value of a target attribute) associated with unseen data. Intuitively, classification learns dependencies of a given attribute (*target*) from other attributes (*predictors*) in the dataset. Datasets contributed by data controllers collectively represent the training data on which the classification task learns. We therefore consider a dataset $R(a_1, \ldots, a_n)$ to be released by a data controller contributing to the classification task to include, besides the quasi-identifier and sensitive attributes, also a

- *target* attribute, denoted τ, of interest for the classification task for which data are released.

Note that the target attribute cannot coincide with the sensitive attribute, by the definition of the problem at hand (as we aim at maintaining as much as possible the correct prediction of the target attribute while protecting instead inferences on the sensitive attribute).

Data may contain identifying, quasi-identifying, or sensitive information, and therefore should be anonymized before being released to external parties. As said, we assume anonymization with k-anonymity and ℓ-diversity, and hence the release to the classification analytics task of (k, ℓ)-anonymous datasets. Anonymization is enforced independently by each data controller, which could even operate with different values of k and ℓ, depending on the degree of protection wished.

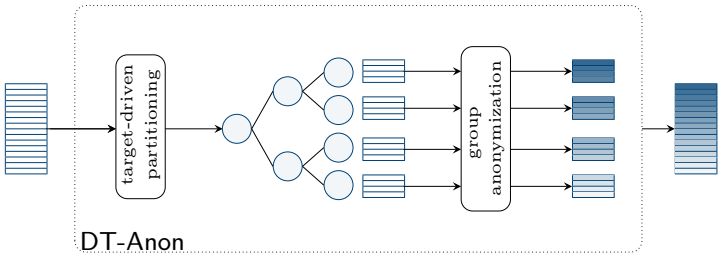


Fig. 3: Overall working of our target-driven anonymization

The problem then becomes the possible negative impact of the anonymization on the ability of the classification task to learn dependencies relevant for the classification, that is, dependencies between attributes that represent good predictors of the target and the target. Intuitively, if predictor attributes are generalized in the anonymization, the classification will be operating with less information (suffering the information loss caused by generalization). Since different anonymous versions of a table can exist, corresponding to different groups of generalized tuples and/or generalization of different attributes/values in the quasi-identifier, our goal is the definition of an anonymization process aware of the data analytics task downstream and driven by it.

Basically, the problem we address is: *Given a relational table* $R(a_1, \ldots, a_n)$ *where the set* $\{a_1, \ldots, a_n\}$ *of attributes includes quasi-identifier attributes* QI*, a sensitive attribute* s*, and a target attribute* τ*, compute a* (k, ℓ)*-anonymous version of* R *that performs well for a classification task with target* τ*.*

In other words, we aim for an anonymization that preserves as much as possible the correlation among quasi-identifier and target values. With generalization as the technique for achieving anonymization, this implies to aim for a generalized version of the dataset that maintains as specific as possible the values of the predictor attributes in the quasi-identifier on which the target values depend more, generalizing instead values of other attributes from which the target attribute is less (or no) dependent. We do so by applying the anonymization process on subsets of the dataset, where each subset groups tuples that are equal or most similar with respect to predictor attribute values (so that generalization does not affect them or affect them with limited information loss).

Our approach, called DT-Anon, comprises two steps (Figure 3):

– *target-driven partitioning* operates on the datasets by partitioning data producing groups driven by the target of the classification task. More precisely, groups are defined through a decision tree guided by the classification target. Intuitively, the decision tree partitions the tuples producing groups that will have equal or close/similar predictor values; hence avoiding their generalization or limiting the effect of a possible generalization on them.

– *group anonymization* applied on each group of tuples produced in the previous step (leaf nodes of the decision tree) with a classical anonymization approach.

Since the target-driven anonymization is enforced independently by each data controller before releasing data, in the following we illustrate our approach with reference to a single dataset. We will consider the presence of different of such anonymized datasets used for the same data analytics task in the experimental evaluation (see Section 5).

Example 1 (Running example). As running example, we consider the table in Figure 2(a), where the pair ⟨`Age`,`State`⟩ is the quasi-identifier, `Income` is the sensitive attribute, and `Job` is the target for the classification task to which data are to be contributed (the example omits identifiers and other attributes since they are not relevant for the work). We will also refer to the two $(2, 2)$-anonymous versions of the table reported in Figures 2(b)-(c).

4 Target-Driven Anonymization

We describe in more details the two phases of our target-driven anonymization.

4.1 Target-Driven Partitioning

The first step of our approach is the partitioning of tuples to produce groups of tuples that - when generalized - best preserve the correlation between (generalized) quasi-identifier and target values. Intuitively, this corresponds to produce groups that maintain in the same group tuples with the same (or as close as possible) values for predictor attributes in the quasi-identifier, so that the anonymization (generalization in particular) would not affect, or has limited impact, on them. Of course, predictors are not known, but should be learned from the data themselves.

Our approach to such target-driven partitioning is to use a machine learning algorithm based on a *decision tree* to make predictions. In other words, we create a model that predicts the value of the target attribute by learning *decision rules* from the other (quasi-identifier) attributes.

