m-Eligibility With Minimum Counterfeits and Deletions for Privacy Protection in Continuous Data Publishing

Adrián Tobar Nicolau, Javier Parra-Arnau, Jordi Forné, and Esteve Pallarés

***Abstract*— Continuous data publishing consists in the republication of updating microdata. The most relevant syntactic notions in continuous data publishing are based on m-invariance. This notion enforces that no user can be distinguished among, at least, $m-1$ other users, each with distinct secret data. To achieve m-invariance, the existing methods must first alter the dataset to satisfy a property called m-eligibility. Essentially, a dataset can be made m-invariant if and only if it satisfies the m-eligibility constraint. Although guaranteeing the m-eligibility property is a crucial step, no theoretical study of the best strategies to achieve it has been carried out. This paper performs such a study by giving strategies and demonstrating their optimality under two approaches: insertion of counterfeit tuples and partial publication. The empirical evaluation of our proposal shows a significant reduction on the number of modifications needed to enforce m-eligbility of up to 41% with respect to the literature.**

***Index Terms*— m-invariance, m-eligibility, syntactic privacy, data privacy, dynamic data.**

I. Introduction

DURING the last decades an increasing number of privacy mechanisms have been proposed in the literature for microdata, that is, data concerning individuals.[1] Among the classical mechanisms, there exists a distinction between syntactic methods, such as k-anonymity [1], l-diversity [2] and t-closeness [3], [4], which ensure a privacy property on the anonymized data, and semantic methods like (ϵ, δ)-differential privacy [5], [6], [7], which enforce the property on the mechanism anonymizing the data. Another key aspect that differentiates them concerns the adversary's background knowledge. While syntactic methods need to specify what information is available to the adversary, semantic methods need not. As a result, k-anonymity and its enhancements classify the attributes of the microdata in quasi identifiers and confidential attributes. The former, which per se are not sensitive, are employed by the adversary to, e.g., reidentify them; and the latter are the confidential information the adversary aims to learn. In general, each approach has its own strengths and weaknesses, but syntactic approaches are preferred to retain utility in the data at the expense of a rigid definition of the attacker.

With the increased necessity to access other forms of data, different dynamic publishing scenarios for microdata have emerged.

- Multiple data release [8]: several views of the same underlying dataset are published simultaneously, i.e., publications with all records but only a subset of their attributes.
- Sequential data release [9], [10]. Partial views of a dataset are published. In general, the number of tuples is fixed.
- Continuous data publishing [11]. Publications of a dataset that is being updated in between releases. The dataset changes via insertions, deletions, reinsertions, and updates of tuples. The set of attributes is fixed.

In continuous data publishing, the main syntactic mechanisms are based on the m-invariance notion [12] and their variations [13], [14], [15], [16], [17], [18]. This notion is deeply related to the m-eligibility property [2], that is, that the dataset has no more than $\frac{1}{m}$ fraction of tuples with the same sensitive value.

The problem of imposing m-eligibility is relevant since it is a necessary condition to achieve recursive $l-$diversity [2] and m-invariance. Essentially, a dataset can be made m-invariant if and only if it satisfies the m-eligibility constraint. This relation was already considered in [12]. To solve this limitation, the main approach in the literature has been to add artificial tuples (counterfeits) to the dataset to make it m-eligible. However, the methodologies used to add the counterfeits (explained in Section IV-B) are not properly justified and are generally overly optimistic.

Manuscript received 25 January 2023; revised 27 July 2023 and 8 November 2023; accepted 10 January 2024. Date of publication 15 January 2024; date of current version 29 January 2024. This work was supported in part by the Spanish Government through the Project "Enhancing Communication Protocols with Machine Learning while Protecting Sensitive Data (COMPROMISE)" funded by MCIN/AEI/10.13039/501100011033 under Grant PID2020-113795RB-C31; in part by the Project "Anonymization Technology for AI-Based Analytics of Mobility Data (MOBILYTICS)" funded by MCIN/AEI/10.13039/501100011033 under Grant TED2021-129782B-I00; and in part by the European Union "NextGenerationEU"/Plan de Recuperación, Transformación y Resiliencia (PRTR), funded by Generalitat de Catalunya, under Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR) Grant 2021 SGR 01413. The work of Javier Parra-Arnau was supported by the Spanish Ministry of Science and Innovation and the European Union—NextGenerationEU/PRTR through the "Ramón y Cajal" Fellowship under Grant RYC2021-034256-I. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Hossein Pishro-Nik. *(Corresponding author: Javier Parra-Arnau.)*

The authors are with the Department of Network Engineering, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain (e-mail: javier.parra@upc.edu).

Digital Object Identifier 10.1109/TIFS.2024.3354557

[1] Microdata are database tables whose records carry data concerning individual subjects.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

As an alternative to adding counterfeits, the authors of [18] present a new approach based on partial publication. The main idea is to extract one tuple for each distinct sensitive value in the dataset to create a so-called Cach table. Before publication, if the dataset is not m-eligible, tuples from the Cach table are reinserted in the dataset to achieve m-eligibility. From what is presented in their empirical tests, no counterfeit was added during their evaluation. Our study (presented in Section III) reveals that this Cach method is also overly optimistic in its ability to enforce m-eligibility.

A. Contributions and Organization of This Paper

The aim of this paper is to provide effective methods to obtain m-eligible datasets with minimal perturbation with respect to the original microdata. In particular, our proposal studies how to enforce m-eligibility in a dataset with counterfeits, deletions, or a combination of both in such a way that the number of modifications performed is minimized. No previous work has studied how to solve this problem. Due to the profound connection between m-eligibility and m-invariance we focus on this notion.

To support our theoretical results we carry three different empirical evaluations in Section IV. First, we compare the effectiveness of our methods against each other, that is, we investigate whether there is a better choice among them. We conclude that they are all reasonable options, with the hybrid approach being the best. Second, we compare the effectiveness of our approach with the state of the art, in particular with [14], for the problem of enforcing m-eligibility on an arbitrary dataset. Finally, we compare our counterfeiting mechanism in a dynamic data publishing scenario to ensure that the quality of our results outperforms the state of the art. All experiments show clear improvements over the literature.

The paper is structured as follows. Section II motivates the use of syntactic privacy, introduces the basic definitions of m-eligibility, the m-invariance problem, preliminary results, and an overview of the existing algorithms based on m-eligibility. Then Section III presents the main contributions of this paper for the different approaches to obtaining m-eligible datasets, providing proofs for upper bounds, correct execution, and optimality of the algorithms. After that, an evaluation of behaviour is done in Section IV where the methods are subject to testing in both static and dynamic scenarios, with a comparison made to existing literature. The experiments evaluate the utility Section IV-B is devoted to the presentation of related work and its comparison with our results. The paper ends with Section V, which summarizes the main conclusions and possible strands of future work.

II. Background & State of the Art

This section examines the privacy notion of m-invariance, its relationship with m-eligibility, describes the main difference between syntactic and semantic notions, and overviews state of the art algorithms which depend on the enforcement of m-elibility. This will provide readers with the necessary depth to understand the technical contributions of this work.

A. Syntactic Vs Semantic Privacy

Differential privacy provides a mathematically rigorous framework for privacy protection. It guarantees that the inclusion or exclusion of an individual's data does not significantly affect the outcome of the analysis, thus safeguarding against re-identification attacks. However, differential privacy was not designed with continuous data publishing in mind [5]. Republishing differentially private data multiple times can lead to cumulative privacy loss. Each time data is republished or shared, additional noise may be introduced, further degrading the utility of the data. This cumulative effect can result in a gradual loss of privacy guarantees over multiple republishing iterations, potentially compromising the privacy protections initially provided by differential privacy or the expected data utility. This behaviour limits the number of publications that can be made using differential privacy to a fixed value.

