Collection and Analysis of Sensitive Data with Privacy Protection by a Distributed Randomized Response Protocol

Faisal Imran
Bahria University, Islamabad
Pakistan

Rosa Meo
University of Torino
Italy

ABSTRACT

The data collected from personal devices is intrinsically private and should be collected through a privacy-guaranteed mechanism. Local differential privacy solves privacy problems by collecting randomized responses from each user, and it does not need to rely on a trusted data aggregator/curator. The proposed approach utilizes the randomized response technique in a novel manner: it guarantees privacy to users during the data collection and simultaneously preserves the high utility of the analysis. It can be seen as a case of synthetic data generation by producing contingency tables (marginals) in a privacy-preserving mechanism. This article describes the proposed randomized response technique and discusses the motivating applications domains. It justifies why it satisfies the property of differential privacy and utility guarantees theoretically and through experimental analysis with excellent results.

CCS CONCEPTS

• **Security and privacy → Data anonymization and sanitization**; *Privacy protections.*

KEYWORDS

Randomized Response; Local Differential Privacy; Contingency Tables; Privacy protection; Distributed computation protocol

ACM Reference Format:
Faisal Imran and Rosa Meo. 2024. Collection and Analysis of Sensitive Data with Privacy Protection by a Distributed Randomized Response Protocol. In *The 39th ACM/SIGAPP Symposium on Applied Computing (SAC '24), April 8–12, 2024, Avila, Spain.* ACM, New York, NY, USA, Article 4, 10 pages. https://doi.org/10.1145/3605098.3636024

1 INTRODUCTION

Data collected from smart devices have become invaluable assets for product designers and application developers. Companies and research centers collect data from end-users and use them to update their knowledge and tailor their products and services.

The problem with massive data collection is that collecting sensitive personal data poses a significant risk to people's privacy rights. To get accurate information from the individuals, the data collection process should enforce robust privacy-preservation mechanisms and consider at the same time the collected data utility. We introduce a novel data collection protocol with randomized responses to achieve data collection with privacy guarantees. The protocol is client-server and occurs in a network/cloud environment where the client represents an end-user or a partner with private data and the server is the data aggregator/curator which is honest-but-curious: it correctly performs the protocol for data exchange and aggregation but might want to know the clients' private data. The risk is of privacy leakage and could occur after an attack with adversarial knowledge or a differential attack. Our proposed method provides strong privacy guarantees combined with a high data utility, as this work shows. We adopt a local differential privacy (LDP) approach rather than the weaker global differential privacy (GDP) approach, where aggregators store the actual data and are a single point of failure and a target for attacks. LDP is stronger because even if adversaries had access to the personal responses, they would still not be able to learn about individuals since the responses are randomized.

Our privacy-preservation randomized response is built on the idea of *randomized response* proposed by Warner in 1965 [26], a data collection technique on sensitive data, where the respondent hesitates to provide a true answer. In Section 1.1 we introduce the principles of the randomized protocol. As discussed in Section 2, surveys generated using randomized responses allow easy computations of correct population statistics while protecting the privacy of the individual and preventing reconstruction attacks. This technique can be used to inject random noise into the answers or the output of a function. Random noise injection protects from the differential attack and is one of the key components of the differential privacy model, the standard de-facto reference for privacy-preserving query answering [8].

Unfortunately, the level of privacy provided by the randomized response [26] degrades if the survey is repeated by the same respondent and does not work for multivariate answers. So, to maintain a strong privacy guarantee with a high utility, we need a better data collection mechanism, as we present in this work.

Finally, we show that the data collected at the aggregator provides a high utility value. The proposed solution gives guarantees, at certain confidence levels, that the statistical dependencies observed in the reconstructed data correspond to the true ones. The proposed solution relies on a combination of sophisticated machine learning modeling and numerical optimization with hypothesis tests, as we show in Section 6.

In Section 2 we compare the randomized response protocol with the related work. We present two improved versions of the randomized protocol and discuss its properties in Section 4 and in Section 5 we discuss its convergence. In Section 6 we present its robustness in preserving statistical associations among variables.

SAC '24, April 8–12, 2024, Avila, Spain

ACM ISBN 979-8-4007-0243-3/24/04.
https://doi.org/10.1145/3605098.3636024

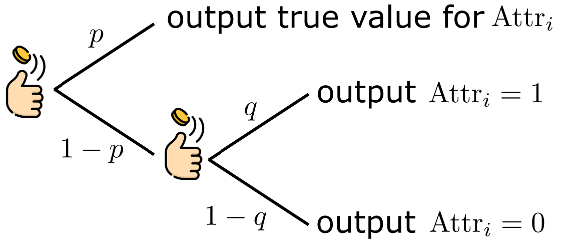

p
output true value for Attr_i
$1-p$
q
output $\mathrm{Attr}_i = 1$
$1-q$
output $\mathrm{Attr}_i = 0$

Figure 1: The flow of the randomized protocol and two flips of coins, with a binary attribute Att$_i$

In Section 7 we show the excellent experimental results on three real use cases in comparison with Laplace noise and 2-step MLE MonteCarlo simulation.