The construction of a decision tree starts from the root node that represents the whole dataset. The decision tree is then recursively built by splitting the dataset represented by a node into subsets that are represented as its child nodes. More precisely, for each node of the tree, a set of possible split values is identified for each attribute. The algorithm for the construction of the decision tree selects the attribute and the split value(s) that are most significant with respect to a specific criterion defined on the target [7] (e.g., information gain). This process terminates when a stopping condition is satisfied (e.g., the values of the target attribute for the tuples in the leaf nodes are sufficiently uniform or all the attributes have been used for splitting). By construction, the attributes used in the splitting operations are those on which the target attribute depends

more since they permit to partition the tuples in groups that are as similar as possible with respect to the target attribute. Furthermore, the tuples in these groups are also similar with respect to these attributes since they all satisfy the same decision rules (i.e., the if-then rules defined over the attributes used in the splitting operation).

We adapt this classical construction of a decision tree to our goal. First, we only use the quasi-identifier attributes for the splitting operations. Other attributes, including the sensitive attribute, are not considered. The rationale for using the quasi-identifiers only is that, as previously mentioned, the construction of the decision tree identifies the (quasi-identifier) attributes on which the target depends more and tuples with similar values for these attributes are grouped together, which is exactly what we need. The reason for not using the sensitive attribute (e.g., `Income` for our running example) is the need to ensure diversity of its values in each generalized group. The reason for not considering other attributes is that they are not affected by generalization and their values therefore are never impacted by the process. Using them not only would not help but could actually have negative effect, as it might prevent optimal consideration of quasi-identifier predictors in the partitioning (which would eventually result in more generalization on them). Second, we add the condition that a node can be split only if the resulting child nodes represent a partition of the parent relation with a sufficient number of tuples for satisfying the k-anonymity and ℓ-diversity requirements. (Intuitively, this requires the parent node to have at least $2k$ tuples to permit at least a binary split.) Each leaf node of a decision tree built considering these two changes is therefore a node that, by construction, represents a group of tuples of size at least k and with at least ℓ well-represented values for the sensitive attribute. This target-driven partitioning phase ensures to result in a (k, ℓ)-compliant decision tree formally defined as follows.

Definition 1 ((k, ℓ)-compliant decision tree). *Let $R(a_1, \ldots, a_n)$ be a relation with QI, s, and τ the quasi-identifier, sensitive, and target attributes in $\{a_1, \ldots, a_n\}$, respectively, and $DT(N, E)$ be a decision tree built over R for predicting target attribute τ, with N the set of nodes and E the set of edges. DT is a (k, ℓ)-compliant decision tree iff for each leaf node $n \in N$, the set of tuples R_n represented by n is such that $\mid R_n \mid \geq k$ and R_n includes tuples with at least ℓ well-represented values for s.*

Example 2 (Decision tree). Figure 4 illustrates an example of a decision tree built over the relation in Figure 2(a). The root node coincides with the whole table that is split over attribute `State`. The resulting child nodes correspond to the set of tuples related to employees working in California (CA), Minnesota (MN), and Texas (TX). For employees working in California, there is a further split that distinguishes between employees with age less than or equal to 40 and over 40. The leaf nodes are labeled with either 'Gov' or 'Non-gov' as job. For each node, attributes with a gray background are those used for splitting, and attributes with gray values are the attributes that cannot be used for splitting (i.e., sensitive attribute or attributes already used for split). In this tree, for

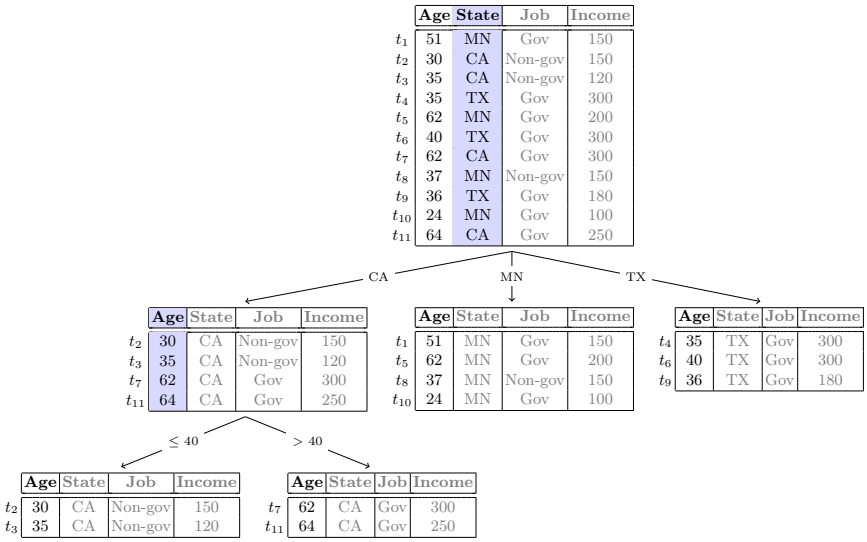


Fig. 4: An example of decision tree built over the relation in Figure 2(a)

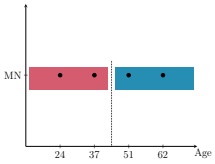
example, the first (from left) leaf node corresponding to the set $\{t_2,t_3\}$ of tuples is associated with a decision rule of the form "IF `State`=CA AND `Age`$\leq$ 40 THEN `Job` is 'Non-gov' ". Since each leaf node represents a group of at least two tuples with at least two different values for the sensitive attribute `Income` this is a $(2, 2)$-compliant decision tree.

