EPiC Series in Computing

Volume 106, 2025, Pages 165–175

Proceedings of the 20th Asia Joint
Conference on Information Security

Privacy-Preserving Attribute Domain Reconstruction for Machine Learning

Yuto Tsujimoto[1] and Atsuko Miyaji[2]

[1] Graduate School of Engineering, The University of Osaka, Osaka, Japan
yuto.tsujimoto@cy2seccomm.eng.osaka-u.ac.jp

[2] Graduate School of Engineering The University of Osaka, Suita, Osaka, Japan
miyaji@comm.eng.osaka-u.ac.jp

Abstract

In modern society, the use of personal data is advancing in many fields. However, such data utilization also increases the risk of privacy leakage. Therefore, Differential Privacy (DP) has been proposed as a measure for privacy protection. DP is a privacy-preserving method used when data collectors release data. However, since DP requires the trust of the data collector, Local Differential Privacy (LDP) was proposed as a privacy protection measure that does not rely on third-party trust. LDP assumes that data providers directly perturb their data, thereby protecting privacy leakage from personal data. LDP is useful in machine learning for data privacy and model privacy. However, a challenge with LDP is balancing privacy protection with utility, especially when dealing with high-dimensional data. To address this, a machine learning framework called SUPM has been proposed to satisfy LDP. SUPM consists of three phases: dimensional reduction, training, and testing. In SUPM, before the dimensional reduction phase, users perform attribute domain reconstruction, transforming all categorical and numerical attributes into ordered discrete representations with a domain size of L. However, L is predetermined independently of the actual data characteristics, which can lead to limitations when adapting to diverse datasets. This study proposes improving SUPM by extending the dimensional reduction phase to incorporate both dimensional reduction and attribute domain reconstruction. These enhancements allow for domain size reduction that accounts for the characteristics of each attribute, thereby improving the accuracy of machine learning. The effectiveness of the proposed method is validated through experiments on two datasets: Adult and WDBC.

1 Introduction

Data analysis and machine learning(ML) are advancing across many fields, driving a growing demand for large volumes of data. In particular, the widespread adoption of IoT devices has led to a rapid increase in the volume of the collected data. However, this expansion also raises significant concerns about the risk of personal privacy leakage.

To address these concerns, Differential Privacy (DP) [4] has been proposed. In the DP framework, an aggregator (Agg) perturbs data to protect individual privacy. However, this approach assumes that Agg is trustworthy. If Agg is compromised, there is a risk of data leakage. To mitigate this issue, Local Differential Privacy (LDP) [7] has been introduced. Under LDP, Users perturb their data directly, thereby ensuring privacy without relying on a trusted Agg. Applications that guarantee LDP include frequency estimation [5] and mean estimation [6].

Applications of LDP have also expanded into ML. In ML, there are two types of privacy: data privacy and model privacy. As methods for achieving privacy-preserving machine learning (PPML), one approach is to aggregate the training data at Agg and apply DP during training by perturbing the data. Another method is to apply LDP where Users directly perturb their data and send it to the Agg for training. Using DP protects model privacy but does not protect data privacy. On the other hand, using LDP ensures both data privacy and model privacy, but there is the challenge of performance degradation when dealing with high-dimensional data.

SUPM (Scalable Unified Privacy Mechanism) has been proposed as a framework for LDP-based ML to address this issue. This framework consists of three phases: dimension reduction, learning, and testing. In SUPM, Users perform attribute domain reconstruction by converting all categorical and numerical attributes into ordered discrete representations with domain size L before dimension reduction. This transformation effectively reduces the domain size of each attribute to L, allowing the noise level to be controlled under a fixed privacy budget. Subsequently, dimension reduction is performed while considering the characteristics of the data. However, the important parameter L is predetermined independently of the actual data characteristics, which may limit the adaptability of the method to different datasets.

To overcome this limitation, we propose improving SUPM by extending its dimension reduction phase to incorporate both dimension reduction and attribute domain reconstruction. This extension enables the reduction of domain size in a manner that reflects the distributional characteristics of each attribute, thereby improving ML performance. More specifically, during the dimension reduction phase, we collect class labeled frequency data for each attribute, merge low-frequency values to reconstruct attribute domains, and apply the chi-squared test (χ^2-test) for further aggregation. Furthermore, to evaluate the effectiveness of the proposed framework, we conducted experiments using two datasets—WDBC and Adult—and applied Random Forests (RF) for classification. As a result of the experiments, it was confirmed that the proposed method achieved higher learning accuracy than the conventional SUPM in both mixed datasets (containing categorical and numerical data) and purely numerical datasets.

The remainder of this paper is organized as follows. Section Ⅱ explains the necessary background knowledge pertinent to this study. Section Ⅲ reviews related works. Section Ⅳ presents our proposed method. Section V evaluates the proposed method and performs comparative experiments with existing methods, followed by a discussion on the performance of the proposed method. Finally, Section Ⅵ concludes the study.

2 Preliminary

In this section, we introduce LDP, specific privacy mechanisms such as Randomized Response (RR) and Optimal Unary Encoding (OUE), and the χ^2-statistic. The notation used in this paper is summarized in Table 1.