Syntactic privacy provides fine-grained control over data privacy. This means that data owners and curators can define privacy rules and access controls at various levels, such as attribute or field level. By customizing privacy protections based on the particularities of each specific data element, syntactic privacy allows for a tailored approach to republishing data. This granular control ensures that privacy is appropriately enforced while still enabling access to meaningful and relevant information. Furthermore, the use of existing proposals such as [14] and [16] enforces a constant bound on the privacy risk for a theoretically unlimited number of republications.

Differential privacy has increased in popularity in recent years, but it is, as we have previously stated, not suitable for all private data publications [19] and is susceptible to similar attacks to those to which syntactic privacy is vulnerable [20] such as DeFinetti attacks and background knowledge attacks.Furthermore, as of today, the only existing proposals DP-based in dynamic data publishing are [21], [22] which are limited to data streams.

In summary, although the privacy provided by syntactic privacy is weaker than that of differential privacy, it is the only one that allows infinite republication of the same information while retaining both utility and privacy simultaneously for an arbitrary number of republications.

B. m-Invariance

m-Invariance [12] was the first method to allow the republication of microdata after being modified with insertions and deletions. A class is a set of tuples with common quasi-identifiers. The m-invariant method enforces that each class has at least m tuples, no two tuples with the same sensitive value, and that if a tuple appears in two releases, both classes where it appears share the same set of sensitive values (signature). The motivation behind this definition is to avoid intersection attacks. An intersection attack is based on partially linking a user to a reduced set of tuples for each publication. The sensitive value must appear in the intersection of the signatures of each candidate set. Since m-invariance enforces that two classes with a common tuple share the same signature, this attack may not reduce such intersection to less than m sensitive values. See Figure 1 for an example.

Id	AGE	S.V.
1	[18-20]	HIV
2	[18-20]	FLU

(a) First 2-diverse publication.

Id	AGE	S.D.
1	[18-19]	HIV
3	[18-19]	ACNE
2	[20-21]	FLU
4	[20-21]	COUCH

(b) Second 2-diverse but not 2-invariant publication.

Id	AGE	S.D.
1	[18-20]	HIV
2	[18-20]	FLU
3	[19-21]	ACNE
4	[19-21]	COUCH

(c) Second 2-invariant publication.

Fig. 1. Example of intersection attack. If an attacker is searching for information of a participant with $age = 18$, from Table Ia they can learn he/she has sensitive value HIV or FLU. From the second publication (Table Ib, on the other hand, the attacker can learn the participant has either HIV or ACNE. Intersecting both sets of possible sensitive values for the participant, the adversary concludes the attacked tuple has HIV. Such attacks are avoidable using m-invariance; in this case, publishing Table Ic instead of Table Ib.

Most algorithms that implement m-invariance or their variations have the same core structure: on the first publication, it enforces that the input dataset is m-eligible; after that, the m-invariant structure is established and the dataset published.

For the consecutive releases, a distinction is made between never-published tuples (new) and published ones (old). The old tuples are structured in the same class as their last publication. If a class is missing a tuple due to a deletion, a new tuple is put in as a replacement. If none exists, a counterfeit tuple is inserted. For the remaining new tuples that are not in a class, the m-invariant structure is enforced. Finally, the whole dataset is published.

Methods that rely on this procedure lack a detailed explanation of how to impose m-eligibility over the datasets. Most publications argue that they can achieve it with the insertion of a few counterfeits, but without a deeper insight into how they accomplish it. Our results show that an optimal strategy exists.

C. m-Invariance and m-Eligibility

Let T be a microdata table (dataset) of n individuals and d attributes, i.e., a matrix $A \in \mathbb{R}^{n\times d}$. The matrix A has the form $(QI|SD)$ where $QI \in \mathbb{R}^{n\times d-1}$ and $SD \in \mathbb{R}^{n\times 1}$. We denote the row $a_{i1}, \ldots, a_{id-1}$ as the quasi identifiers of tuple i and the value a_{id} as the sensitive value of tuple i. We define the m-invariant problem as follows:

Definition 1 (m-invariant problem): Given a dataset T with l distinct sensitive values and a number $m \in [2, l]$, the m-invariant problem is partitioning T into subsets of tuples (clusters) of at least size m satisfying that no two tuples in the same subset have the same sensitive value.

In general, this problem can have no feasible solutions, for instance, when a sensitive value is much more frequent than the rest (see Proposition 1).

Definition 2: Let $T \in \mathbb{R}^{n\times d}$ be a dataset with l distinct sensitive values.

- *We denote by $|T|$ the number of tuples, i.e., the number of rows in T.*
- *We denote by $\{c_1, \ldots, c_l\}$ the counts of each sensitive value (there are c_1 tuples with sensitive value sd_1 and so on).*

Now we state the m-eligibility condition.

Definition 3: A dataset $T \in \mathbb{R}^{n\times d}$ is m-eligible if no more than $\frac{|T|}{m}$ tuples have the same sensitive value in the dataset.

D. m-Eligibility Based Algorithms

Our main contributions are optimal approaches with respect to the number of counterfeits/deletions needed to obtain an m-eligible dataset. To put this in context, the literature has developed various techniques for dynamic data publishing and recognises the need to enforce m-eligibility as a prerequisite on the dataset. The different proposals do not discuss the difficulty of enforcing m-eligibility over a dataset and use simple heuristics. We are the first to provide an optimal approach to this problem. In this section, we discuss the diverse statements and results presented in the literature with regards to the enforcement of m-eligibility.

The authors of [14] present τ-safety as an improvement of m-invariance, that expands the privacy guarantee even with reinsertions of previously deleted tuples. They present an algorithm with better performance than the m-invariance proposal, giving lower query error, time cost and number of counterfeits.

Another related work is [18], which propose an alternative implementation of τ-safety that improves utility and privacy by considering non-consecutive reinsertions of tuples. Their main proposal is the creation of a Cach table. This table is created by extracting, for each senstive value, one tuple with that attribute. Whenever the dataset needs to be made m-eligible, they utilize tuples from the Cach table to enforce m-eligibility. Since only one tuple for each sensitive value exists in Cach, the capacity to enforce m-eligibility is limited unless one tuple is added more than once. In their empirical evaluation, no countefeit was added, which implies that all their datasets could be made m-eligible by removing, at most, one tuple for each sensitive value.

An improvement on the efficiency of the τ-safety algorithm is proposed in [16] that employs Cuckoo filters. The authors, however, do not provide any insight on how to make a dataset m-eligible. Due to the lack of more information, we are unable to develop a fair comparison between their solution and our theoretical results in an empirical evaluation. Nonetheless, our contributions do not depend on any empirical argument to be justified since they are formally proven.

Finally, we would like to mention [23], which investigates the problem of submitting false ratings and suppressing genuine ratings in the context of untrusted recommender systems. Although the scenario is completely different to the one at hand and the object of protection is a probability distribution, the theoretical analysis of forgery (i.e., the equivalent to generating counterfeits) and suppression derives analogous

optimal bounds on the rates of forgery and suppression beyond which any privacy risk is vanished.

III. m-Eligibility With Minimum Counterfeits and Deletions

This section contains the main results of this paper. We start with the necessary definitions and results and then introduce the m-invariant problem with counterfeits and with partial publication. Each subsection provides different properties that are used to prove a strategy to obtain m-eligible datasets while minimizing the counterfeits and deletions necessary, respectively. Additionally, for each problem, an upper bound on the minimal number of counterfeits/deletions necessary to obtain m-eligibility is provided. To conclude, the hybrid problem is presented and a fast strategy to compute a solution is discussed.

We focus our study on enforcing m-eligibility for an arbitrary dataset. This reduction is possible because for any subsequent release, there is a strategy to simplify it to an initial release. For any non-initial release, we separate the dataset into T_{new} and T_{old} of new tuples and previously published ones. Then we reconstruct the cluster of the tuples of T_{old} as in the previous release. If some clusters are incomplete, we add tuples from T_{new} to complete them; If none are available, add counterfeits. This step is a deterministic process that always generates the same number of counterfeits. Then, for the remaining tuples of T_{new} enforce m-eligibility and proceed as a first release. As we can see, for any subsequent release, the problem is reduced to enforcing m-eligibility to an arbitrary dataset. The difficulty of achieving m-eligibility is highly dependent on how T has changed via additions and deletions.