4.2 Group Anonymization

The goal of the second phase is to independently anonymize each group of tuples represented by the leaf nodes of the (k, ℓ)-*compliant* decision tree built in the previous phase. The construction of the groups, clustering together tuples that are equal or close in values for predictors, ensures minimizing the impact of generalization on predictors. The problem becomes then to compute a generalization that produces a (k, ℓ)-anonymous version of each leaf node while minimizing information loss. This can be achieved with classical approaches for k-anonymity and ℓ-diversity. In particular, we consider the application of Mondrian [10], a multi-dimensional algorithm that provides an efficient and effective approach for achieving k-anonymity (which we consider extended with ℓ-diversity). Mondrian leverages a spatial representation of the data, mapping each quasi-identifier attribute to a dimension, and each combination of values of the quasi-identifier attributes to a point in such a multi-dimensional space (multiple tuples with the same coordinates translate into a point with a multiplicity greater than 1). Mondrian then recursively partitions the multi-dimensional space in two sub-spaces by selecting a dimension (i.e., an attribute in the quasi-identifier) and a split

	Age	State	Job	Income
t_1	51	MN	Gov	150
t_5	62	MN	Gov	200
t_8	37	MN	Non-gov	150
t_{10}	24	MN	Gov	100

(a)



(b)

	Age	State	Job	Income
t_1	[51-62]	MN	Gov	150
t_5	[51-62]	MN	Gov	200
t_8	[24-37]	MN	Non-gov	150
t_{10}	[24-37]	MN	Gov	100

(c)

Fig. 5: An example of a relational table (a) with its spatial representation and partitioning (b), and the corresponding $(2, 2)$-anonymous version (c)

point (i.e., a value in the domain of such an attribute), in such a way that each sub-space includes at least k points/tuples with at least ℓ different values for the sensitive attribute. The process terminates when any further partitioning would generate sub-spaces with less than k points (or the points in the sub-spaces would have less than ℓ different values for the sensitive attribute). Finally, all the tuples in each subspace are generalized to the same combination of (generalized) values for the quasi-identifier. The motivation for choosing Mondrian is that, while it being a well established reference in the field as efficient and effective approach, its approach to cutting multi-dimensional space to partition tuples is similar to how a decision tree works.

Example 3 (Anonymization). Consider the table in Figure 2(a) and the decision tree in Figure 4, where again the quasi-identifier is the pair ⟨`Age`,`State`⟩. Anonymization is applied independently on the four leaves. For the groups of two tuples no further split can be performed and hence only generalization is applied, reporting the age interval (instead of the specific values). For the group of four tuples, also reported in Figure 5(a) and in the multi-dimensional space in Figure 5(b), Mondrian will perform a split over attribute `Age` (the only one with different values) resulting in two groups to be generalized (Figure 5(c)).

The anonymized version of the dataset is finally obtained through the union of the anonymized groups represented by the leaf nodes of the (k, ℓ)-*compliant* decision tree. Clearly, being each group of tuples (k, ℓ)-anonymous, also the union is, as formally captured by the theorem in Appendix A. For instance, the whole dataset comprising the results of the anonymization of the different groups produced by the decision tree in Figure 4 is the table in Figure 2(b). Appendix B presents the algorithm used to compute a target-driven anonymization of a relation.

5 Experimental Results

We conducted a series of experiments to evaluate the effectiveness of DT-Anon in producing an anonymized dataset that can be used for training a classifier with good performance. In the following, we first describe the methodology applied and the datasets used in the experimental evaluation (Section 5.1), and then report and discuss the experimental results (Section 5.2).

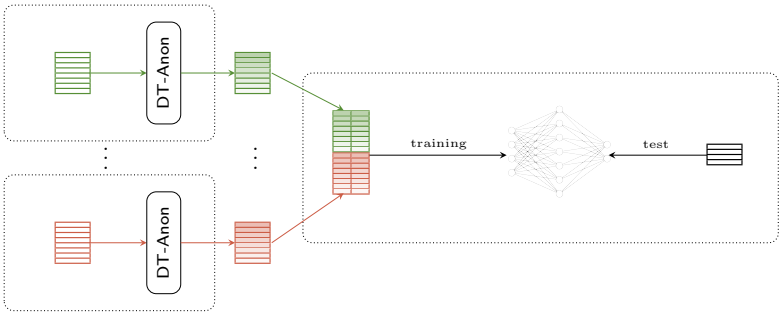


Fig. 6: Experimental scenario

5.1 Experimental Settings

Methodology. Our experimental evaluation has the goal of comparing the performance of a classifier when considering anonymization of the dataset of each single data controller running independently from the classification goal (i.e., following a classical approach) and when considering anonymization produced by DT-Anon. To evaluate the impact of anonymization on the classification task, we also evaluate the performance of the classifier trained over the original (raw) datasets.

Figure 6 illustrates the experimental scenario with multiple data controllers simulated by our experiments. We evaluated the classifier using different available datasets, partitioning the data in a training and a test set. To simulate the presence of multiple data controllers, we randomly partitioned the training set in different datasets (on which we then operated independently). In the paper, we report the results for the case of two data controllers, assuming two partitions of the original data on which the target-driven anonymization operated independently. For simplicity, we assumed k and ℓ (for which we considered different values) to be the same for all the data controllers.