2.1 Local differential privacy (LDP)

In the local differential privacy model [7], each of n records has data $\mathsf{X}_i = [\mathsf{x}_{\mathsf{i},1}, \cdots, \mathsf{x}_{\mathsf{i},\mathsf{m}-1}, \mathsf{C}_i]$. Each data record contains $m-1$ attributes $A_1, \cdots, A_{m-1}$ and a class label C_i. In this case, Users sends $M(\mathsf{X}_i)$ through a randomized mechanism M to Agg.

Definition 1. *A privacy mechanism M satisfies ϵ-local differential privacy(ϵ-LDP) when M satisfies the following probabilitiy for all possible input data x_1, x_2 and all possible output y*

Table 1: List of Notation and Their Meanings

Notation	Meaning
ϵ-LDP, ϵ	ϵ-local differential privacy, privacy budget
Agg, Users, n	Aggregator, data providers, total number of users
X, X_i	Record set; i-th user's record: $\mathsf{X}_i = [\mathsf{x_{i,1}}, \cdots, \mathsf{x_{i,m-1}}, \mathsf{C_i}]$
$\mathsf{x}_{i,j}$, C_i	j-th attribute and class label of user i
A, A_j, Ω_j, $\lvert\Omega_\mathsf{j}\rvert$, $\mathsf{A}_j[k]$	Attribute set, j-th attributes, domain, its size, k-th element in A_j
C, $\lvert\mathsf{C}\rvert$, c_ℓ	Class label set, its size, and ℓ-th class label
$\mathsf{b}_{i,j}$, $\mathsf{b}_{i,j}[k]$	Binary vector of $\mathsf{x}_{i,j}$ and its k-th bit
$\mathsf{x'}_{i,j}$, $\mathsf{b'}_{i,j}$, $\mathsf{b'}_{i,j}[k]$	Perturbed versions of $\mathsf{x}_{i,j}$, $\mathsf{b}_{i,j}$, and $\mathsf{b}_{i,j}[k]$
d, $\tilde{N}()$	Dimension (number of selected attributes), estimated frequency
L, L_{ini}, L_{opt}	Domain sizes: specified, initial, and optimal
$\mathsf{WA}_j[k]$, $\mathsf{LowA_j[k]}$, $\mathsf{ChiA_j[k]}$	k-th element of A_j after WA, LFM, and $\mathsf{ChiMerge}$, respectively

results

$$Pr[M(x_1) = y] \leq exp(\epsilon) \cdot Pr[M(x_2) = y]$$

Theorem 1 ([8]). *Let $\{M_i\}$ be a sequence of privacy mechanisms, each satisfying ϵ_i-LDP. Then, the composed mechanism $M(D)$ satisfies $(\sum_i \epsilon_i)$-LDP.*

Randomized Response (RR) [6] is a randomized mechanism that satisfies LDP for discrete-valued inputs. Let $\mathsf{x}_{i,j}$ be the input and $\mathsf{x'}_{i,j}$ be the perturbed output, where the domain of attribute A_j is denoted by Ω_j. The mechanism is represented as $\mathsf{x'}_{i,j} \leftarrow \mathsf{RR}(\epsilon, \mathsf{x_{i,j}}, \Omega_\mathsf{j})$ and proceeds as follows

$$\Pr(\mathsf{x'}_{i,j}) = \begin{cases} \frac{\exp(\epsilon)}{\lvert\Omega_\mathsf{j}\rvert - 1 + \exp(\epsilon)}, & \text{if} \quad \mathsf{x'}_{i,j} = \mathsf{x}_{i,j}, \\ \frac{1}{\lvert\Omega_\mathsf{j}\rvert - 1 + \exp(\epsilon)}, & \text{if} \quad \mathsf{x'}_{i,j} \neq \mathsf{x}_{i,j}. \end{cases}$$

Optimized Unary Encoding (OUE) [10] is a frequency estimation mechanism that satisfies LDP. Suppose there are n users, each holding an attribute value $\mathsf{x}_{i,j}$. The OUE mechanism, denoted as $\mathsf{b'}_{i,j} \leftarrow \mathsf{OUE}(\epsilon, \mathsf{x_{i,j}}, \Omega_\mathsf{j})$, consists of two processes: **Encoding** and **Perturbation**, and proceeds as follows.

- **Encoding:** If $\mathsf{x}_{i,j} = \mathsf{A}_j[k]$, the encoding function outputs a unary vector of length $\lvert\Omega_\mathsf{j}\rvert$ with the k-th bit set to 1:

$$\mathsf{Encode}(\mathsf{x_{i,j}}) = [0, \ldots, 0, \underbrace{1}_{\text{k-th}}, 0, \ldots, 0].$$

- **Perturbation:** Each bit of the encoded vector is independently flipped with the following probabilities:

$$\Pr(\mathsf{b'}_{i,j}[k] = 1) = \begin{cases} p = \frac{1}{2} & \text{if } \mathsf{b}_{i,j}[k] = 1, \\ q = \frac{1}{e^\epsilon + 1} & \text{if } \mathsf{b}_{i,j}[k] = 0. \end{cases}$$