1.1 The principle of the randomised response protocol

The survey respondent is asked to flip two fair coins in secret; if the first coin is "Head", the respondent is asked to flip a second coin whose outcome will determine if the answer is "Yes" or "No". Figure 1 shows the flow of the randomization protocol. It is simple to see that in a situation where both "Yes" and "No" answers can be denied (flipping two fair coins), the true number of "Yes" answers can be accurately estimated by $2(q - 0.25)$, where q is the proportion of "Yes" responses. The unknown probability of a successful event studied on a population, represented by a random variable, is correctly and even more efficiently inferred by the randomized protocol in Figure 1 if parameter q is close to the true probability of the successful event. This observation led to our proposal that the parameter q be adjusted to its true value as the protocol evolves. The natural and more general setting is where each client has multiple attributes, and the server is interested in learning their joint distribution after observing only a sample of the population. Knowledge of the joint distribution opens the way to powerful descriptive and predictive analytical models, such as statistical inference models and Bayesian networks. In the adopted local differential privacy (LDP) approach, in the proposed protocol, a respondent's private data is generated (possibly modified by the randomised protocol itself) after selecting subsets of attributes. These values are communicated to one or more aggregators in a distributed environment. As a final step, the aggregator receiving the randomised data has the task of calculating contingency tables (CTs) with the frequencies of the observed values.

Thanks to the protocol properties, we demonstrate that it is possible to reconstruct the true joint probability of the attributes from the possibly noisy values communicated by the individuals. The transmitted values do not need to correspond to the true ones for each individual, in virtue of the deniability property of the protocol. Moreover, the randomization is local to the individual users, and there is no need for a different, trusted organization to perform the randomizer or add a verified amount of noise.

Often, Machine Learning algorithms rely on low-order marginals as a building block and compute accurate approximations by the Maximum Likelihood principle and vine-copulas [4, 17]. Our proposed method generates low-dimensional tables on m attributes (*m-way*). We can generate even lower-dimensional CTs, with $k < m$ from these m-dimensional ones by further marginalizations. We call these further CTs higher-level. We propose to apply linear programming to the *m-way* CTs to make consistent the marginals of the higher level ones (*k-way*). This approach is also followed by [2].

2 RELATED WORK

Privacy-preserving data statistics are often considered in a centralized setting in which the data is perturbed by adding random noise from Laplace distribution or applying the Exponential mechanism. These perturbation techniques reduce the risk for an individual to be identified [9, 11]. However, in this classical approach, with true data in the database, individual privacy is still not guaranteed from external attacks or internal adversaries (e.g., eavesdropping). Our approach is based instead on the decentralized setting with local differential privacy. Each client randomizes its true values using a local randomization mechanism. The noisy values are then sent on the network to the aggregator without the need to be protected and then aggregated to produce the desired statistics.

A multitude of approaches exist: they combine randomized response techniques [26] to create sophisticated noise addition mechanisms [10, 12, 18, 21, 24]. Google RAPPOR [10] collects users' data in a private setting, where the responses are mapped to a Bloom filter using a hash function. RAPPOR implements a two-step randomization technique: first, by mapping the user string onto a Bloom filter using a hash function, and second by flipping each bit in the Bloom filter with given probabilities.

Apple implements privacy in their iOS to collect user statistics through users sketching [3, 24]. Microsoft collects users' app statistics privately using rounding and memorization techniques [7]. Wang *et al* [25] proposed an optimization technique with asymmetric randomization response and hashing function. Kairouz *et al* [14] propose the optimal generalizations of randomized responses to estimate the frequency of a single categorical attribute.

3 PRELIMINARIES

We consider a setting where each client owns a set of attributes. The centralized server collects these attributes in a privacy-preserving manner and releases the joint distribution of their values.

3.1 Notations

We consider a dataset D with d attributes $X = (A_1, A_2, \cdots, A_d)$. We use $\mathcal{V}_i$ to denote the domain of the values of A_i and v_{ij} to represent a possible value in $\mathcal{V}_i$. A subset of attributes in X is denoted by S_i. A contingency table (CT) involving the attributes in S_i is called T_{S_i}. We use $T_{r,c}$ to represent the attributes values (entry points) in a CT with the values for a subset of attributes r in S_i as rows and another subset c as columns in the CT. We use $T_{r,c}[v]$ to represent the cell value of that CT at those entry points. $|T_{S_i}|$ denotes the cardinality of the CT. The probability of an attribute value v_{ij} is denoted by $p(v_{ij})$. Each row in D represents a single user or client u. The notations are summarized in Table 2.

Example 3.1. Database D in Table 1a has six attributes: **A** = {adult, old}; **R** = {big, small}; **E** = {high, uni}; **O** = {emp, self}; **S** = {M, F}; and **T** = {car, train, other}., It is aggregated with `count` function applied

(a) Dataset consists of 6 attributes (Age, Region, Education, Occupation, Sex, Transportation)

	A	R	E	O	S	T
1	adult	big	high	emp	M	car
2	adult	big	high	emp	F	train
3	adult	small	high	emp	F	other
4	adult	small	high	self	F	car
5	old	big	uni	emp	F	train
6	old	small	uni	self	M	other
7	old	small	uni	self	M	train
8	old	small	high	self	M	train

(b) CT with 2 attributes T_{AT}

V	$T_{\text{RE}}[v]$
(big, high)	2
(big, uni)	1
(small, high)	3
(small, uni)	2

(c) Marginal table for Region

V	$T_{\mathbf{R}}[v]$
(big, $*$)	3
(small,$*$)	5

Table 1: Example of a dataset, CT, and the marginals

Meaning	Notation
Attribute name	A_i
single value of A_i	v_{ij}
attribute set	S_i
contingency table on S_i	T_{S_i}
contingency table entry point	$T_{r,c}$
contingency table cell value	$T_{r,c}[v]$

Table 2: Summary of notations

to subsets of their values. Table 1b shows a CT over a set of two attributes. Table 1c shows a marginalization.

3.2 Differential Privacy

The current de facto standard of privacy protection is differential Privacy [8, 9]. It is interpreted as a statistical property that compares the output of a query on the database when the individual is included in the database with the alternative without the individual. To protect the individual's privacy, noise is added either on the data or in the query mechanism ($\mathcal{M}$) that answers requests on the data. The privacy guarantee of the randomization mechanism is quantified by the parameter of the privacy budget ϵ that controls how different are the probabilities that the query returns the same output in the two databases, differing for a single individual.