As classifier, we considered a neural network, which implied transforming data into numeric variables to be fed in the neural network. Transformation, dependent on the type of attributes, worked as follows.

– *Categorical attributes*. Each categorical attribute is replaced with x binary attributes (one for each possible value of the attribute). Data values are then encoded through their representation via the binary attributes, setting to 1 the binary attribute(s) corresponding to their value(s). Scalar values will have only one binary attribute set to 1 (single-hot encoding) while sets of values (resulting from generalization) may have more than one attribute set to 1 (multi-hot encoding). For instance, with reference to the table in Figure 2(c), attribute `State` will be represented using three binary attributes ($\texttt{State}_{\texttt{CA}}$, $\texttt{State}_{\texttt{MN}}$, and $\texttt{State}_{\texttt{TX}}$), and its generalized value in tuple t_2 encoded with the first two attributes set to 1 and the latter set to 0.

– *Numerical attributes*. Scalar values of numerical attribute a are standardized, meaning that each value v is substituted with $(v - \mu_a)/\sigma_a$, where μ_a is the mean of the values in the relation for attribute a and σ_a is their standard deviation. For interval values (resulting from generalization), transformation is preceded by replacing each interval value with its mid point. For instance, with reference to our running example, `Age` interval [62-64] in generalized tuples t_7 and t_{11} is replaced by 63.

The transformed anonymized datasets have then been used for training a classifier. We built a neural network with three hidden layers with 64, 32, and 16 neurons, respectively. We used the ReLu activation function and the Adam optimizer. These are the default parameters also used in the scikit-learn[1] implementation.

Evaluation metrics. We evaluated the performance of the neural network (trained over anonymized datasets or raw datasets) by measuring the *accuracy* and the *F1$_{macro}$ score*. Accuracy is the ratio between the number of correct predictions and the total number of predictions. It then measures the percentage of correct classifications that a trained machine learning model achieves. F1_{macro} score is defined as the average of the class-wise F1 scores and is used for a multi-class classification problem. Formally, given a classification problem with a set C of classes, the F1_{macro} score is defined as: $\text{F1}_{\text{macro}} = \frac{\sum_{c \in C} \text{F1}(c)}{n}$ where $\text{F1}(c)$ is the F1 score (i.e., the harmonic means of precision and recall) computed for class c.

Datasets. We performed experiments on different publicly available real-world datasets. We report here the results on datasets: *Bank* and *Nursery*, from the UCI machine learning repository and *Customer_segmentation*, from the Kaggle platform. The datasets, whose attributes are reported in Table 1, are as follows.

– *Bank* dataset[2] describes individuals using both numeric and categorical attributes (45,211 tuples). The dataset refers to direct marketing campaigns based on phone calls related to a Portuguese banking institution. We consider as sensitive the binary attribute `default` that can assume two values stating whether the individual has credit in default. The target attribute is `y` that represents whether a client of the bank has subscribed a bank term deposit and has two possible values: 'yes' and 'no'.
– *Nursery* dataset[3] describes individuals using categorical attributes (12,960 tuples). It contains information derived from a hierarchical decision model that was realized to rank nursery school applicants. We consider as sensitive the categorical attribute `social` that represents the social conditions of the family of the applicant and can assume three values. The target attribute is `class` that represents the decisions and has five possible values: 'not_recom', 'recommend', 'very_recom', 'priority', and 'spec_prior'.

[1] https://scikit-learn.org/stable
[2] https://archive.ics.uci.edu/dataset/222/bank+marketing
[3] https://archive.ics.uci.edu/dataset/76/nursery

Table 1: Overview of the attributes in the considered datasets

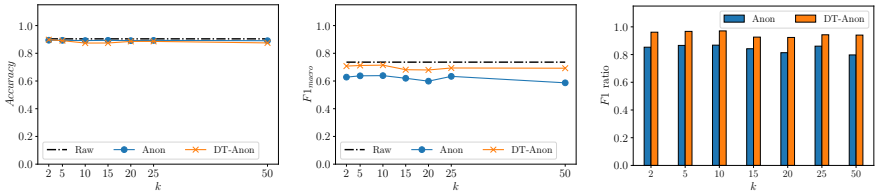
Dataset	QI		Target (τ)	Sensitive (s)	Others
	Categorical	Numeric			
Bank	job marital education housing loan	age balance duration	y (*2 values*)	default (*2 values*)	pdays previous campaign poutcome contact day_of_week month
Nursery	parents, has_nurs form children housing finance health		class (*5 values*)	social (*3 values*)	
Cust_segm	gender ever_married graduated profession	age work_exp fam_size	segment (*4 values*)	spending_score (*3 values*)	var_1

– *Customer_segmentation* dataset[4] describes individuals using both numeric and categorical attributes (13,330 tuples). The dataset includes information on the customers of an automotive company. We consider as sensitive the categorical attribute `spending` that represents the individual's spending score and can assume three values. The target attribute is `segment` that represents the market segment and has four possible values: 'A', 'B', 'C', and 'D'.