Proposition 1: A dataset $T \in \mathbb{R}^{n\times d}$ has a feasible solution for the m-invariant problem if and only if T is m-eligible.

Proof: First, we prove that m-eligibility implies the existence of a solution. The original argument appears in [12]. We will show that for any m-eligible dataset T with decreasing counts $\{c_1, \ldots, c_l\}$ there exist $\alpha \geq 1$ and $\beta \in [m, l]$ such that $\{c_1-\alpha, c_2-\alpha, \ldots, c_\beta-\alpha, c_{\beta+1}, \ldots, c_l\}$ is m-eligible. Clearly, if the previous claim holds, we can use it repeatedly to divide the dataset into classes of size at least m.

Let $|T| = n$, $\alpha = 1$ and $\beta = \max(i, m)$ where i is the largest value such that $c_i = c_1$, in other words, $c_{i+1} + 1 \leq c_1$. We distinguish two cases.

- If $\beta = m$: From m-eligibility, we have $c_1 \leq \frac{n}{m}$ which implies $c_1 - 1 \leq \frac{n}{m} - 1 = \frac{n-m}{m}$ proving the m-eligibility.
- If $\beta > m$: From the definition of β we have $c_1\beta = \sum_{j=1}^{i} c_i \leq n$ which implies $c_1 \leq \frac{n}{\beta}$. We derive $c_1 - 1 \leq \frac{n}{\beta} - 1 < \frac{n-\beta}{m}$ as desired.

Now we prove that the existence of a solution implies m-eligibility. Assume there exists a feasible solution for the m-invariant problem and the most frequent sensitive value of T with count c satisfies $c > \frac{n}{m}$ (negation of m-eligibility). Since each class must have at least m tuples, the number of classes k satisfies $km \leq n$ (with equality only when all classes have size m). We deduce that $k \leq \frac{n}{m} < c$. But since no class can have two tuples with the same sensitive value, we have $c \leq k$ contradicting our previous inequality. □

Proposition 1 implies that in order to guarantee a solution for non m-eligible datasets, some form of relaxation to the combinatorial problem must be made. Two possible variations exist: the counterfeit method and the partial publication (Cach) [18]. We now state the necessary notation to formalize the counterfeit method.

Definition 4: A dataset $T' \in \mathbb{R}^{p\times d}$ is:

- *A subset of dataset $T \in \mathbb{R}^{n\times d}$ if it is a submatrix of p rows of T. We indicate it as $T' \subseteq T$.*
- *A superset of dataset $T \in \mathbb{R}^{n\times d}$ if $T \subseteq T'$.*
- *A maximal m-eligible subset of T (if it is an m-eligible subset and) if $|T'| \geq |T''|$ holds for any m-eligible subset T'' of T.*
- *A minimal m-eligible superset of T (if it is an m-eligible superset and) if $|T'| \leq |T''|$ holds for any m-eligible superset T'' of T.*

With the previous definitions, we are now able to present the counterfeit approach.

A. m-Invariant Problem With Counterfeits

Since the first publication on m-invariance [12] the necessity to enforce m-eligibility has been tackled with the addition of counterfeit tuples to the dataset [13], [14], [16]. Despite that, no study on how to minimize the number of counterfeit tuples has been carried out. This Section gives tight results, showing the minimal number of counterfeit tuples needed to enforce m-eligibility and an algorithm that achieves that optimal bound.

First, we state the m-invariant problem with counterfeits.

Definition 5 (m-invariant problem with counterfeits): Given a dataset T with l distinct sensitive values and a number $m \in [2, l]$, the m-invariant with counterfeits problem is partitioning $T' \supseteq T$ into subsets of tuples (clusters) of at least size m satisfying that no two tuples in the same subset have the same sensitive value, where T' is a minimal m-eligible superset of T.

Proposition 2 determines the minimum number of tuples needed to obtain a minimal m-eligible superset.

Proposition 2: Let $T \in \mathbb{R}^{n\times d}$ be a dataset and let $T' \in \mathbb{R}^{j\times d}$ be a minimal m-eligible superset of T then

$$|T'| - |T| = \max(0, cm - n)$$

where c is the number of tuples with the most frequent sensitive value in T.

Proof: Observe that the m-eligibility condition $c - \frac{n}{m} \leq 0$, whenever we add a tuple with a new sensitive value changes to $c - \frac{n}{m} - \frac{1}{m}$ and, in general, for x tuples to $c - \frac{n}{m} - \frac{x}{m}$, is then straightforward that the minimal number of tuples to be added to ensure $c \leq \frac{j}{m}$ is at least $cm - n$. That can be achieved if we add $cm - n$ tuples, each with a unique sensitive value not appearing in the dataset. □

In general, the previous result can be of no interest since the addition of new sensitive values can be detrimental for the practical objectives of the computation. Next, we present an improvement to Proposition 2 since it does not need the insertion of new sensitive values.

Proposition 3: Let $T \in \mathbb{R}^{n\times d}$ be a dataset with $l \geq m$ distinct sensitive values and let $T' \in \mathbb{R}^{j\times d}$ be a minimal m-eligible superset of T with $SD(T') \subseteq SD(T)$ then

$$|T'| - |T| = \max(0, cm - n),$$

where c is the number of tuples with the most frequent sensitive value in T and $SD(T)$ is the set of sensitive values of T.

Proof: Assume T is not m-eligible; otherwise, the proof is trivial. Consider T' the minimal m-eligible superset of the proof of Proposition 2. The counts of sensitive values of T and T' are $\{c_1, \ldots, c_l\}$ and $\{c_1, \ldots, c_l, c_{l+1} = 1, \ldots, c_k = 1\}$, in descending order, respectively. Now observe that $c_1 \geq c_l + 1$ otherwise $c_i = c_j$ for all $i, j \in [1, l]$ which would imply that T is m-eligible. Consider now the process of changing the sensitive value of the tuple with sensitive value k to c_l, which yields a dataset T_1 with counts $\{c_1, \ldots, c_l+1, 1, \ldots, 0\}$ which is clearly m-eligible since $c_l + 1 \leq c_1 \leq \frac{|T'|}{m}$, as previously stated. We can repeat this strategy with T_1, i.e., at each step remove a tuple with a sensitive value in $[l+1, \ldots, k]$ and add a new tuple with the least frequent sensitive value in $[1, l]$, thus maintaining the m-eligibility of the dataset. After the last tuple with a sensitive value in $[l+1, \ldots, k]$ is replaced, we will have a minimal m-eligible superset of T with l distinct sensitive values. □

From the proof of Proposition 3 we yield an algorithm to compute minimal m-eligible supersets.

Corollary 1: Let $T \in \mathbb{R}^{n\times d}$ be a non m-eligible dataset with $l \geq m$ sensitive values. Algorithm 1 computes a minimal m-eligible superset of T.

Algorithm 1 Algorithm to Obtain Minimal m-Eligible Superset of a Given Dataset T

```
Data: dataset T
Result: minimal m-eligible superset of T
while T not m-eligible do
    T = T ∪ {t} ;   /* t with least frequent
      sensitive value in T */
end
return T;
```

Observe that step 2 of the Algorithm 1 could choose between several options, showing that more than one optimal solution exists.

Assume that a data holder is interested in making Table I 3-eligible using counterfeits. From Proposition 3, they know that they need to make $5 \cdot 3 - 10 = 5$ counterfeits. They run Algorithm 1 obtaining counts 5, 4, 3, 3 for the attributes FLU, ACNE, ADHD, and HIV, respectively. Since the counts of Table I are {5, 3, 1, 1} they add 1 counterfeit with attribute ACNE, 2 counterfeits with ADHD and 2 counterfeits with HIV matching the number of counts with those retrieved from Algorithm 1. Table II(a) shows the result.