Agg collects all perturbed vectors and estimates the frequency of value $\mathsf{A}_j[k]$ using the following formula:

$$\tilde{N}_\mathsf{OUE}(\mathsf{A}_j[k]) = \frac{\sum_{i=1}^{n} 1_{\{\mathsf{b'}_{i,j}[k]=1\}} - n \cdot q}{p - q}.$$

Here, the indicator function $1_{\{\mathsf{b'}_{i,j}[k]=1\}}$ returns 1 if $\mathsf{b'}_{i,j}[k] = 1$, and 0 otherwise. This estimation process is denoted $\tilde{N}_\mathsf{OUE}(\mathsf{A}_j[k]) \leftarrow$ $\text{Estimation}_\mathsf{OUE}(\{\mathsf{b'}_{i,j}\}_{i=1}^{n} \Omega_\mathsf{j})$.

2.2 χ^2-statistic

The χ^2-test is for the test of independence of two events. The χ^2-statistic is calculated as follows.

$$J_{\chi^2}(C, \mathsf{A}_j) = \sum_{\ell=1}^{|C|}\sum_{k=1}^{|\Omega_\mathsf{j}|} \frac{(n_o(\mathsf{c}_\ell, \mathsf{A_j[k]}) - \mathsf{n_e}(\mathsf{c}_\ell, \mathsf{A_j[k]}))^2}{n_e(\mathsf{c}_\ell, \mathsf{A_j[k]})}$$

$$n_o(\mathsf{c}_\ell) = \sum_{\mathsf{k}} \mathsf{n_o}(\mathsf{c}_\ell, \mathsf{A_j[k]}), \quad \mathsf{n_o}(\mathsf{A_j[k]}) = \sum_{\mathsf{c}_\ell} \mathsf{n_o}(\mathsf{c}_\ell, \mathsf{A_j[k]}), \quad \mathsf{n_e}(\mathsf{c}_\ell, \mathsf{A_j[k]}) = \frac{\mathsf{n_o}(\mathsf{c}_\ell) \cdot \mathsf{n_o}(\mathsf{A_j[k]})}{\mathsf{n}}$$

A_j and C are considered dependent if the test statistic exceeds the threshold:

$$J_{\chi^2}(C, \mathsf{A}_j) > \chi^2(\alpha, \Phi)$$

where α is the significance level and Φ is the degrees of freedom.

3 Related works

This section reviews related work, focusing on two LDP-based frameworks: LDP-FS for dimensionality reduction and SUPM for ML.

3.1 LDP-FS framework

LDP-FS [3] is a framework that performs dimension reduction that satisfies LDP. LDP-FS consists of two phases: frequency estimation and dimension reduction, which are called LDP-FS-FE (LDP-FS Feature Estimation) and LDP-FS-DR (LDP-FS Dimension Reduction), respectively, which are denoted in Algorithms 1, 2, and 3. In LDP-FS-FE, frequency estimation for the combinations of class labels and attribute values is performed using OUE. In LDP-FS-DR, dimension reduction is performed based on information gain or χ^2 statistic for each attribute from the frequency distribution. This study adopts the χ^2 statistic.

Algorithm 1 LDP-FS-FE$^{\text{User}}$ [3]

```
Require: privacy budget $\epsilon$, set of attribute domains $\{\Omega_j\}_{j=1}^{m-1}$, set of class labels $C$, $i$-th user record $X_i = [x_{i,1}, \cdots, x_{i,m-1}, C_i]$
Ensure: perturbed data $\{b'_{i,j}\}_{j=1}^{m-1}$
for $x_{i,j} \in X_i$ do
    $b'_{i,j} \leftarrow \mathsf{OUE}(\epsilon/(m-1), [x_{i,j}, C_i], \Omega_j \times C)$
end for
return $\{b'_{i,1}, \cdots, b'_{i,m-1}\}$
```

Algorithm 2 LDP-FS-FE$^{\mathsf{Agg}}$ [3]

```
Require: perturbed data $\{b'_{i,j}\}_{i=1}^{n}$, domain $\Omega_j$ of attribute $A_j$, set of class labels $C$
Ensure: estimated frequency $\{\tilde{N}(A_j[k])\}_{(\ell,k)}$
$\tilde{N}(c_\ell, A_j[k])$
    $\leftarrow$ Estimation$_{\mathsf{OUE}}(\{b'_{i,j}\}_{i=1}^{n}, \Omega_j \times C)$
return $\{\tilde{N}(c_\ell, A_j[k])\}_{(\ell,k)}$
```

3.2 SUPM

Scalable Unified Privacy Mechanism (SUPM) [9] is a privacy-preserving framework in which each data record is directly perturbed by Users. The authors also proposed a privacy mechanism suitable for ML, called WALDP, which is used in SUPM. In PPML that perturbs data, the number of attributes (referred to as dimensions) and the domain size of each attribute