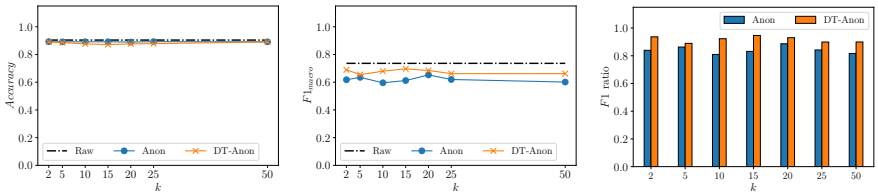
5.2 Results

Our experiments compare the performance of classifiers trained with anonymized datasets varying the privacy parameters k and ℓ. In the following, we use DT-Anon to refer to the neural network trained over a dataset anonymized with DT-Anon, and Anon to refer to the neural network trained over a dataset anonymized with a classical anonymization algorithm (which, in our implementation, is Mondrian revised to support the ℓ-diversity requirement). The performance of classifiers has been measured considering both the accuracy as well as the F1_{macro} score varying k (considering values 2, 5, 10, 15, 20, 25, 50) and ℓ (considering values 1, 2, 3, according to the number of distinct values for the sensitive attribute in the datasets). We consider as a baseline the neural network trained with the original (non anonymized) datasets. Figures 7-9 illustrate the results of our experiments. In the figures, the orange (light gray in b/w) lines and bars refer to DT-Anon and the blue (dark gray in b/w) lines and bars refer to Anon. For each dataset and for each value of ℓ, we draw two line charts and one bar chart showing: accuracy, F1_{macro} score, and the ratio between the F1_{macro} score of DT-Anon (or Anon) and the F1_{macro} score of our baseline, respectively. This

[4] https://www.kaggle.com/datasets/kaushiksuresh147/customer-segmentation



(a) ℓ=1; k=2,5,10,15,20,25,50



(b) ℓ=2; k=2,5,10,15,20,25,50

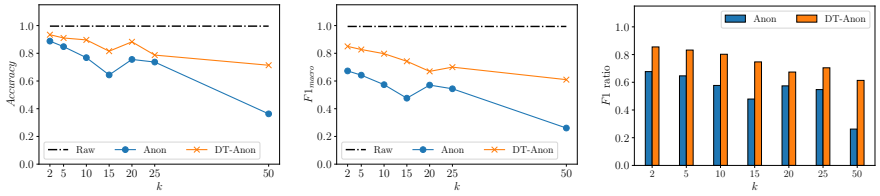
Fig. 7: Accuracy, F1_{macro}, and F1_{macro} ratio varying k and ℓ for the *Bank* dataset

ratio shows how well a supervised learning model (our neural network) learns from the anonymized datasets with respect to how well it learns from the original raw dataset. In other words, it shows how much the anonymization of the dataset impacts the capability of the model to learn from (anonymized) data. As it is visible from the figures, DT-Anon performs better than Anon with almost all datasets and values of k and ℓ.

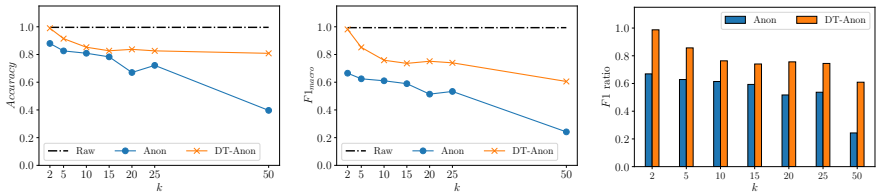
Accuracy. The experiments (line charts in the first column of Figures 7-9) show that the accuracy tends to decrease as the value of k and ℓ increases. This trend is due to the fact that higher values for k and ℓ require a higher amount of generalization, thus implying higher information loss that in turn produces a decrease in the effectiveness of the anonymized datasets in the learning process.

F1$_{\text{macro}}$. The F1_{macro} score has a similar trend as the accuracy across all datasets (line charts in the second column of Figures 7-9). The experiments also show that, even though the enforcement of both the k-anonymity and ℓ-diversity requirements with $\ell > 1$ is stricter than the enforcement of the k-anonymity requirement only, the impact on the capability of the neural network to learn from the anonymized data remains similar to the cases where $\ell = 1$. Furthermore, the F1_{macro} score of DT-Anon is constantly higher than the F1_{macro} score of Anon.

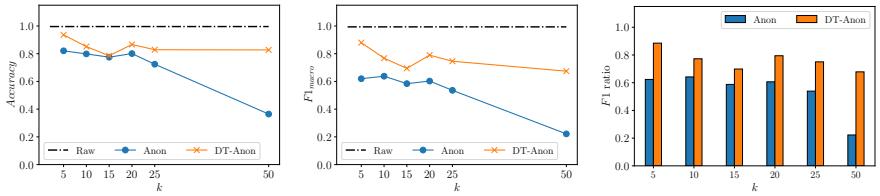
F1$_{\text{macro}}$ *ratio.* The F1_{macro} ratio of DT-Anon for the *Bank* and *Customer_segmentation* datasets remains always higher than 0.82 (bar chars in Figures 7- 9), again confirming that, with DT-Anon, the neural network preserves the capability of learning from the anonymized datasets. The F1_{macro} ratio of Anon is constantly lower. For the *Nursery* dataset the values are lower than those obtained for the other two datasets (from 0.60 with $k = 50$ and $\ell = 2$ to 0.98 with $k = 2$ and $\ell = 2$). Such values are, however, much higher than