B. m-Invariant Problem With Partial Publication

Recently, the authors of [18] raised a new strategy to tackle the m-invariant problem: instead of adding counterfeits, they considered the removal of a small sample of tuples, which they used in substitution of counterfeits. This subsection is devoted to the presentation of our results in relation to this problem, namely, we provide an upper bound on the minimal number of deletions as well as an algorithm that constructs an optimal solution.

TABLE I
EXAMPLE DATASET WITH SENSITIVE VALUE COUNTS {5, 3, 1, 1}. THE DIFFERENT RESULTS OF ENFORCING 3-ELIGIBILITY APPEAR IN TABLE II

ID	AGE	S.D.
1	15	FLU
2	18	FLU
3	18	FLU
4	19	FLU
5	22	FLU
6	15	ACNE
7	17	ACNE
8	19	ACNE
9	15	ADHD
10	18	HIV

First, we state the m-invariant problem with partial publication.

Definition 6 (m-invariant problem with partial publication): Given a dataset T with l distinct sensitive values and a number $m \in [2, l]$, the m-invariant partial publication problem is partitioning $T' \subseteq T$ into subsets of tuples (clusters) of at least size m satisfying that no two tuples in the same subset have the same sensitive value, where T' is a maximal m-eligible subset of T'.

This process demands a previous computation of T'. We present a fast strategy to find one instance.

Proposition 4: If $T' \subseteq T$ is a maximal m-eligible subset of T and $\{c_1, \ldots, c_l\}$, $\{c'_1, \ldots, c'_l\}$ are the counts of each sensitive value in T and T' respectively (possibly 0) then for all $i \in [1, l]$ $c'_i \leq c_i - \max(0, \lceil \frac{mc_i - n}{m-1} \rceil)$.

Proof: Observe that a dataset T is m-eligible if for all $i \in [1, l]$ holds $c_i - \frac{n}{m} \leq 0$. Notice that the function $f_i(x, y) = c_i - x - \frac{n-x-y}{m}$ returns the difference in-between the elements of the m-eligibility condition after removing from the dataset x tuples with sensitive value i and y tuples with a different sensitive value. It is straightforward to see that removing tuples with sensitive value i reduces f_i and removing tuples with sensitive value different from i increases f_i. Since we want $f_i \leq 0$, we compute the minimum number of tuples with sensitive value i that need to be removed to make $f_i \leq 0$:

$$\begin{aligned} f_i(x, y) &\leq 0 \\ c - x - \frac{n - x - y}{m} &\leq 0 \\ \frac{mc - n + y}{m - 1} &\leq x, \end{aligned}$$

which implies that at least $\lceil \frac{mc-n+y}{m-1} \rceil$ tuples must be removed. Since $y \geq 0$ we conclude the desired result. □

This result gives a simple lower bound on the difference $|T| - |T'|$ and, as we see next, a method to compute T'.

Proposition 5: Let $T \in \mathbb{R}^{n\times d}$ be a dataset, T' a subset of T and T° a maximal m-eligible subset of T' and let $\{c_1, \ldots, c_l\}$ and $\{c'_1, \ldots, c'_l\}$ be the sensitive values counts of T and T'

TABLE II

TWO ENFORCEMENTS OF 3-ELIGIBILIY OF A DATASET. THE ORIGINAL DATASET COUNTS ARE {5, 3, 1, 1}. THE RESULTS OF ALGORITHMS 1,2,3 INDICATE THAT THE 3-ELIGIBILITY DATASETS MUST HAVE COUNTS {5, 4, 3, 3}, {2, 2, 1, 1} AND {3, 3, 2, 1} FOR COUNTERFEITING, DELETING, AND THE HYBRID APPROACH, RESPECTIVELY. THEN, ENFORCING 3-ELIGIBILITY IS REDUCED TO OBTAINING SUCH COUNTS. TO OBTAIN TABLE II(A) TUPLES WITH ACNE, ADHD. AND HIV HAVE BEEN ADDED. THE CHOICE OF THE QUASI IDENTIFIER AGE WILL BE DETERMINED WHEN THE CLUSTERS ARE CREATED. ALTERNATIVELY, THEY CAN BE SET TO THE VALUES OF ANOTHER TUPLE WITH COMMON SENSITIVE ATTRIBUTE. TO OBTAIN TABLE II(B) THE TUPLES 3, 4, 5, 8 HAVE BEEN DELETED (THE CHOICE OF DELETION IS ARBITRARY). FINALLY, TO GENERATE TABLE II(C) THE TUPLES 4,5 HAVE BEEN DELETED, AND A FAKE TUPLE WITH THE ATTRIBUTE ADHD HAS BEEN INSERTED. A TUPLE WITH THE VALUE *HIV* COULD HAVE BEEN INSERTED INSTEAD

(a) With counterfeits.

ID	AGE	S.D.
1	15	FLU
2	18	FLU
3	18	FLU
4	19	FLU
5	22	FLU
6	15	ACNE
7	17	ACNE
8	19	ACNE
c_1	-	ACNE
9	15	ADHD
c_2	-	ADHD
c_3	-	ADHD
10	18	HIV
c_4	-	HIV
c_5	-	HIV

(b) With deletions.

ID	AGE	S.D.
1	15	FLU
2	18	FLU
6	15	ACNE
7	17	ACNE
9	15	ADHD
10	18	HIV

(c) With both methods.

ID	AGE	S.D.
1	15	FLU
2	18	FLU
3	18	FLU
6	15	ACNE
7	17	ACNE
8	19	ACNE
9	15	ADHD
c_1	-	ADHD
10	18	HIV

respectively, then if for all $i \in [1, l]$ *holds* $c_1 - c'_i \leq \lceil \frac{mc_i - n}{m-1} \rceil$ *then* T° *is also a maximal m-eligible subset of* T.

Proof: Suppose otherwise, that is, that T° is not maximal w.r.t. T, then there exists $\bar{T}$ a maximal m-eligible subset of T such that $|T^\circ| < |\bar{T}|$. Since T° is maximal m-eligible subset of T' we know that $\bar{T} \nsubseteq T'$, in other words there is some sensitive value i such that $c'_i < \bar{c}_i$ where $\bar{c}_i$ is the count of that attribute in $\bar{T}$. However we deduce that $c_i - \bar{c}_i < c_i - c'_i \leq \lceil \frac{mc_i - n}{m-1} \rceil$ contradiction with Proposition 4. □

Proposition 5 allows for a fast method to compute a maximal m-eligible subset since, as we see next, it can be used algorithmically.

Proposition 6: Let $T \in \mathbb{R}^{n \times d}$ *be a dataset with* $l \geq m$ *sensitive values. Algorithm 2 computes a maximal m-eligible subset of* T.

Algorithm 2 Algorithm to Obtain Maximal m-Eligible Subset of a Given Dataset T

```
Data: dataset T
Result: maximal m-eligible subset of T
while T not m-eligible do
    Compute {c_1, ..., c_l} and {r_1, .., r_l} ;
     /* r_i = max(0, ⌈(m c_i − n)/(m − 1)⌉) */
    for i ← 1 to l do
        Remove r_i tuples from T with the ith sensitive
         value;
    end
end
return T;
```

Proof: First, we prove that the algorithm halts and then that the output is the expected result.

With each loop, we are removing tuples from dataset T and checking the m-eligibility of the result. Let us prove that if we do not remove at least one tuple from T then T' is m-eligible. If no element is removed, then $\frac{mc_i - n}{m-1} \leq 0$ which implies $c_i \leq \frac{n}{m}$ for all $i \in [1, l]$ exactly the condition of m-eligibility. Now, since each extra iteration implies the removal of at least one tuple, no more iterations than tuples can be done. We conclude that the algorithm always halts and that the output is m-eligible.