Definition 3.2. (Differential Privacy [9]) A randomization mechanism $\mathcal{M}$ is $\epsilon-$ differentially private if for any two neighbouring databases $D^1 \in \mathbb{N}^{|X|}$ and $D^2 \in \mathbb{N}^{|X|}$ that differ for a single entry, and any subset $\mathcal{R}$ of the output of $\mathcal{M}$,

$$\frac{P[\mathcal{M}(D^1) \in \mathcal{R}]}{P[\mathcal{M}(D^2) \in \mathcal{R}]} \leq \exp^{\epsilon} \quad (1)$$

where the probability is taken over the randomness of $\mathcal{M}$. In our case, the mechanism (or query) $\mathcal{M}(D)$ is represented with a collection of CTs T_{S_i} returned by the randomized response protocol on D, with S_i one of the subsets of attributes in X.

3.2.1 Utility Goal of Our Randomization Method. The utility of our randomization protocol stems from the possibility of reconstructing k-way CTs whose values are close to the true ones T_{S_i}. Given a reconstructed noisy k-way CT T'_{S_i}, we consider three error measures to evaluate the performance of the proposed randomization method (the lower the better).

In our first experiment, we calculate the χ^2 independence testing between the true and the noisy CT.

The second is the ℓ_2 distance between T'_{S_i} and T_{S_i}, in which the CTs are viewed as vectors of 2^k elements. In the context of the randomization method, the error distance can be regarded as a random variable due to its dependency on the noise introduced by the method itself. *Expected Squared Error (ESE)* is the expected value of the square of the error distance, an aggregation of squared errors across individual cells. ESE is frequently employed to assess the utility of a given method.

The third method is the Jensen-Shannon divergence between T'_{S_i} and T_{S_i}, both normalized by dividing each cell value with the sum of the cells (so that the probability mass is 1). It is natural to apply Kullback-Leibler divergence between T'_{S_i} and T_{S_i}, since $D_{KL}(T_{S_i}||T'_{S_i})$ measures the information lost when T'_{S_i} is used to approximate T_{S_i}. However, $D_{KL}(T_{S_i}||T'_{S_i})$ can be undefined when $T_{S_i}[v_i] = 0$ or $T_{S_i}[v_i] \neq 0$ for some v_i. Thus, we use Jensen-Shannon divergence [19], which is a symmetrized and smoothed version, given as:

$$D_{JS}(T_{S_i}||T'_{S_i}) = \frac{1}{2}D_{KL}(T_{S_i}||Q) + \frac{1}{2}(T'_{S_i}||Q) \quad (2)$$

where $Q = \frac{T_{S_i}+T'_{S_i}}{2}$ and $D_{KL}(T_{S_i}||T'_{S_i}) = \sum_{ij} log\left(\frac{T_{S_i}[v]}{T'_{S_i}[v]}\right) T_{S_i}[v]$

4 RANDOMIZED RESPONSE BLOCK AGGREGATION

This section presents the proposed method **Randomized Response Block Aggregation (RRBA)**.

Before querying the end-users, the aggregator generates disjoint subsets S_i of k attributes taken from the original set of d attributes to form a certain number of *size-k* CTs called *views* **V**. The subsets **V** form separate views on the sample population. The union of the subsets in views should be as large as possible. The aggregator arbitrarily selects a combination of views from the possible ones for querying the single client whose attribute values could be randomized in his/her response. This arbitrary selection that changes for each client provides an extra layer of protection in the randomization protocol. These views privately publish a synopsis of the entire dataset. Successively, the server reconstructs any higher-order marginals from these views. To show how to assign attributes into views, we show a running example with the number of attributes $d = 6$ and attributes: $\{A, R, E, O, S, T\}$. With $k = 2$, we have three combinations of 2 distinct attributes per view. This is the list of the alternative views for each individual.

$$V_1 = \{AR, EO, ST\}, V_2 = \{AE, RS, OT\}, V_3 = \{AT, RE, OS\},$$
$$V_4 = \{AO, RT, ES\}, V_5 = \{AS, RO, ET\}$$

We have a total of five views ($V_1, V_2, \cdots, V_5$) to cover all the possible combinations of attributes. Now suppose we have $d = 5$ then,

$$V_1 = \{AR, EO\}, V_2 = \{RE, OS\}, V_3 = \{AE, RS\},$$
$$V_4 = \{AO, ES\}, V_5 = \{AS, RO\}$$

For the first view V_1, we left out the attribute S, for the second one A, and so on. Just a single one because it could not be paired with another one without allowing repetition of one attribute in the same view. If the first alternative is selected, the view is formed by the two combinations of attributes $\{AR\}$ and $\{EO\}$. Both combinations are considered for the same individual. The attributes in any combination are randomized together, thus keeping intact possible statistical dependencies between them.

This step is necessary because the randomization protocol must not generate multiple times randomized values of the same attribute from the same individual. Indeed, if an eavesdropper observed the multiple outcomes of the same attribute, even if combined with others, it would observe with higher probability the true values, thus distinguishing them from the randomized ones. An alternative solution would be to maintain the value generated for each attribute in the internal memory of the clients' devices. However, this solution is not always possible for all devices and would require a large memory size for data sets with many attributes. Observe that any pair of attributes is assigned in at least one view. Since independent noise is added through these views, marginalizing two different CTs from these views to obtain the same marginals would likely give different results. To make consistent the marginalisations generated from these views, we perform the constraint optimization technique discussed in Section 4.3.