(a) ℓ=1; k=2,5,10,15,20,25,50



(b) ℓ=2; k=2,5,10,15,20,25,50



(c) ℓ=3; k=2,5,10,15,20,25,50

Fig. 8: Accuracy, F1_{macro}, and F1_{macro} ratio varying k and ℓ for the *Nursery* dataset

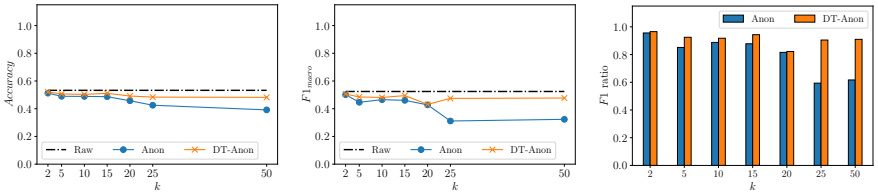
those obtained for Anon. This is probably due to the correlation between the quasi-identifier attributes and the target attribute in *Nursery*, which DT-Anon captures and preserves.

6 Related Work

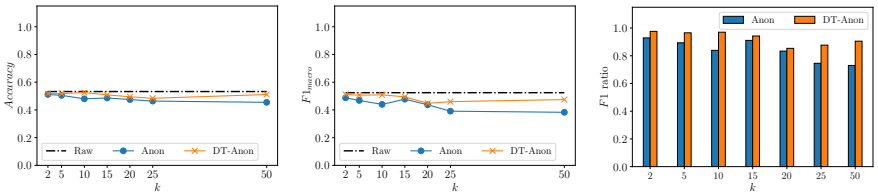
The problem of studying the effects of anonymization (e.g., k-anonymity, ℓ-diversity) on machine learning models has been the subject of several works (e.g., [3, 6, 9, 13, 14]).

Some of these proposals address the problem of evaluating the impact of different existing anonymization algorithms on the result of machine learning models (e.g., classifiers) and whether data anonymization can be enough to achieve privacy in machine learning (e.g., [13, 14]).

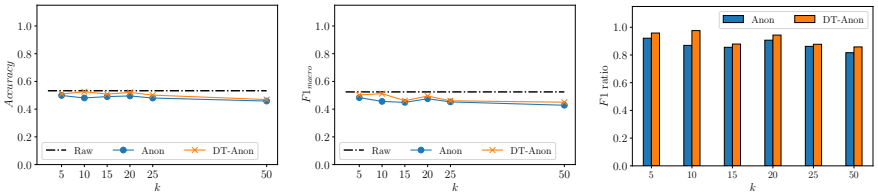
Other proposals instead consider a problem similar to the one addressed in this paper and define an anonymization strategy that takes into account the subsequent use of the anonymized datasets (e.g., [3, 6, 8, 9]). The work in [9] has introduced the problem of anonymizing data depending on a workload, which



(a) ℓ=1; k=2,5,10,15,20,25,50



(b) ℓ=2; k=2,5,10,15,20,25,50



(c) ℓ=3; k=2,5,10,15,20,25,50

Fig. 9: Accuracy, F1_{macro}, and F1_{macro} ratio varying k and ℓ for the *Customer_segmentation* dataset

can be a classification or regression model or selection/projection predicates (i.e., selection or projection predicates that identify a subset of the data on which the anonymization is applied). The authors propose a variation of Mondrian where data are split in a way that minimizes the weighted entropy over the set of resulting partitions without violating the k-anonymity requirement. The main differences with our approach are that we consider a scenario with multiple data controllers and we can anonymize data using any anonymization algorithm. The work in [6] defines a method for learning how to generalize unseen data for classification analysis. It starts from an existing machine learning model and learns how unseen data should be generalized by training a *generalized model* (i.e., a decision tree) with data labeled with the existing model's predictions. The decision tree is then used to derive a set of generalization ranges obtained by combining the split values of each attribute from the tree's internal nodes. While sharing with us the idea of using a decision tree to build groups of "similar tuples", the problem addressed is completely different. Also the work in [8] builds a decision tree to determine the attributes that most influence the value of the target attribute. Leaf nodes of the decision tree with more than k data items

are then anonymized by suppressing all attributes that are not used along the path from the root to the considered leaf node. Otherwise, a prune procedure is applied to obtain new leaf nodes of size at least k. The anonymization via suppression is then applied on these new leaf nodes. This proposal differs from our proposal in several aspects. The decision tree is build in a different way, the ℓ-diversity requirement is not considered, and the anonymization is enforced only through suppression, thus potentially reducing the information available for the classification task. In [3] the authors propose an approach for anonymizing data guided by relaxed functional dependencies. Such dependencies specify what subsets of attributes can be generalized and at which level, so to achieve a minimum level of anonymity (expressed through the k-anonymity requirement) while preserving data utility as much as possible. Data utility is measured in terms of classification accuracy and information gain. A set of generalization rules is extracted from the relaxed functional dependencies and then used for anonymizing datasets.