Algorithm 3 LDP-FS-DR [3]

```
Require: dimension $d$, set of estimated frequency $\{\tilde{N}(\mathsf{c}_\ell, \mathsf{A_j[k]})\}_{\mathsf{j},(\ell,\mathsf{k})}$
Ensure: $d$-attribute $\mathsf{A}_{\mathsf{j}_1}, \cdots, \mathsf{Aj_d}$
n ← number of users
for $\mathsf{A}_j \in \mathsf{A}$ do
    $J_{\chi^2}(C, \mathsf{A}_j) \leftarrow \sum_{\ell=1}^{|\mathsf{C}|} \sum_{j=1}^{|\Omega_\mathsf{j}|} \frac{(n_o(\mathsf{c}_\ell, \mathsf{A_j[k]}) - \mathsf{n_e}(\mathsf{c}_\ell, \mathsf{A_j[k]}))^2}{n_e(\mathsf{c}_\ell, \mathsf{A_j[k]})}$
end for
$\{j_1, \cdots, j_d\} \leftarrow \mathrm{argmax}_d(J_{\chi^2}(C, \mathsf{A}_1), \cdots, \mathsf{J}_{\chi^2}(\mathsf{C}, \mathsf{A_m}))$
return $d$-attribute $\mathsf{A}_{\mathsf{j}_1}, \cdots, \mathsf{Aj_d}$
```

affect utility as well as privacy. Specifically, according to Theorem 1, if each attribute is assigned a privacy budget ϵ, then m attributes collectively consume a total privacy budget of $\epsilon \cdot m$. Additionally, since WALDP uses RR, the domain size of each attribute impacts the utility. Therefore, SUPM consists of three phases: dimension reduction (PPDR), model training (PPTR), and testing (PPTEST). In PPDR, the attributes to be used and their domain sizes are determined while preserving privacy.

Here, we describe the privacy mechanism WALDP used in SUPM for arbitrary input data. WALDP consists of three functions: DTO, WAT, and RR. First, DTO transforms categorical data into ordered discrete values. Next, WAT uniformly maps both numerical data and the output of DTO into L ordered discrete values (WA). In particular, numerical data is discretized by evenly dividing its range into L intervals. Finally, given a domain size of L, a perturbation is applied using RR. PPTR is shown in Algorithm 4.

Algorithm 4 Privacy-preserving training (PPTR) [9]

```
Require: dimension $d$, data $\mathsf{X} = \{\mathsf{X_i}\}_{\mathsf{i=1}}^{\mathsf{n}}$, specified domain size $L$, privacy budget $\epsilon$
Ensure: Training model
$\epsilon_{d+1} \leftarrow \epsilon/(d+1)$
for $\mathsf{X}_i \in \mathsf{X}$ do
    $[\mathsf{x_{i,j1}}, \cdots, \mathsf{x_{i,jd}}] \leftarrow$ Sample $d$ from $[\mathsf{x_{i,1}}, \cdots, \mathsf{x_{i,m-1}}]$
    for $\mathsf{x}_{i,j} \in [\mathsf{x_{i,j1}}, \cdots, \mathsf{x_{i,jd}}]$ do
        $(y_{i,j}, \mathsf{WA}_j[1], \cdots, \mathsf{WA}_j[L]) \leftarrow \mathsf{WAT}(\mathsf{x}_{i,j}, \Omega_\mathsf{j}, \mathsf{L})$
        $z_{i,j} \leftarrow \mathsf{RR}(\epsilon, y_{i,j}, \{\mathsf{WA}_j[1], \cdots, \mathsf{WA}_j[L]\})$
    end for
    $z_{i,d+1} \leftarrow \mathsf{RR}(\epsilon_{d+1}, \mathsf{C}_i, \mathsf{C})$
    Send $d+1$-tuple perturbed data to Agg.
end for
Agg collects perturbed data and constructs Training model.
return Training model
```

3.3 Problem of Previous Work

In SUPM, before executing PPDR, attribute domain reconstruction is performed by Users, where all categorical and numerical attributes are transformed into ordered discrete representations with a domain size of L. This transformation effectively reduces the domain size to L, where L allows control over the noise level under the same privacy budget. Subsequently, PPDR is executed while taking the data characteristics into account. However, the important parameter L is predetermined, independently of the actual data characteristics.

In ML, techniques such as attribute domain reconstruction, which implicitly leads to domain size reduction, are often applied to improve accuracy by considering the nature of the data. Therefore, PPDR should be enhanced to support attribute domain reconstruction in addition to dimension reduction.

4 Proposal

This study improves SUPM by extending its PPDR mechanism to perform both dimension reduction and attribute domain reconstruction. These improvements enable domain size reduction tailored to the characteristics of each attribute, thereby enhancing ML model accuracy.

4.1 Characteristics of the proposed method

Our framework consists of three phases: Dimension Reduction and Attribute Domain Reconstruction (DR-ADR), PPTR, and PPTEST. Among them, DR-ADR is newly improved in this work, while the latter two phases remain the same as in SUPM. Note that PPTR uses WALDP, while PPTEST employs either WA or WALDP, depending on whether the environment is trusted or untrusted. In our experiments, only WA is used in PPTEST for simplicity.