We have two different versions of our protocol. In the first version, the aggregator selects arbitrary combinations from a view $V_i \in \mathbf{V}$. The aggregator sends this combination as a question, such as: "What is your age and which region do you belong to". Clients' responses are collected in the randomized mechanism to ensure that either randomly selected responses or true responses are collected by the aggregator. In the second version, we divide the clients into groups called blocks B. We then perform randomized data aggregation in parallel within the blocks. Once all responses are collected, the aggregator moves to the next block. Before the next block is processed, the probability distribution used to generate random responses is updated to be closer to the true one. This is done by updating the probability distribution with the responses collected in the previous block.

4.1 Fundamentals of the Randomized Response Block Aggregation Method

Given a set of views $\mathbf{V}$, the aggregator arbitrarily selects a view $V_i \in \mathbf{V}$ comprised of multiple combinations of attributes. On all these combinations of attributes, the responses are collected from the client in the ϵ−LDP setting. The aggregator initializes for each combination of attributes in V_i the joint distribution by a CT whose cell values are initialized with the uniform distribution, i.e., $\frac{1}{|T_{r,c}|}$.

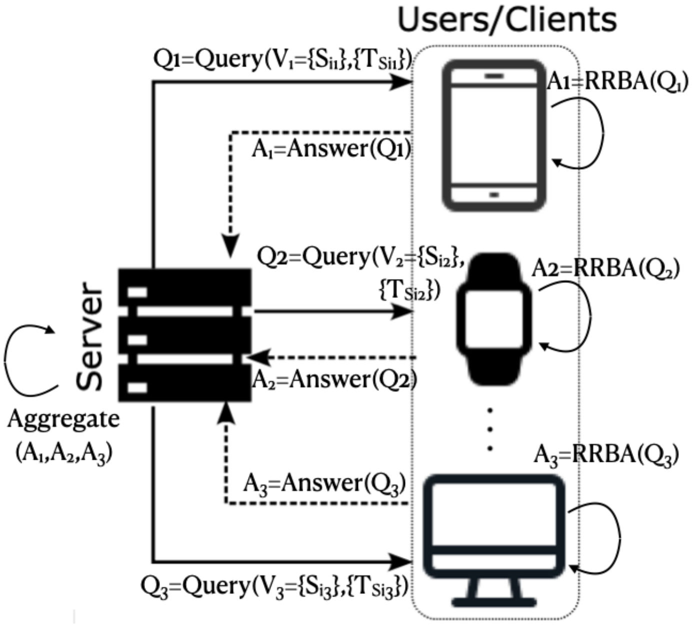


Figure 2: Overview of communication between data aggregator and mobile clients to generate noisy CT on views V_i

```
Algorithm 1: Randomized response on single client
Input: Set of attributes $X$, probability (first coin is head) $p$
Output: Noisy table $T'_{r,c}$
Function Aggregator($X$):
    make views $V = V_1, V_2, \cdots, V_d$ ;
    randomly generate the views and check that the
      combinations of attributes are not repeated in the
      views ;
    generate uniform distribution in $T_{C_i}$ of all views;
    while exists a client that has not yet communicated do
        select arbitrary view $V_i \in \mathbf{V}$ ;
        o ← Client($T_{r,c}$, query($r, c$)) ;      /* Call client
          procedure */
        reconstruct $T'_{r,c}$ from $o$ and $T_{r,c}$ using equation 4 ;
        update: $T_{r,c} \leftarrow T'_{r,c}$
    end
Function Client($T_{r,c}$, $query(r, c)$):
    Sample a Bernoulli variable B ;
    if B = "Head" then
        Respond true value $v \in T_{r,c}$
    end
    else
        Respond a fake value using equation 3, with a
          random probability $q$ drawn between $[0, 1]$ ;
    end
```

Upon receiving a question from the aggregator on each set of attributes in a view V_i, the client responds according to the outcomes of the random variables, drawn with the predefined probabilities p and q. Probability p is tunable to adjust the privacy and utility of the responses. Probability q is randomly drawn between 0 and 1: it represents the value of the cumulative joint probability function of the attributes values. It makes each combination of categorical attribute values represented in the multivariate CT correspond to a continuous probability value that these categorical values are observed. Monte Carlo sampling exploits it to draw first the probability value and then returns the corresponding combination of attribute categorical values.

The random variable p is implemented by drawing a random value, between 0 and 1, uniformly distributed. This random variable controls if the user communicates the true values of the combination of queried attributes. If the random value is above p, "fake" values are communicated to the aggregator, according to the second random variable q, drawn between $[0, 1]$. The outcome of this latter random variable corresponds to one of the cells (denoted by v) in the CT by their probability. In turn, each cell corresponds to some combinations of the categories of the attributes. The variable q for emitting a "fake" value is a type of Monte Carlo sampling from the given discrete joint distribution $T_{r,c}$, such that:

$$\sum_j T_{r,c}[v_j] = 1$$

and $0 \leq q \leq 1$ then

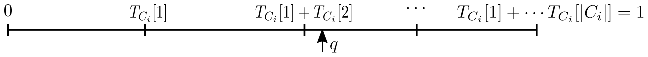

0
$T_{C_i}[1]$
$T_{C_i}[1] + T_{C_i}[2]$
$\cdots$
$T_{C_i}[1] + \cdots T_{C_i}[|C_i|] = 1$
q

Figure 3: The intervals of the cumulative probability distribution function that make each probability interval correspond to a cell l of the CT T_{C_i}

$$\sum_{j=1}^{l-1} T_{r,c}[v_j] \leq q < \sum_{j=1}^{l} T_{r,c}[v_j] \tag{3}$$

This "fake" response is emitted in such a way as to disclose a *"controlled"* amount of information about the client's true attribute values. Hence, limiting the aggregator's ability to learn with confidence the true values of the client, Monte Carlo sampling improves the utility of our protocol by emitting combinations of values based on their probability as stored in the CT.