During the execution of the strategy, we have created a finite list $T \supseteq T_1 \supseteq \ldots \supseteq T_k$ verifying the hypothesis of Proposition 5. Since T_k is a maximal m-eligible subset of itself, we deduce, using Proposition 5, that T_k is a maximal m-eligible subset of $T_{k-1}, \ldots, T_1, T$. □

Assume that a data holder is interested in making Table I 3-eligible using partial publication. They run Algorithm 2 obtaining counts $2, 2, 1, 1$ for the attributes FLU, ACNE, ADHD, and HIV, respectively. Since the counts of Table I are $\{5, 3, 1, 1\}$ they remove 3 tuples with attribute FLU and 1 tuple with attribute ACNE matching the number of counts with those retrieved from Algorithm 2. Table II(b) shows the result.

Algorithm 2 leads to a fast way to obtain an m-eligible subset, which can be then used to compute the desired solution. Notice that the non-unicity of solutions should be taken into account (a removed tuple is interchangeable with an existing one if they have the same sensitive value) if a utility metric is being considered in the solution as an objective to minimize (reduce the SSE,...). Such considerations are beyond the scope of this paper.

C. Hybrid m-Invariance Problem

We define the hybrid m-invariance problem as allowing, simultaneously, the removal and insertion of tuples. Consider a dataset with sensitive value counts $\{10, 9, 7, 1\}$ where we seek 3-invariance. Via 3 additions, we obtain $\{10, 9, 7, 4\}$, a minimal 3-eligible superset. Via 3 deletions, we obtain $\{8, 8, 7, 1\}$, a maximal 3-eligible subset. But the frequencies $\{9, 9, 7, 2\}$ are obtained with only one addition and one deletion, strictly

reducing the number of modifications in the dataset while obtaining 3-eligibility.

The hybrid approach has not been extensively tackled in the literature, only in a particular case of [18], so the following results are focused on establishing the basis for future algorithms that need a fast enforcement of m-eligibility with a reduced number of changes on the dataset over the disjoint choice of counterfeits or deletions. Since the desired output is neither a subset nor a superset, we define the similarity of two datasets as follows.

Definition 7: Let $T, T' \in \mathbb{R}^{n\times d}$ be datasets with l distinct sensitive values and respective sensitive value counts $\{c_1, \ldots, c_l\}$ and $\{c'_1, \ldots, c'_l\}$ (possibly 0). We define the distance $d(T, T')$ as

$$d(T, T') = \sum_{i=1}^{l} |c_i - c'_i|,$$

where $|a|$ denotes the absolute value of a. That is the sum of the absolute differences between the counts of each sensitive value on each dataset.

Observe that the defined distance can be conceptualized as the sum of non-redundant[2] additions and deletions performed in the dataset T to obtain T' or vice versa. Now we define the concept of closest m-eligible dataset.

Definition 8: Let $T \in \mathbb{R}^{n\times d}$ be a dataset, we say that $T' \in \mathbb{R}^{j\times d}$ is a closest m-eligible dataset of T if it is m-eligible, $SD(T') \subseteq SD(T)$ and $d(T, T') \leq d(T, T'')$ for any T'' m-eligible dataset with $SD(T'') \subseteq SD(T)$. Where $SD(T)$ is the set of distinct sensitive values of tuples of T.

From Propositions 3 and 5 of the m-invariant problem with counterfeits and partial publication, we obtain an upper bound for the hybrid problem.

Proposition 7: Let $T \in \mathbb{R}^{n\times d}$ be a dataset and T' a closest m-eligible dataset of T then

$$d(T, T') \leq \min(d(T, T_{super}), d(T, T_{sub})) \leq \max(0, cm - n)$$

where T_{super} is a minimal m-eligible superset of T and T_{sub} a maximal m-eligible subset of T.

Proof: From Proposition 3, we know that there exists T_{super} a minimal m-eligible superset of T such that $|T_{super}| - |T| = d(T, T_{super}) = \max(0, cm - n)$ and $SD(T_{super}) \subseteq SD(T)$. Now since T' is closest to T we have $d(T, T') \leq d(T, T_{super}) = \max(0, cm - n)$. Similarly, since $SD(T_{sub}) \subseteq SD(T)$ holds $d(T, T') \leq d(T, T_{sub})$. □

Proposition 7 gives us a reduced search space for the optimal solution; in other words, no more than $cm - n$ modifications will be needed to obtain a closest m-eligible dataset for a non m-eligible dataset T.

Proposition 8: Let T° be a closest m-eligible dataset of a dataset T, and let a, d be the minimal number of necessary additions and deletions, respectively, done to T to obtain T°, then the dataset $\bar{T}^\circ$ made by iteratively adding a times a tuple with minimal frequency sensitive value and iteratively removing d times a tuple with maximal frequency sensitive value is also a closest m-eligible dataset of T.

[2]Redundant means that a tuple has been deleted and a counterfeit with its sensitive value has been added.

Proof: Let c°_1 be the count of the most frequent sensitive value in T° and $\bar{c}^\circ_i$ the ith most frequent sensitive value in $\bar{T}^\circ$. From the construction of $\bar{T}^\circ$ we have $\bar{c}^\circ_1 \leq c^\circ_1$. Since the same number of additions and deletions have been performed on T° and $\bar{T}^\circ$, we know that $|T^\circ| = |\bar{T}^\circ|$. We conclude that $\bar{c}^\circ_i \leq \bar{c}^\circ_1 \leq c^\circ_1 \leq \frac{|T^\circ|}{m} = \frac{|\bar{T}^\circ|}{m}$ which proves the m-eligibility. It is easy to see that $d(T, T^\circ) = a + b = d(T, \bar{T}^\circ)$ completing the proof. □

From Proposition 8 we reduce the search space,at each step, choosing between adding minimal sensitive value frequency tuple or removing a maximal sensitive value frequency tuple. Algorithm 3 does that process greedily.

Proposition 9: Let T be a dataset with $l \geq m$ distinct sensitive values. Algorithm 3 outputs a m-eligible dataset of T.

Algorithm 3 Heuristic Algorithm to Obtain Closest m-Eligible Dataset of a Given Dataset T. Where t_{min} Is a Counterfeit Tuple With Sensitive Value With Minimal Frequency in T, t_{max} Is a Tuple of T With Maximal Frequency Sensitive Value. T_{del} and T_{add} Have Sensitive Value Counts c_i^{del} and c_i^{add} Respectively for $i \in [1, l]$.

Data: dataset T
Result: m-eligible dataset of T
while T *not m-eligible* **do**
  Compute $T_{add} = T \cup \{t_{min}\}$ and $T_{del} = T \setminus \{t_{max}\}$;
  Compute $R_{del} = \sum_{i=1}^{l} \max(0, c_i^{del} - \frac{|T_{del}|}{m})$;
  Compute $R_{add} = \sum_{i=1}^{l} \max(0, c_i^{add} - \frac{|T_{add}|}{m})$;
  if $R_{del} \leq R_{add}$ **then**
    $T = T_{del}$;
  else
    $T = T_{add}$;
  end
end
return T;

Assume that a data holder is interested in making Table I 3-eligible using the hybrid approach. From Proposition 7, they know that they need at most $5 \cdot 3 - 10 = 5$ modifications. They run Algorithm 3 obtaining counts 3, 3, 2, 1 for the attributes FLU, ACNE, ADHD, and HIV, respectively. Since the counts of Table I are $\{5, 3, 1, 1\}$ they remove 2 tuples with attribute FLU and add 1 counterfeit with ACNE matching the number of counts with those retrieved from Algorithm 2. Table II(c) shows the result.

Although we do not have a formal proof of optimality for the Algorithm 3, we have observed that its results are good, outperforming in many cases the non-hybrid approaches (see Figure 2). We expect to develop a provably optimal output algorithm in future research.