We now explain the key idea of extending PPDR in SUPM to DR-ADR. In SUPM, PPDR performs only dimension reduction on data whose attribute domains have already been reconstructed and reduced to a fixed size L by Users. This implies that the domain size of each attribute has already been reduced before perturbation. In contrast, DR-ADR integrates both attribute domain reconstruction and dimension reduction. As a result, the attribute domains are insufficiently reduced at the time of perturbation during data collection for DR-ADR. Therefore, the same randomization mechanism RR used in SUPM is not suitable for DR-ADR, as the resulting noise level may increase significantly when the attribute domain size becomes large. However, the exact domain size is typically unknown in advance. To address this challenge, we adopt OUE, which is well-suited for handling attributes with large and unknown domain sizes. From the viewpoint of efficient privacy budget utilization, each Users perturbs and transmits only d randomly selected attributes. Thus, a key contribution of our approach is that DR-ADR operates effectively on both incomplete and perturbed data.

4.2 Attribute Domain Reconstruction and Dimension Reduction

This subsection describes DR-ADR. This method consists of Algorithm 5, which merges low frequency elements, and Algorithm 7, which performs merging based on the χ^2-test (ChiMerge). For categorical attributes, Algorithm 6 is executed within Algorithm 7.

First, we describe Algorithm 5, which performs Low-Frequency Merge (LFM) for a given threshold T. Let Ω_j^{LFM} denote the reconstructed domain for attribute A_j. Low-frequency elements (below threshold T) are handled as follows: categorical values are grouped together, while numerical ones are merged with the nearest larger neighbor. Note that the LFM process is essential not only for improving the accuracy of ML models, but also for mitigating the influence of noise in low-frequency elements.

Next, Algorithm 6 rearranges the elements of each categorical attribute in descending order based on the proportion of class label c_1 under the two-class setting $[\mathsf{c}_1, \mathsf{c}_2]$. This preprocessing step is necessary because ChiMerge requires all attributes to be ordered in advance.

Next, we describe Algorithm 7, which performs ChiMerge. Let Ω_j^{CM} denote the reconstructed domain for attribute A_j. For attributes with an inherent order, a χ^2-test is conducted between

Algorithm 5 Low-Frequency Merge (LFM)

Require: Threshold T, domain Ω_{j} of attribute A_j, estimated frequency $\{\tilde{N}(\mathsf{c}_\ell, \mathsf{A}_{\mathsf{j}}[\mathsf{k}])\}$
Ensure: Merged domain $\Omega_{\mathsf{j}}^{\mathsf{LFM}}$ of A_j

```
$\tilde{N}(\mathsf{A}_j[k]) = \sum_{l=1}^{|C|} N(c_l, \mathsf{A}_j[k])$
if $\mathsf{A}_j$ is numerical then
    Initialize $\mathsf{LowA_j} \leftarrow []$, $i \leftarrow 1$
    for $k = 1$ to $|\Omega_{\mathsf{j}}|$ do
        Append $\mathsf{A}_j[k]$ to $\mathsf{LowA_j[i]}$
        if $\tilde{N}(\mathsf{A}_j[k]) \geq T$ then
            $i \leftarrow i + 1$
        end if
    end for
else
    Sort $\Omega_{\mathsf{j}}$ by descending $\tilde{N}$
    Let $k$ be the first index where $\tilde{N}(\mathsf{A}_j[k]) < T$
    $\mathsf{LowA_j[1{:}k{-}1]} \leftarrow \mathsf{A_j[1{:}k{-}1]}$,    $\mathsf{LowA_j[k]} \leftarrow \bigcup_{\mathsf{i=k}}^{|\Omega_{\mathsf{j}}|} \mathsf{A_j[i]}$
end if
return $\Omega_{\mathsf{j}}^{\mathsf{LFM}} = [\mathsf{LowA_j}[1], \ldots, \mathsf{LowA_j}[|\mathsf{LowA_j}|]]$
```

Algorithm 6 Class Aware Categorical Order (CACO)

Require: domain Ω_{j} of categorical attribute A_j, set of estimated frequency $\{\tilde{N}(\mathsf{c}_\ell, \mathsf{A}_{\mathsf{j}}[\mathsf{k}])\}_{(\ell,\mathsf{k})}$
Ensure: Ordered domain Ω'_{j} of Categorical AttributeA_j

```
orderd_list $\leftarrow$ Sort $\Omega_{\mathsf{j}}$ in descending order of the proportion of class label $c_1$.
for $i = 1, \cdots, |\Omega_{\mathsf{j}}|$ do
    $\mathsf{A}_j^{ord}[i] \leftarrow$ orderd_list[i]
end for
return $\Omega'_{\mathsf{j}} = \{\mathsf{A}_{\mathsf{j}}^{\mathsf{ord}}[1], \cdots, \mathsf{A}_{\mathsf{j}}^{\mathsf{ord}}[|\Omega'_{\mathsf{j}}|]\}$
```

adjacent elements and the class label. If the test indicates independence, the adjacent elements are merged. This process is repeated until all adjacent element pairs are statistically independent. If all adjacent element pairs are found to be independent, the merging process continues in ascending order of χ^2-values until the domain size falls below L_{opt}. Note that by applying a χ^2-test, the domain size can be compressed from the perspective of data utility.