Once the aggregator receives a response from the client, it reconstructs a noisy CT $T'_{r,c}$ using the CT $T_{r,c}$ used for the previous client. For the reconstruction it applies equation 4:

$$T'_{r,c}[v] = \left(\frac{o_l}{n} - T_{r,c}[v] \cdot (1-p)\right) \cdot \frac{1}{p} \tag{4}$$

where o_l is the observed number of clients who communicated those attributes values represented by v and n is the total number of clients. The above equation is justified by the fact that o_l is the number of observed responses corresponding to the same cell l in CT $T_{r,c}[v]$ and the responses come from the execution of the randomization protocol: they are outcomes of the true probability distribution with probability p (the first coin gives "Head") and are random outcomes controlled by the probability distribution in $T_{r,c}$ (the first coin is "Tail" with probability $(1-p)$).

The aggregator updates its table $T_{r,c} = T'_{r,c}$ and sends this updated table $T_{r,c}$ to the next client u_{i+1} for the next randomized response. The next client now uses the updated probabilities $T_{r,c}$ in the Monte Carlo sampling.

Observe that the aggregator has no access to the client's true values. Thus, the proposed mechanism ensures local differential privacy. Algorithm 1 outlines the complete working of our protocol, including both client-side and aggregator procedures.

4.1.1 The improved version of the protocol. The second improved version of the randomized response data aggregation works similarly to the first version, except now, the clients are divided into groups called blocks B. The aggregator now executes the collection of responses from each client in parallel within the blocks. The aggregator aggregates the responses from the blocks and updates the CT using equation 4, where now n is the block size. When all the responses are collected, the aggregator publishes the noisy CT $T_{r,c}$ to the server. The overview of our proposed randomized responses protocol and the communication between the aggregator and its end-users is shown in Figure 2. It shows that multiple combinations of attributes $\{S_i\}$ contained within a view V_i are sent to clients together with the corresponding noisy CTs T_{S_i} for the execution of the randomized protocol. The server receives the responses and aggregates them. The block size n is defined by the data aggregator/curator. With the algorithm of Section 5 and the experiments in Section 7.1, we demonstrate the selection of the optimal block size, which leads to the convergence of the estimated probabilities in the CTs to the true probabilities.

4.2 Differential Privacy of Randomized Response Block Aggregation

The proposed mechanism aims to minimize the risk of disclosure to ensure a strong privacy guarantee while satisfying the strict concept of $\epsilon-$LDP. It promises strong privacy despite the amount of background knowledge of an adversary. Hence, with a substantial amount of auxiliary information, an adversary could not confidently identify the true responses from the clients. Since a single report from the client contributes to the count measure of a single cell v in $T_S = T_{r,c}$, the privacy level ϵ is independent of the number of cells in $T_{r,c}$. Hence, we need to prove the satisfaction of $\epsilon-$differential privacy for only a single CT cell.

Theorem 4.1. *The proposed randomized response protocol satisfies $\epsilon-$ differential privacy, with: $\epsilon \geq ln\left(\frac{1}{1-p}\right)$ where p is the probability that the first coin gives "Head".*

Proof. Let us consider two CTs $T^1_{r,c} \in$ and $T^2_{r,c} \in$, realizations of the CT T_S on the attribute subset S, that come respectively from two databases D_1 and D_2 that differ for a single record. Let $T_{r,c}$ be the reported combination of attribute values returned by the proposed randomization protocol from the record u_i that differs in the two databases. It corresponds to the cell of the CT $T_{r,c}[v]$. According to the definition of differential privacy [9] we need to consider when the proposed randomization protocol works as a randomized mechanism and transforms the input databases D_1 and D_2 into the same CT T_S, regardless of having in input the database D_1 or D_2. Let us assume that q is the probability that a combination of attribute categorical values corresponding to the cell $T_{r,c}[v]$ occurs in the database. According to the proposed randomized protocol, these attribute values are reported if the first coin draws "Head" and if they are the true values: this occurs with probability pq. In addition, the first coin could give instead "Tail", but the emitted values are drawn as a consequence of the second random event: this overall event occurs with probability $(1-p)q$. On the other database, with a different record u_i', the only possibility that the randomized protocol returns the same value as above is that the first coin gives "Tail" and the second random event returns those values corresponding to cell $T_{r,c}[v]$, and this occurs with probability $(1-p)q$. Mathematically, we obtain:

$$\frac{P[\mathcal{M}(D_1) = T_S]}{P[\mathcal{M}(D_2) = T_S]} \leq \exp^{\epsilon}$$

$$\frac{P[\mathcal{M}(u_i) = T_{r,c}]}{P[\mathcal{M}(u_i') = T_{r,c}]} \leq \exp^{\epsilon}$$

$$\frac{pq + (1-p)q}{(1-p)q} \leq \exp^{\epsilon}$$

$$\Rightarrow \epsilon \geq ln\left(\frac{1}{1-p}\right) \tag{5}$$

From the opposite side, when D_1 does not contain u_i but D_2 does, we obtain $\epsilon \geq ln\,(1-p)$ that is always satisfied with $0 \geq p \geq 1$. □

The equation 5 shows the relationship between the parameter ϵ (the privacy budget that controls the privacy amount) and the parameter p of the randomized response protocol (the fraction of times clients respond trustfully). Notice that it does not depend on q, the probability of the emitted value; thus, it is valid regardless of the response.