Alternatively, a solution can be obtained with Integer Programming. The problem of eligibility with value m can be stated as:

$$\min_X d(T, X) = \min \sum_{i=1}^{l} |c_i - x_i| = \min \sum_{i=1}^{l} z_i$$

subject to the constraints:

$$\begin{aligned} &x_i, z_i \in \mathbb{N} && \forall i \in [1, l] \text{ (integer constraint)} \\ &mx_i - \sum_{i=1}^{l} x_i \leq 0 && \forall i \in [1, l] \text{ (eligibility constraint)} \\ &c_i - x_i \leq z_i, x_i - c_i \leq z_i && \forall i \in [1, l] \ (z_i = |x_i - c_i| \ ) \end{aligned}$$

Which is swiftly solved for any counts and m parameters of the adult dataset and alike (see Section IV). To check the quality of the results of our heuristic algorithm of Proposition 9, we have compared them with the optimal results obtained with the integer optimization problem, and, in all the cases considered, the results have coincided.

D. Practical Implementation Aspects

This section aims to provide practical guidance on how to implement a procedure capable of enforcing m-eligibility on a dataset T. For any approach, the steps to enforce m-eligibility are the following:

Parameter computation. Compute the counts $\{c_1, \ldots, c_n\}$ of the dataset T. Recall that c_i indicates the number of tuples with sensitive value number i in T. Depending on the chosen approach, that is, counterfeiting, deleting or both, use Algorithm 1, 2 or 3 to compute $\{c'_1, \ldots, c'_n\}$ respectively.

Counterfeiting. For each $c'_i > c_i$ add counterfeits until the counts match. There are two main possible ways to add counterfeits. First, generate a tuple with the desired sensitive value, but without quasi-identifiers. These empty values are filled when the clusters are generalized in the m-invariance algorithm. For example, filling them with the centroid of the non-fake tuples of the cluster.[3] Alternatively, generate a tuple with the desired sensitive value and assign quasi-identifiers to it using a synthetic data generator. A simple example of generating quasi-identifiers would be to randomly select them from the tuples with the same sensitive value as the counterfeit.

Deletion. Whenever $c'_i < c_i$, delete tuples with sensitive value number i until the counts match. Any criteria can be used to decide which tuples to delete, but for a generic approach, randomly selecting them is a simple and fast method.

Note that for the hybrid approach, all steps are performed, while for the other methods, only the corresponding step is used.

There are other practical considerations to take into account when running the Algorithms 1,2,3, in particular which sensitive attribute to choose in the case of tights. In the Algorithm 1, when multiple tuples are tight with the least frequency the decision to choose one over the other can be made explicit with some order of preference. Analogously, for the Algorithm 2, to decide the order of the most frequent sensitive attributes in the case of tights can be made explicit with some order of preference. In the Algorithm 3, in the case of a tie, it must be decided whether to prefer counterfeits or deletions.

[3]Clusters always have at least, one non-fake tuple. Otherwise the whole cluster could be removed from the dataset.

IV. Evaluation

We evaluate the different strategies presented in this paper for a real dataset commonly used in data privacy, known as the adult dataset.[4]We divide the evaluation in three experiments: the effectivenes of our methods to enforce m-eligibility with counterfeits and/or deletions; the utility comparison of our method with the state of the art in [14]; comparison with the literature on the number of countefeits to enforce m-eligibility in a dynamic data publishing scenario.

Proposition 10 justifies the use of m strictly below the number of distinct sensitive values in the dataset.

Proposition 10: Let $T \in \mathbb{R}^{k \times d}$ be a dataset with l distinct sensitive values. Then if $m = l$ the closest m-eligible dataset T' has counts $\{\frac{n}{l}, \ldots, \frac{n}{l}\}$ where $|T'| = n$.

Proof: We show it by induction. Let T be a m-eligible dataset with $l = m$ distinct sensitive values and decreasing counts $\{c_1, \ldots, c_l\}$, then

$$c_1 \leq \frac{n}{m} = \frac{n}{l} \Rightarrow c_1 l \leq n$$

it follows that

$$n = \sum_{i=1}^{l} c_i \leq c_1 l \leq n$$

which implies $c_1 = \frac{n}{l}$. Assume that $c_1, \ldots, c_{k-1}$ equal $\frac{n}{l}$ and let us prove for c_k. From induction hypothesis, we have

$$\sum_{i=1}^{l} c_i = (k-1)\frac{n}{l} + \sum_{i=k}^{l} c_i = n \Rightarrow \sum_{i=k}^{l} c_i = \frac{n(l-k+1)}{l}$$

since $c_k \geq c_j$ for all $j \in [k, l]$ we obtain

$$(l-k+1)c_k \geq \sum_{i=k}^{l} c_i = \frac{n(l-k+1)}{l}$$

which yields $c_k \geq \frac{n}{l}$. Now since $\frac{n}{l} = c_1 \geq c_k \geq \frac{n}{l}$ we obtain the desired result. □

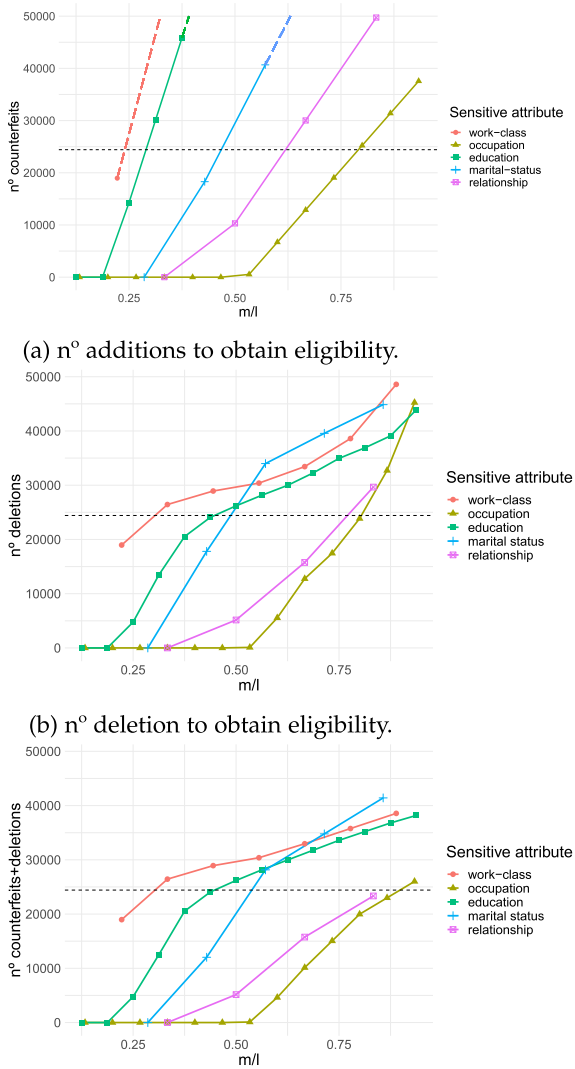
A. Enforcing m-Eligility

For our experimental evaluation, we implemented the Algorithms 1, 2 and 3 of Section III and computed their results. To do so, we considered the complete Adult dataset as a release and studied the necessary number of modifications needed to obtain m-eligibility. We set as sensitive values the columns work-class, occupation, education, marital status, and relationship. The results are shown in Figures 2, 3.

Figure 2 compares the number of modifications over the relation m/l, that is, the eligibility parameter over the number of distinct sensitive values in the dataset.

All figures share the same scale to ease comparisons. Each graph on Figure 2 has a dashed horizontal line corresponding to half the dataset size. Figure 3 is the not cropped version of Figure 2a.

[4]https://www.kaggle.com/uciml/adult-census-income



(a) nº additions to obtain eligibility.

(b) nº deletion to obtain eligibility.

(c) nº modifications to obtain eligibility.

Fig. 2. nº of modifications to obtain m-eligibility via counterfeits, deletions, and the hybrid method. The x-axis is the relation m/l between m the eligibility parameter, and l the number of distinct sensitive values. The dashed horizontal lines indicate when the number of modifications reaches half the size of the dataset.

1) Observations: As we can see from Figure 2 the number of modifications needed to enforce m-eligibility grows as the parameter m increases, which was expected since m-eligibility is a descendent property; that is, m-eligibility implies (m-1)-eligibility. The use of counterfeits over deletions or vice versa is a matter of preference since neither method improves on the other in all cases. The heuristic algorithm of Proposition 9 presents the best results, making the hybrid method the preferred approach if the objective is reducing modifications. Although the hybrid approach is best, for small values of m. the use of any method yields similar results (see Figure 2).