The whole procedure of $\mathsf{DR\text{-}ADR}$ is given in the below:
Initial Setup

(1) Agg determines the dimension d, the initial domain size L_{ini}, the threshold T and the optimal domain size L_{opt}, where both d and L_{ini} are sent to use[1].

User$_i$ executes the following steps (1) to (3).

(1) Selects d attributes $\mathsf{A_{i1}}, \cdots, \mathsf{A_{id}}$ randomly. $\mathsf{X_i} = [\mathsf{x_{i,i1}}, \cdots, \mathsf{x_{i,id}}, \mathsf{C_i}]$.

(2) Execute Algorithm 1, $\{\mathsf{b}'_{i,i1}, \cdots, \mathsf{b}'_{i,id}\} \longleftarrow$LDP-FS-FE$^{\text{User}}(\epsilon, \{\Omega_{\mathsf{j}}\}_{\mathsf{j=i1}}^{\mathsf{id}}, \mathsf{C}, \mathsf{X_i})$.

(3) Sends $\mathsf{b}'_{i,j1}, \cdots, \mathsf{b}'_{i,jd}$ to Agg.

[1] It is recommended to determine T and L_{opt} based on the estimated frequencies of each attribute.

Algorithm 7 ChiMerge

```
Require: Optimized size $L_{opt}$, domain $\Omega_\mathsf{j}$ of $\mathsf{A}_j$, estimated frequency $\{\tilde{N}(\mathsf{c}_\ell, \mathsf{A_j[k]})\}$
Ensure: Merged domain $\Omega_\mathsf{j}^\mathsf{CM}$ of $\mathsf{A}_j$
if $\mathsf{A}_j$ is categorical then
    $\Omega_\mathsf{j} \leftarrow \mathrm{CACO}(\Omega_\mathsf{j}, \tilde{\mathsf{N}}(\mathsf{c}_\ell, \mathsf{A_j[m]}))$ (Alg. 6)
end if
repeat
    Initialize $\mathsf{ChiA_j} \leftarrow []$, $i \leftarrow 1$
    for $k = 1$ to $|\Omega_\mathsf{j}| - 1$ do
        if $J_{\chi^2}(C, \mathsf{A}_j[k], \mathsf{A}_j[k+1]) < \chi^2(\alpha, |C|)$ then
            Merge: $\mathsf{ChiA_j[i]} \leftarrow \mathsf{ChiA_j[i]} \cup \mathsf{A_j[k]}$
        else
            $\mathsf{ChiA_j[i]} \leftarrow \mathsf{ChiA_j[i]} \cup \mathsf{A_j[k]}$, then $i \leftarrow i + 1$
        end if
    end for
    $\Omega_\mathsf{j} \leftarrow [\mathsf{ChiA_j[1]}, \ldots, \mathsf{ChiA_j[i]}]$
until No pairs satisfy merge condition
$\Omega_\mathsf{j}^\mathsf{CM} \leftarrow \Omega_\mathsf{j}$
while $|\Omega_\mathsf{j}^\mathsf{CM}| > \mathsf{L_{opt}}$ do
    $k_{min} \leftarrow \arg\min J_{\chi^2}(C, \mathsf{A}_j[k], \mathsf{A}_j[k+1])$
    Merge: $\mathsf{ChiA_j[k_{min}]} \leftarrow \mathsf{ChiA_j[k_{min}]} \cup \mathsf{ChiA_j[k_{min}+1]}$
    Update $\Omega_\mathsf{j}^\mathsf{CM}$ and reindex
end while
return $\Omega_\mathsf{j}^\mathsf{CM} = [\mathsf{ChiA_j[1]}, \ldots, \mathsf{ChiA_j}[|\Omega_\mathsf{j}^\mathsf{CM}|]]$
```

Agg executes the following steps after collecting data from Users.

(1) Execute Algorithm 2 to estimate $\{\tilde{N}(\mathsf{c}_\ell, \mathsf{A_j[k]})\}_{(\ell,\mathsf{k})} \longleftarrow$ LDP-FS-FE$^{\mathsf{Agg}}(\{\mathsf{b'}_{i,j}\}_{i=1}^n, \Omega_\mathsf{j}, \mathsf{C})$.

(2) Execute Algorithm 7, $\Omega_\mathsf{j}^\mathsf{LFM} \longleftarrow \mathsf{LFM}(T, \Omega_\mathsf{j}, \{\tilde{N}(\mathsf{c}_\ell, \mathsf{A_j[k]})\}_{(\ell,\mathsf{k})})$, then execute Algorithm 5, $\Omega_\mathsf{j}^\mathsf{CM} \longleftarrow \mathrm{CM}(L_{opt}, \Omega_\mathsf{j}^\mathsf{LFM}, \{\tilde{N}(\mathsf{c}_\ell, \mathsf{A_j[k]})\}_{(\ell,\mathsf{k})})$.