Other complementary solutions propose different anonymization strategies to improve the trade-off between privacy and utility also in machine learning scenarios (e.g., [15]).

7 Conclusions

We addressed the problem of anonymizing data in a scenario where multiple data controllers contribute to a classification task. We proposed DT-Anon, a data anonymization approach aware of and driven by the classification task downstream. DT-Anon enables data controllers contributing with their data to a classification task to anonymize their data while maintaining utility for the classification task. The experimental results confirm the ability of DT-Anon to limit, with respect to classical anonymization approaches, the information loss caused by data anonymization and hence its effect on the performance of the classification task. The paper leaves space for future work, including the consideration of other anonymization approaches (e.g., differential privacy) and the consideration of multiple data analytics tasks (e.g., tasks with different target attributes).

Acknowledgements This work was supported in part by the EC under projects EdgeAI (101097300) and GLACIATION (101070141), by the Italian MUR under PRIN project POLAR (2022LA8XBH), and by project SERICS (PE00000014) under the MUR NRRP funded by the EU - NGEU. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the Italian MUR. Neither the European Union nor the Italian MUR can be held responsible for them.

A Theorem and Proof

Theorem 1 ((k,ℓ)-anonymous dataset). *Let $R(a_1,\ldots,a_n)$ be a relation with QI, s, and τ the quasi-identifier, sensitive, and target attributes in $\{a_1,\ldots,a_n\}$, respectively, k and ℓ be the privacy parameters, and $DT(N,E)$ be a (k,ℓ)-*compliant *decision tree built over R for τ. The relation $\widehat{R} = \bigcup_{n\in N} \widehat{R}_n$, with $\widehat{R}_n$ the (k,ℓ)-anonymous version of relation R_n and $n \in N$ a leaf node of DT, is (k',ℓ)-anonymous, with $k' \geq k$.*

Proof. Let R_n be the partition of R represented by leaf node n in DT, and $\widehat{R}_n$ its (k,ℓ)-anonymous version. Table $\widehat{R}$ obtained merging the (k,ℓ)-anonymous partitions $\widehat{R}_{n_1},\ldots,\widehat{R}_{n_m}$ of the leaf nodes $n_1,\ldots,n_m$ of DT includes m (k,ℓ)-anonymous partitions of R. Since each combination of (generalized) values for the quasi-identifier QI have either 0 or at least k occurrences in each $\widehat{R}_{n_i}$, such a combination will have either 0 or at least k occurrences also in $\widehat{R}$. Since a specific combination of (generalized) values for the quasi-identifier could appear in more than one anonymized partition, the number of occurrences of such a combination of values could be higher than k (i.e., $\geq j \cdot k$ if appearing in j partitions). Therefore, $\widehat{R}$ is k'-anonymous with $k' \geq k$. Furthermore, table $\widehat{R}$ still satisfies the ℓ-diversity property since the groups of tuples having the same combination of (generalized) values for QI can only grow (due to the presence of groups of tuples with the same value for QI in more than a partition of R). Hence, the number of well represented values can either grow or remain the same. □

B DT-Anon Algorithm

We now describe DT-Anon algorithm that enforces the target-driven anonymization. Figure 10 illustrates the pseudocode of the algorithm implementing the two phases of our approach.

Target-driven partitioning. Figure 10 illustrates the pseudocode of **BuildDT**, a recursive procedure used in the first phase of the DT-Anon algorithm for computing a (k,ℓ)-*compliant* decision tree. Procedure **BuildDT** receives as input a node n of the decision tree (which corresponds to the root node at its first invocation). The procedure first identifies the set $R_n \subseteq R$ of tuples that satisfy the decision rule, denoted d_n, associated with node n (line 1). The procedure then verifies whether such a set of tuples can be further split (line 2). Existing decision tree algorithms split a node into child nodes until a stopping criterion is met or until the node represents a set of tuples with an homogeneous value for the target attribute. The **BuildDT** procedure adds a further check and verifies whether the set R_n of tuples represented by the considered node includes at least $2k$ tuples (i.e., at least a binary split can be enforced on the node). In this case, the procedure identifies all the possible candidate splits for n (line 3), considering the available attributes and partitions of their domains (like classical

Input
$R(a_1, \ldots, a_n)$: original relation
QI: quasi-identifier attributes in $\{a_1, \ldots, a_n\}$
s: sensitive attribute in $\{a_1, \ldots, a_n\}$
τ: target attribute in $\{a_1, \ldots, a_n\}$
k: anonymity requirement
ℓ: diversity requirement

Output
$\widehat{R}$: (k, ℓ)-anonymous version of R

DT-Anon
/* **Phase 1**: Compute a (k, ℓ)-*compliant* decision tree $DT(N, E)$ */
1: $N :=$ ROOT; $E := \emptyset$ /* set N of nodes and set E of edges of DT */
2: **BuildDT**(ROOT) /* ROOT node representing R */
/* **Phase 2**: anonymize the leaves of DT */
3: $\widehat{R} := \emptyset$
4: **for each** leaf node $n \in N$ **do**
5: R_n=$d_n(R)$
6: $\widehat{R} := \widehat{R} \cup$ **Anonymize**(R_n)
7: **return**$(\widehat{R})$