2) Modifications-Accuracy Trade-off: To better understand the impact of adding counterfeits or deleting tuples, we have tested different classification algorithms using the MATLAB Classification Learner. From the Adult dataset, we have selected the attributes: workclass, occupation, educational, marital status, and relationship.

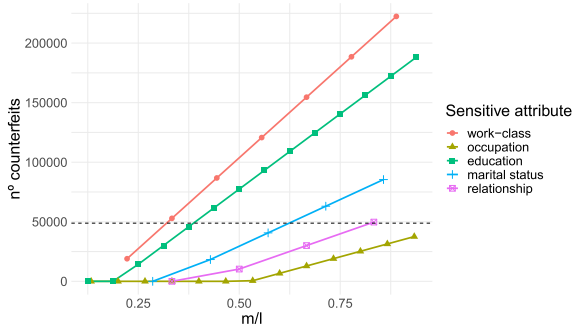


Fig. 3. Not cropped version of Figure 2a. If a value is above the dashed line, more than half the dataset is made of counterfeits.

For the attribute occupation, we have enforced m-eligibility with $m \in [10, 14]$ using counterfeits, deletions, and the hybrid approach. For each case, we have computed the accuracy of the classifier. The accuracy computation is based on partitioning the dataset into 75% training and 25% evaluation. An analogous procedure has been carried out with relationship and $m \in [3, 4, 5]$.

The hybrid procedure to enforce m-eligibility was the following: first compute the counts of the dataset $\{c_1, \ldots, c_n\}$. Use Algorithm 3 to compute $\{c'_1, \ldots, c'_n\}$. Recall that c_i indicates the number of tuples with sensitive attribute number i. For each $c'_i > c_i$ add counterfeits until the counts match. To generate the counterfeits, we added copies of randomly selected tuples with the sought sensitive value. Whenever $c'_i < c_i$ randomly delete tuples with that particular sensitive value until the counts match. The counterfeiting and deleting approaches were analogous to each subcase of the hybrid one but using Algorithms 1, 2 respectively.

The results are shown in Figure 4. We can observe a significant loss in accuracy as the m parameter increases in all cases of the relationship attribute. The hybrid method retains more accuracy as m grows. On the other hand, the occupation attribute evolves better as m increases, except for $m = 14$ where the accuracy changes its behaviour due to the drastic removal of tuples, which makes the training dataset small. The utility of occupation under counterfeiting and the hybrid approach remains stable. That behaviour is expected since counterfeiting increases the relevance of underrepresented attributes in the dataset, reducing model overfitting. For the hybrid approach, the deletion of tuples is mitigating the overrepresentation of the most frequent senstive values, while the counterfeiting is limiting the dataset shrinkage.

From this evaluation, we observe that highly eligible attributes (m high) retain utility after the insertions and deletions of tuples, while less eligible attributes (m small) lose significant amounts of utility as the number of modifications grows.

B. Comparison With State of the Art

The authors of [14] state that, to enforce m-eligibility, it is enough to add at most $m - 1$ counterfeits with randomly selected sensitive values. We find, however, the upper bound of no more than $m - 1$ counterfeits added in such a

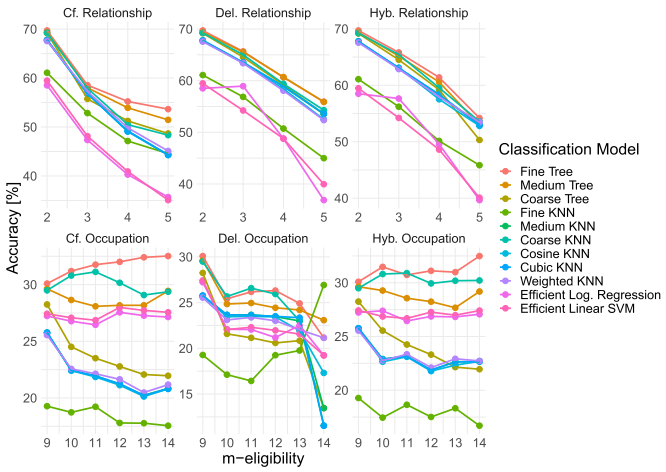


Fig. 4. Accuracy of different classification models for different values of m-eligibility for the attributes relationship and occupation.

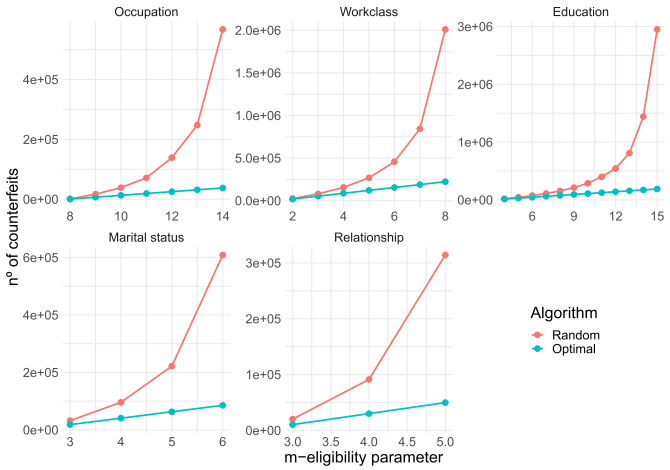


Fig. 5. Comparative number of counterfeits to enforce m-eligibility between the proposed algorithm in [14] and our scheme.

process to be not entirely justified. For example, a dataset with counts $\{10, 2\}$ requires the inclusion of 8 counterfeits to ensure 2-eligibility. Their proposed method of randomly selecting sensitive attributes to enforce m-eligibility appears to be an excessively naive strategy. That being said, their results support a controlled level of counterfeiting. For the sake of comparison, we implemented their proposal and compared it with ours. The results are shown in Figure 5. This evaluation shows that such assumptions lead to a significant increase in the number of counterfeits with respect to our approach.

C. Counterfeits in Continuous Data Publishing

In this experiment, we aim to investigate the effectiveness of our method to enforce m-eligibility with counterfeits in a continuous data publishing scenario in comparison with the proposal of [14]. To achieve this, we have adopted the commonly used approach in the literature of dividing the Adult dataset into multiple releases as performed in [13], [14], and [18]. The initial release, T_1, consists of a randomly selected subset of 20,000 tuples. For each subsequent release T_i is made by randomly deleting 2,000 tuples of T_{i-1} and inserting

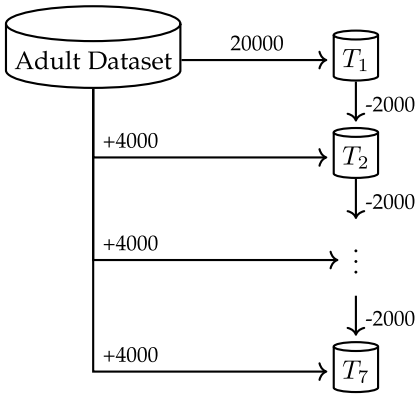


Fig. 6. Dynamic dataset simulation by subdivision of the Adult dataset in multiple releases.

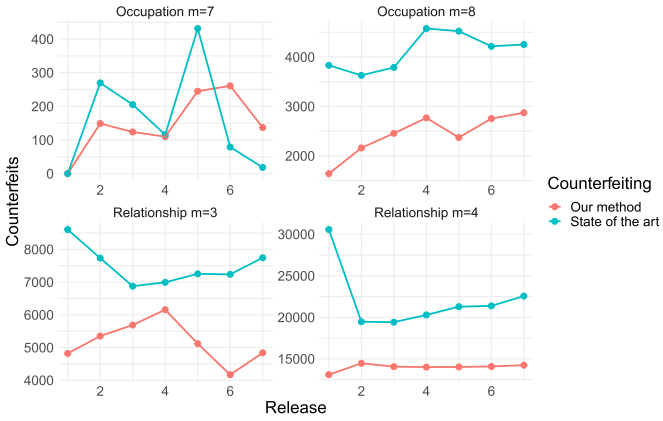


Fig. 7. Comparative number of counterfeits to enforce m-eligility between the proposed algorithm in [14] and our scheme in a continuous data publishing scenario.