Decreasing p makes ϵ arbitrarily low, the desired situation since it allows the randomized protocol to make stronger privacy preservation. As a drawback, with low p the convergence of the reconstruction of the true probability distribution from the observed responses becomes slower, as we will see from the experimental results. On the opposite side, as p grows, it increases the risk that true values are emitted too frequently, and ϵ cannot be reduced to small values. The relationship between ϵ and p is shown in Figure 4.

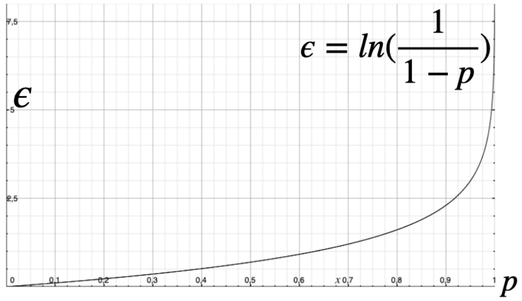


Figure 4: Graph of the relationship between the protocol parameter p of the first coin "Head" and the privacy budget ϵ

We drive the values of p using the Equation 6:

$$e^{\epsilon} \geq \frac{1}{1-p} \Rightarrow \epsilon \geq \ln(\frac{1}{1-p}) \Rightarrow p \leq 1 - e^{-\epsilon} \tag{6}$$

and identify at what value of p we see convergence in the observed values from the randomization protocol, using a given block size. In the experiments of Section 7.2 we discussed the effect of different values of p on the convergence at different block sizes.

4.3 Consistency between Noisy Tables

Given a set of noisy views, the server wishes to release marginals of some attributes with a privacy guarantee. Since independent noise is added in each attribute combination within a view, aggregating marginals from the different views will create inconsistencies in the marginals of the common attributes.

Suppose we have T_S, where $S' \subseteq S \subseteq X$ are subsets of the attributes. We use the symbol $\mathbf{T}_{S' \leftarrow S[v]}$ to denote the marginal over S' calculated from T_S by aggregating the corresponding entries.

Consistency between views. We consider the marginal CTs $\mathbf{T}_S^1$ and $\mathbf{T}_S^2$ with a common attribute A coming from two noisy views $A \in V_i$ and $A \in V_j$. The two marginal CTs $\mathbf{T}_S^1$ and $\mathbf{T}_S^2$ are consistent if and only if the marginal table over the common attributes in $V_i \cap V_j$ reconstructed from view V_i is the same as reconstructed from view V_j.

Given a set of views in $\mathbf{V}$ and a set of attributes S, we can compute k-way marginals $\mathbf{T}_S$. When at least one view $V_i \in \mathbf{V}$ includes all the attributes in S, i.e $S \subseteq V_i$, we can reconstruct $\mathbf{T}_S$ by summing over the corresponding entries of T_S in T_{V_i}, that is using $\mathbf{T}_{S \leftarrow V_i}$. However, when we have multiple views V_i such that $S \subseteq V_i$, we need to perform a linear optimization technique to return consistent marginals from all the views V_i that cover all the attributes in S. When $S \cap V_i$ contains j attributes, then T_{S_i} provides exactly 2^j constraints on the cells for T_S. We can extract all these linear constraints from all the views to generate an under-specified system of equations.

One can utilize the $\ell_1 - norm$ optimization technique discussed in [2] to reconstruct the marginals in T_S. This technique does not create a unique solution, and linear programming has no preference among different solutions. So we employ another constraint optimization technique $\ell_2 - norm$ (least square solution). We will follow the quadratic programming approach similar to the work in [20] to solve the under-specified system of equations as a minimizing problem:

$$\min_{v} \quad \sum_{v \in T_S} \mathbf{T}_S[v]^2$$

$$s.t., \quad {}_{v \in \mathbf{T}_S} \mathbf{T}_S[v] \geqslant 0$$

$$_{V_i \in \mathbf{V} \;\; v' \in S \cap V_i} T_S[v] = \mathbf{T}_S[v']$$

It has been shown that this is a quadratic optimization problem, and we solved it with convex optimization approaches [6].

5 CONVERGENCE AND BLOCK SIZE ESTIMATION

We show that the probabilities generated from $T_{r,c}$ converge to the true probabilities after we used the protocol aggregating the observations sent from the individuals in a certain number of blocks of size n. The value $T_{r,c}[v]^{B_k}$ allows to compute the probability of a cell of the CT $T_{r,c}$ created by running the randomized protocol on the users of block B_k, where we use the superscript B_k to denote the block number. The estimated probability at round k sending outcomes from block B_k is done with $T_{r,c}[v]^{B_k}$. The estimation of the probabilities, done by the protocol, converges to the true probabilities by oscillating around the true value within a tolerance interval related to the error in observing a Bernoulli variable. The tolerance interval is given by the width of the confidence interval of the Bernoulli variable, with the success probability equal to the true but unknown value v, and the interval width is estimated as follows.

If the approximation of the Bernoulli distribution with the Normal distribution holds (i.e., if $v > 5$, with $v = T_{r,c}[v]$ and v/n the probability estimation), we can use a symmetrical interval, where the confidence interval size can be estimated by $2z_{1-\alpha/2\sigma}$ with $\sigma = \sqrt{\frac{(v/n\cdot(1-v/n))}{n}}$ the standard deviation of the Bernoulli distribution. Otherwise, maximum likelihood confidence intervals must be used with the log odds. We set the α confidence level equal to the standard values, e.g., 0.05 or 0.01. This latter means that the estimated probability value will remain within the confidence interval with a probability equal to $1 - \alpha$.