(3) Execute Algorithm 3 to get d attributes $\mathsf{A_{j_1}}, \cdots, \mathsf{A_{j_d}} \longleftarrow$ LDP-FS-DR$(d, \tilde{N}(\mathsf{c}_\ell, \mathsf{ChiA_j[k]})\}_{\mathsf{j},(\ell,\mathsf{k})})$.

5 Feasibility Evaluation

This section evaluates our expansion of SUPM based on DR-ADR. Experiments are conducted on Adult [1] and WDBC [2] datasets, representing mixed and numerical datasets, respectively, where TABLE 2 summarizes the data types, number of records, attributes, and classes for each dataset. We use RF implemented in the 'scikit-learn' machine learning library, with all hyperparameters set to their default values. Evaluation is performed using 10-fold cross-validation, and results

Dataset	Data Type	#Records	#Attr.	#Class
Adult	Numerical & Categorical	32,560	15	2
WDBC	Numerical	569	31	2

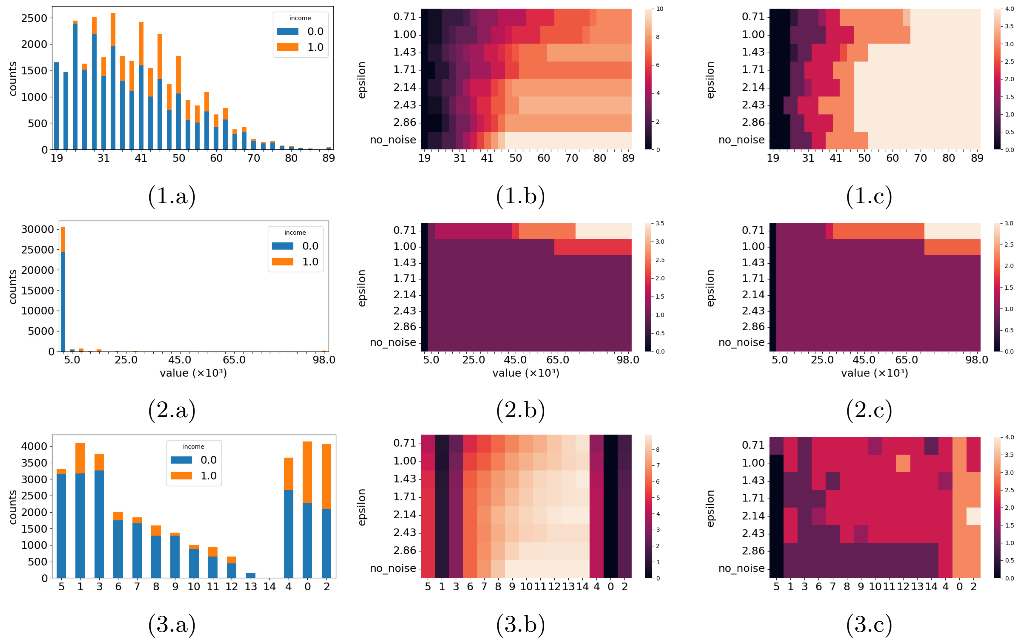
Table 2: Dataset summary

are averaged over three independent runs, each with a different instance of noise addition during training. The balanced accuracy metric is used for performance assessment. The initial parameter settings are as follows: Adult:$(d, L_{ini}, T, L_{opt}) = (7, 30, 0.05 \cdot \#\text{Records}, 5)$, WDBC: $(d, L_{ini}, T, L_{opt}) = (5, 10, 0.05 \cdot \#\text{Records}, 2)$.

5.1 Evaluation of attribute domain reconstruction

We evaluate the change in domain resulting from domain reconstruction through DR-ADR. During the evaluation, we vary ϵ used for perturbation per attribute from 0.71 to 2.85, and analyze how the size of the reconstructed domain changes, particularly in comparison to the unperturbed baseline. Due to space limitations, the evaluation focuses on three attributes from the Adult dataset: two numerical attributes, age and capital gain, and one categorical attribute, occupation. These attributes were also identified as highly influential through dimension reduction. Figure 1 illustrates the reconstruction processes for age, capital gain, and occupation during DR-ADR. Figs. 1.a, 2.a, and 3.a show the initial distribution of each attribute's elements labeled by class 1 or 0 before any processing. Figs. 1.b, 2.b, and 3.b show the reconstruction after LFM. Figs. 1.c, 2.c, and 3.c show the reconstruction after ChiMerge.

These results are presented as heatmaps, where the horizontal axis denotes the elements of each attribute, and the vertical axis indicates the privacy budget ϵ. Each cell's color reflects the reconstructed element to which the original attribute value has been mapped through domain reconstruction. The color gradient corresponds to the order of the elements of the reconstructed domain, with the darkest color representing the first and the lightest color representing the last, up to a maximum of L_{opt}. The reconstructed elements were determined based on the median result from 30 domain reconstruction iterations.