BuildDT(n)
1: let R_n=$d_n(R)$ /* set of tuples in R satisfying decision rule d_n of n/*
2: **if** $|R_n| \geq 2k$ AND $\exists t_i, t_j \in R_n$: $t_i[\tau] \neq t_j[\tau]$ AND stop condition is not satisfied
3: **then** let *Split* be all possible splits of R_n
4: **repeat**
5: choose the most promising *split* in *Split*
6: let N' be the set of nodes resulting applying *split* on R_n
7: *found* := TRUE
8: **while** *found*=TRUE AND $N' \neq \emptyset$ **do**
9: let $n_i \in N'$ and R_{n_i}=$d_{n_i}(R_n)$
10: **if** $|R_{n_i}| < k$ OR R_{n_i} has less than ℓ well-represented values for s
11: **then** *found* := FALSE
12: **else** $N' := N' \setminus \{n_i\}$
13: *Split* := *Split* $\setminus$ {*split*}
14: **until** *Split*=$\emptyset$ OR *found*=TRUE
15: **if** *found*=TRUE
16: **then** let N' be the set of nodes resulting applying *split* on R_n
17: $N := N \cup N'$
18: **for each** $n_i \in N'$ **do**
19: $E := E \cup (n, n_i)$
20: **BuildDT**(n_i)

Fig. 10: Pseudocode of the DT-Anon algorithm

algorithms for building a decision tree). The procedure then evaluates, in decreasing order or effectiveness on the classification, the candidate splits checking whether the considered split *split* guarantees that the decision tree satisfies Definition 1 (lines 4-14). If *split* produces a (k, ℓ)-*compliant* decision tree, the split is enforced and the procedure recursively invokes itself on each (child) node resulting from the split (lines 15-20). As an example, suppose that the data owner wishes to compute a $(3, 1)$-anonymous version of the relation in Figure 2(a). DT-Anon starts by invoking procedure **BuildDT** that, as shown in Figure 4, splits the relation on attribute `State` in three child nodes. The second split of the first (from left) child node in Figure 4 would be instead prevented because it generates two partitions with less than 3 tuples each.

Group anonymization. The second phase of the DT-Anon algorithm in Figure 10 consists in independently anonymizing the (sub)relations represented by the leaf nodes of the (k, ℓ)-*compliant* decision tree built in the first phase. For each leaf node $n \in N$ of the decision tree, the algorithm invokes an anonymization algorithm on relation R_n, and returns the (k, ℓ)-anonymous version of R_n, which is appended to the other (k, ℓ)-anonymous relations.

References

1. Abukmeil, M., Ferrari, S., Genovese, A., Piuri, V., Scotti, F.: A survey of unsupervised generative models for exploratory data analysis and representation learning. ACM CSUR **54**(5), 1–40 (June 2021)
2. Bhattacharjee, K., Islam, A., Vaidya, J., Dasgupta, A.: PRIVEE: A visual analytic workflow for proactive privacy risk inspection of open data. In: Proc. of IEEE VizSec. Oklahoma City, OK, USA (October 2022)
3. Caruccio, L., Desiato, D., Polese, G., Tortora, G., Zannone, N.: A decision-support framework for data anonymization with application to machine learning processes. Information Sciences **613**, 1–32 (October 2022)
4. Ciriani, V., De Capitani di Vimercati, S., Foresti, S., Samarati, P.: k-Anonymity. In: Yu, T., Jajodia, S. (eds.) Secure Data Management in Decentralized Systems. Springer-Verlag (2007)
5. De Capitani di Vimercati, S., Foresti, S., Livraga, G., Samarati, P.: Data privacy: Definitions and techniques. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **20**(6), 793–817 (December 2012)
6. Goldsteen, A., Ezov, G., Shmelkin, R., Moffie, M., Farkas, A.: Data minimization for GDPR compliance in machine learning models. AI Ethics **2**(3), 477–491 (August 2022)
7. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers (2006)
8. Kisilevich, S., Rokach, L., Elovici, Y., Shapira, B.: Efficient multidimensional suppression for k-anonymity. IEEE TKDE **22**(3), 334–347 (March 2010)
9. LeFevre, K., DeWitt, D., Ramakrishnan, R.: Workload-aware anonymization. In: Proc. of KDD. Philadelphia, PA, USA (August 2006)
10. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Mondrian multidimensional k-anonymity. In: Proc. of ICDE. Atlanta, GE, USA (2006)

11. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: ℓ-diversity: Privacy beyond k-anonymity. ACM TKDD **1**(1) (2007)
12. Samarati, P.: Protecting respondents identities in microdata release. IEEE TKDE **13**(6), 1010–1027 (2001)
13. Senavirathne, N., Torra, V.: On the role of data anonymization in machine learning privacy. In: Proc. of IEEE TrustCom. Guangzhou, China (December 2020)
14. Slijepčević, D., Henz, M., L.D.l Klausner, Dam, T., Kieseberg, P., Zeppelzauer, M.: k-anonymity in practice: How generalisation and suppression affect machine learning classifiers. COSE **111** (December 2021)
15. Verdonck, J., De Boeck, K., Willocx, M., Lapon, J., Naessens, V.: A hybrid anonymization pipeline to improve the privacy-utility balance in sensitive datasets for ml purposes. In: Proc. of ARES. Benevento, Italy (August-September 2023)