The convergence algorithm proceeds as follows:

(1) **Initialization with** $k = 0$**:** $T_{r,c}[v]^{B_0} = \frac{1}{|T_{r,c}|}$
(2) **At iteration** $k = k + 1$**:** run the RRBA protocol and estimate $T_{r,c}[v]^{B_k}$ from equation 4
(3) **Repeat:** step 2 until convergence, i.e. $\left|T_{r,c}[v]^{B_k} - T_{r,c}[v]^{B_{k-1}}\right| < \delta^*$, for some $\delta^* > 0$
(4) **Return:** $T_{r,c}[v] = \frac{T_{r,c}[v]^{B_k}+T_{r,c}[v]^{B_{k-1}}}{2}$ which is the average between the two consecutive observed values in consecutive blocks.

where δ^* is the size of the confidence interval.

6 TESTING FOR ASSOCIATION

One of the first questions posed while dealing with categorical attributes is whether they are independent. The test of independence χ^2 [1] is one of the most common statistical tests with categorical attributes that mainly compares the observed frequencies of the combined attribute values with the estimated frequencies, assuming the attribute are independent. This latter estimation is obtained by the **Maximum Likelihood Estimation**, denoted by $\widehat{m}_{i,j}$ for cell (i, j) in the CT $T_{r,c}$. To perform a similar test of independence for a noisy version of the table, we need to determine an estimation for $\widehat{m}$ where we do not have access to the true cell counts in the CT. Suppose we only have access to the noisy cell values in $T_{r,c}$, where noise is added in each cell independently, for instance, using our randomization protocol. To find the best estimates for $\widehat{m}$ given the noisy cells we perform a two-step MLE calculation similar to the work of [15, 16].

In a two-step MLE procedure, we first find the most likely CT $\widehat{T}_{r,c}$ given the noisy table $T_{r,c}$, and in the second step, we calculate MLE given a table of counts $\widehat{T}_{r,c}$. For the first step, we need to minimize $\left\|T_{r,c} - \widehat{T}_{r,c}\right\|$ subject to $\sum_j \widehat{T}_{r,c}[v] = n$ and $\widehat{T}_{r,c}[v] \geqslant 0$. Note that if we add independent noise in each cell of a table $T_{r,c}$, the above optimization problem gives multiple solutions. The ℓ_1 norm in our objective function in Equation 7 is not strongly convex, which means it has an optimal solution but may not be unique and sensitive to an initial guess. To overcome this problem, we add a strongly convex function in the objective function:

Datasets	Records	Attributes	Categories
Survey	500	6	14
Alarm	10,000	37	103
Child	10,000	20	60

Table 3: Summary of the selected datasets

$$\begin{aligned} \underset{\widehat{T}_{r,c}}{\text{minimize}} \quad & \gamma \left\| T_{r,c} - \widehat{T}_{r,c}\right\|_1 + (1-\gamma)\left\|T_{r,c}-\widehat{T}_{r,c}\right\|_2^2 \\ \text{subject to} \quad & \sum_{ij} \widehat{T}_{r,c}[v] = n, \\ & \widehat{T}_{r,c}[v] \geqslant 0. \end{aligned} \tag{7}$$

where γ is a mixing parameter in the range $[0, 1]$. The above objective function is in the form of *elastic net regularize* [27] function proposed by [16]. The solution of this objective function converges to the solution provided by the ℓ_1 norm when γ is sufficiently large. For the test of independence, in the two-step **MLE** calculation, if any cell value in $\widehat{T}_{r,c}[v] < 5$, we follow the commonly chosen rule of thumb to Accept H_0.

7 EXPERIMENTS

For experimental reproducibility, we use three publicly available datasets (for Bayesian networks) [1]: *Survey* [22], *Alarm* [5], and *Child* [23]. They vary in the number of instances and attributes as described in the overview of Table 3. All attributes are discrete.

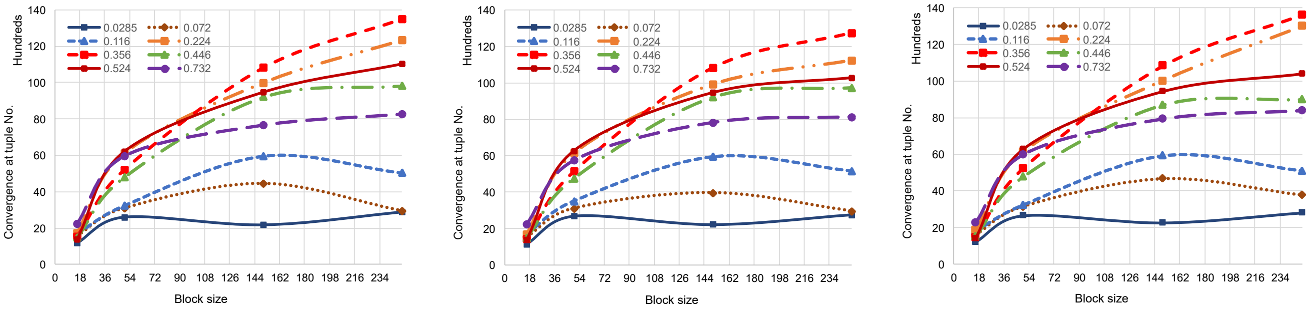
7.1 Monte Carlo simulation: Convergence of the randomization protocol

To perform a test of convergence of the second version of the proposed randomized response protocol, we test with any of the values of the attributes whose probability of occurrence is in $\{0.0285, 0.072, 0.116, 0.224, 0.356, 0.446, 0.524, \text{ and } 0.732\}$ and let vary the block size $s = \{18, 50, 150, \text{ and } 250\}$. We perform 40 trials on 200 blocks on each probability value and block size. We average the number of tuples emitted when the condition holds $\left|T_{r,c}[v]^{B_k} - T_{rc}[v]^{B_{k-1}}\right| < \delta^*$, and remains valid throughout the blocks.