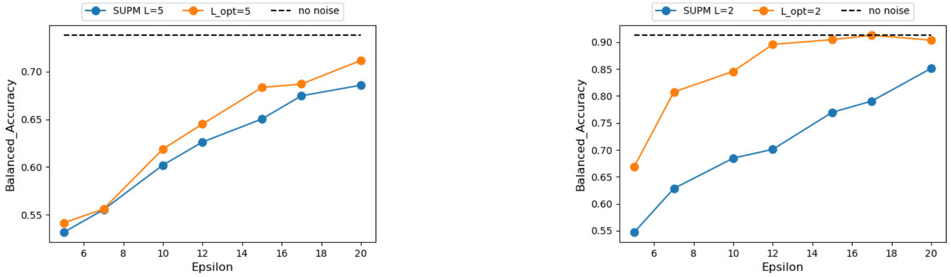
(1.a) (1.b) (1.c)

(2.a) (2.b) (2.c)

(3.a) (3.b) (3.c)

Figure 1: Integration results for attributes (1) age, (2) capital gain, and (3) occupation. Each row shows (a) original, (b) LFM, and (c) ChiMerge, labeled as (1.a)–(3.c).

As shown in Fig. 1.b, when noise is added to the age attribute with ϵ ranging from 0.71



(a) Adult (b) WDBC

Figure 2: Comparison of balanced accuracy between proposed methods and SUPM

to 2.86, the domain size reduction achieved by LFM is greater than that in the case without noise. In contrast, for the ChiMerge, the final domain size remains the same as L_{opt} regardless of whether noise is added or not. In the case of capital gain, as shown in Fig. 2.b, when $\epsilon \geq 1.43$, both LFM and ChiMerge output the same domain size regardless of the presence of noise. This suggests that, compared to age, a larger ϵ is required for capital gain to achieve a comparable level of domain reduction. For categorical attributes such as occupation, shown in Fig. 3.b, adding noise with ϵ ranging from 0.71 to 2.86 results in nearly the same domain size being output by both LFM and ChiMerge, as in the case without noise.

These results suggest that attributes whose domain sizes fall below L_{opt} tend to be more sensitive to noise. Therefore, determining an appropriate value of L_{opt} for each attribute may lead to better overall outcomes.

5.2 Evaluation of learning performance

The learning accuracy between the proposed method and SUPM is compared using the Adult and WDBC datasets. The comparison is made under the condition that the L_{opt} of the proposed method and the L in the WA of SUPM are the same. For the Adult and WDBC datasets, we set $L_{opt} = L = 5, 2$, respectively. The graphs plot balanced accuracy on the vertical axis and $\epsilon = [5, 7, 12, 15, 17, 20]$ on the horizontal axis. The blue line represents the training results of SUPM, while the orange line represents the training results of the proposed method. The black dashed line represents the training results using the reconstructed elements of the proposed method without adding any noise.

First, the training results on the Adult dataset are shown in Fig. 2a. It can be observed that the proposed method achieves better learning accuracy compared to SUPM. While SUPM reaches an accuracy of 68% at $\epsilon = 20$, the proposed method achieves the same accuracy at $\epsilon = 15$. This indicates that for mixed datasets, such as the Adult dataset—which contains both categorical and numerical attributes—reconstructing the attribute domain contributes to improved ML performance. Next, the training results on the WDBC dataset are shown in Fig. 2b. It can be observed that the proposed method achieves higher training accuracy than SUPM when $\epsilon \geq 10$. While SUPM reaches 85% at $\epsilon = 20$, the proposed method achieves approximately the same accuracy at $\epsilon = 10$. These results indicate that even for datasets composed solely of numerical data, such as WDBC, reconstructing the attribute domain can contribute to improving ML performance.

6 Conclusion

In this study, we proposed SUPM based on DR-ADR. Specifically, during the dimension reduction process, we collect class labeled frequency data for each attribute, initially merging low-frequency values and then performing additional merging based on the χ^2-test to reconstruct the attribute domains. In the experiments, we evaluated the reconstruction accuracy of attribute components that were randomly selected and perturbed. We also compared the learning performance achieved using the proposed framework to the performance of the conventional SUPM method. The results demonstrate that the proposed method outperforms SUPM in terms of learning accuracy on both mixed datasets and purely numerical datasets. Furthermore, we performed a visual evaluation of the component reconstruction using heatmaps. As future work, developing methods for quantitatively evaluating the differences in reconstructed domains with and without noise will be important. Additionally, although this study focused on binary classification, extending the proposed method to multi-class classification is an important direction for future research.

References

[1] Adult data set. UCI Machine Learning Repository https://archive.ics.uci.edu/dataset/2/adult.

[2] Breast cancer wisconsin (diagnostic) data set. UCI Machine Learning Repository https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic).

[3] Mina Alishahi, Vahideh Moghtadaiee, and Hojjat Navidan. Add noise to remove noise: Local differential privacy for feature selection. *Computers & Security*, 123:102934, 2022.

[4] Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer, 2006.

[5] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067, 2014.

[6] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Extremal mechanisms for local differential privacy. *Advances in neural information processing systems*, 27, 2014.

[7] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

[8] Frank D McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 19–30, 2009.

[9] Wang Yamatsuki Mimoto Miyaji, Takahashi. Privacy-preserving data analysis without trusted third party. In *2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 710–717. IEEE, 2022.

[10] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. Locally differentially private protocols for frequency estimation. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 729–745, 2017.