4,000 new ones. The process can be visualized in Figure 6. From this process, we have generated seven datasets that depict the progression of a dynamic dataset throughout time. For this experiment, we used occupation and relationship as sensitive attributes. Occupation was chosen imitating existing research as an m-eligible attribute with high m (occupation in Adult dataset is 7-eligible), while relationship was chosen as a point of contrast to occupation (relation in Adult dataset is 2-eligible). The values of m were chosen for the first two values from which it was compulsory the use of counterfeits for subsequent releases. The experiment calculated the number of counterfeits added by each method in each publication. Since the state of the art uses an heuristic which depends on randomness, the results are averages of the results of 5 runs of each experiment. The results are shown in Figure 7.

The results presented in Figure 7 show a general reduction on the average number of counterfeits and a reduction for each individual release except from the last two publications of occupation with $m = 7$. The average number of tuples is reduced by $8\%, 41\%, 31\%$ and 37% for occupation $m = 7$, occupation $m = 8$, relationship $m = 3$ and relationship $m = 4$ respectively.

The two releases for occupation $m = 7$ in which our method was outperformed were preceded by a sudden increase in counterfeits. This suggests that our method identified

a subdivision that used fewer counterfeits but resulted in larger clusters, consequently becoming a burden in subsequent releases. Conversely, using more counterfeits led to smaller clusters that were easily filled with new tuples added to the database. Such phenomenon occurs only once and does not invalidate the other findings. It may be worth investigating as a potential area for future research whether increasing the number of counterfeits used is beneficial in achieving simpler clusters and reducing, on average, the total number of counterfeits.

V. Conclusion and Future Work

This paper gives a formal approach to the problem of enforcing m-eligibility over a dataset. We present upper bounds on the number of necessary modifications to achieve m-eligibility for the m-invariant problem with counterfeit and partial publication. Effective algorithms to compute optimal m-eligible datasets are presented with proofs of their correctness. With this work we are the first proposal to give a optimal approach in the literature to the problem of enforcing m-eligibility on a dataset. We illustrate the novel hybrid problem and give initial results for practical implementations. We end up with an empirical evaluation of our results using a classical dataset in statistical disclosure control. We expect our results to emerge as a new metric from which to compare future empirical evaluations of novel approaches to the m-invariance problem, for example, as a theoretical lower bound on the amount of modification needed in a dataset to achieve m-invariance.

In future work, we expect to extend our results on the hybrid m-invariant problem, proving the optimality of our algorithm or of a new one that we design. We also expect to extend our study on the formalization of the m-invariant problem and its variations. Finally, we plan to study other definitions of closest m-eligible dataset based on different distances that may be of interest, for example, to penalize repeated deletion of the same sensitive value.

References

[1] P. Samarati, "Protecting respondents identities in microdata release," *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 6, pp. 1010–1027, Nov. 2001.

[2] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," in *Proc. 22nd Int. Conf. Data Eng. (ICDE)*, Apr. 2006, p. 24.

[3] N. Li, T. Li, and S. Venkatasubramanian, "T-closeness: Privacy beyond k-anonymity and l-diversity," in *Proc. IEEE 23rd Int. Conf. Data Eng.*, Istanbul, Turkey, Apr. 2007, pp. 106–115.

[4] D. Rebollo-Monedero, J. Forne, and J. Domingo-Ferrer, "From t-closeness-like privacy to postrandomization via information theory," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 11, pp. 1623–1636, Nov. 2010.

[5] C. Dwork, "Differential privacy," in *International Colloquium on Automata, Languages, and Programming*. Cham, Switzerland: Springer, 2006, pp. 1–12.

[6] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *Proc. 42nd ACM Symp. Theory Comput.*, Jun. 2010, pp. 715–724.

[7] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2013.

[8] C. Yao, X. S. Wang, and S. Jajodia, "Checking for k-anonymity violation by views," in *Proc. 31st Int. Conf. Very Large Data Bases*, 2005, pp. 910–921.

[9] E. Shmueli, T. Tassa, R. Wasserstein, B. Shapira, and L. Rokach, "Limiting disclosure of sensitive data in sequential releases of databases," *Inf. Sci.*, vol. 191, pp. 98–127, May 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025511006694

[10] E. Shmueli and T. Tassa, "Privacy by diversity in sequential releases of databases," *Inf. Sci.*, vol. 298, pp. 344–372, Mar. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025514010627

[11] J.-W. Byun, Y. Sohn, E. Bertino, and N. Li, "Secure anonymization for incremental datasets," in *Secure Data Management*. Berlin, Germany: Springer-Verlag, 2006.

[12] X. Xiao and Y. Tao, "M-invariance: Towards privacy preserving re-publication of dynamic datasets," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*. New York, NY, USA: Association for Computing Machinery, Jun. 2007, pp. 689–700, doi: 10.1145/1247480.1247556.

[13] A. Anjum and G. Raschia, "Anonymizing sequential releases under arbitrary updates," in *Proc. Joint EDBT/ICDT Workshops*. New York, NY, USA: ACM, Mar. 2013, pp. 145–154, doi: 10.1145/2457317.2457342.

[14] A. Anjum et al., "τ-safety: A privacy model for sequential publication with arbitrary updates," *Comput. Secur.*, vol. 66, pp. 20–39, May 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404817300019

[15] H. Zhu, H.-B. Liang, L. Zhao, D.-Y. Peng, and L. Xiong, "τ-safe (l, k)-diversity privacy model for sequential publication with high utility," *IEEE Access*, vol. 7, pp. 687–701, 2019.

[16] O. Temuujin, J. Ahn, and D.-H. Im, "Efficient L-diversity algorithm for preserving privacy of dynamically published datasets," *IEEE Access*, vol. 7, pp. 122878–122888, 2019.

[17] F. Amiri, N. YAzdani, A. Shakery, and S.-S. Ho, "Bayesian-based anonymization framework against background knowledge attack in continuous data publishing," *Trans. Data Priv.*, vol. 12, no. 3, pp. 197–225, 2019.

[18] R. Khan et al., "(τ,m) slicedBucket privacy model for sequential anonymization for improving privacy and utility," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 6, Jun. 2022, Art. no. e4130.

[19] A. Blanco-Justicia, D. Sánchez, J. Domingo-Ferrer, and K. Muralidhar, "A critical review on the use (and Misuse) of differential privacy in machine learning," *ACM Comput. Surv.*, vol. 55, no. 8, pp. 1–16, Dec. 2022.

[20] C. Clifton and T. Tassa, "On syntactic anonymity and differential privacy," in *Proc. IEEE 29th Int. Conf. Data Eng. Workshops (ICDEW)*, Apr. 2013, pp. 88–93.

[21] B. C. Leal, I. C. Vidal, F. T. Brito, J. S. Nobre, and J. C. Machado, "δ-doca: Achieving privacy in data streams," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology, J. Garcia-Alfaro, J. Herrera-Joancomart*, J. Garcia-Alfaro, J. Herrera-Joancomartí, G. Livraga, and R. Rios, Eds. Cham, Switzerland: Springer, 2018, pp. 279–295.

[22] J. Parra-Arnau, T. Strufe, and J. Domingo-Ferrer, "Differentially private publication of database streams via hybrid video coding," *Knowl.-Based Syst.*, vol. 247, Jul. 2022, Art. no. 108778. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705122003665

[23] J. Parra-Arnau, D. Rebollo-Monedero, and J. Forné, "Optimal forgery and suppression of ratings for privacy enhancement in recommendation systems," *Entropy*, vol. 16, no. 3, pp. 1586–1631, Mar. 2014. [Online]. Available: https://www.mdpi.com/1099-4300/16/3/1586