7.2 Convergence Results

We perform the test of convergence in the datasets (`Survey`, `Alarm`, and `Child`). We plot the results of the experiments in Figure 5, where the x-axis represents the block size, and the y-axis shows the number of tuples emitted when the convergence is reached. The behavior of convergence of the proposed randomized method is similar in all three datasets. A smaller block size makes it easier to achieve early convergence at both low and high probability values. Hence, it is sufficient to have a block size equal to the dimension of the CT.

[1] Available at https://www.bnlearn.com/bnrepository/



(a) Survey dataset

(b) Alarm dataset

(c) Child dataset

Figure 5: Convergence in probabilities $P(A_i = v_{ij}) = \{0.072, 0.116, 0.224, 0.356, 0.446, 0.524, 0.732\}$ and block size $s = \{18, 50, 150, 250\}$ on *Survey, Alarm* and *Child*

We perform similar experiments on convergence with different values of p (the probability the first coin is "Head"). Due to the computational limitations, we focused on a few probability values to analyze convergence on the varying value of p. The selected probabilities of the true attribute values $P(v_{ij}) = \{0.072, 0.116, 0.356, 0.446\}$, block size $s = \{18, 50, 150, 250\}$ and the probability of the first coin "Head" $p = \{0.009, 0.048, 0.095, 0.139, 0.221\}$.

At $p = \{0.009, 0.0480, 0.095\}$ none of the processes of reconstruction of the probabilities converges at given block size s. Instead, at $p = 0.1390$, the reconstruction process of the higher probability values $P(v_{ij})$ (set at 0.356 and 0.446) converges with higher block sizes, i.e., 150 and 250. At $p = 0.221$, the reconstruction process of all the probability values converges with the higher block sizes, as shown in the graph of Figure 6. In the graph, there is no convergence for all the probability values when the block size is 18 and 50. If we increase the block size, the reconstruction processes converge for all the probabilities $P(v_{ij})$. A similar behavior is observed at $p = 0.295$. The results of Figure 6 show that if we have a smaller value of p we must select a larger block size so that the reconstruction process of the probabilities converges; if we select a higher value of p we see the convergence at smaller block sizes, as shown in Figure 5.

7.3 Monte Carlo Simulation: Test of Independence

We want to test if the addition of noise destroys independence (null hypothesis rejected). We generate a k−way noisy CT $T_{r,c}$ using the proposed randomization technique. We calculated the estimations $\widehat{m}_{i,j}$ of the cells using the two-step MLE procedure. Using these estimates, we sample $l > 1/\alpha$ many CTs (where α is the significance level, 0.05). We then add noise to these sampled tables using the randomized response protocol. Using the same two-step MLE calculation, we obtain l different $\widehat{\chi}^2$ values from these sampled noisy tables. We rank these statistics by choosing $\lceil (l+1)(1-\alpha) \rceil$ as threshold $\widehat{\vartheta}^{\alpha}$. If $\widehat{\chi}^2 > \widehat{\vartheta}^{\alpha}$ we Reject H_0 else, we Accept H_0. If at any point the two-step MLE calculation outputs any cell count < 5 then we Accept H_0.

7.3.1 Significance Results. We show how the tests of Independence perform on real-world data when H_0 is both rejected or accepted.

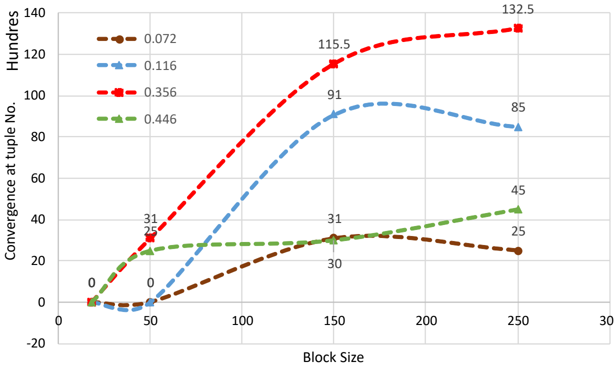


Figure 6: Convergence in probabilities $P(v_{ij}) = \{0.072, 0.116, 0.224, 0.356, 0.446, 0.524, 0.732\}$ and block size $s = \{18, 50, 150, 250\}$ at probability p (first coin is Head) = $\{0.009, 0.048, 0.095, 0.139, 0.221\}$

We set $\alpha = 0.05$ (significance level), $\gamma = 0.01$ as the parameter in the two-step MLE, and the privacy budget $\epsilon = 0.25$ in all our tests.

We perform the independence testing on 2−way, 3−way, and 4−way CTs with binary attributes. Note that the independence tests can also be performed on arbitrary $r \times c$ noisy CTs generated by the proposed method. Notice that as soon as the number of values increases, the proposed protocol is more robust than the others and succeeds in the tests a higher number of times.

In the above experiments with Laplace distribution, since it does not provide critical values, we used the true values of the attributes as the values for the comparison with noisy data (they are known in advance). If this was not possible, one could also find the critical values of simulated data using **R** package "CompQuadForm".

Table 4 compares the performance of the proposed method with state-of-the-art competitors (Laplace noise and MCIndep [13]) using a confusion matrix. We perform 100 trials for H_0 rejected and 100 trials for H_0 accepted with CTs generated parametrically. The accuracy of the proposed method is excellent (96.5%, 94%, and 93.5%) in all k−way CTs. These results are better than both Laplace and MCIndep methods. Further, our block randomization protocol is

Table 4: Comparison of independence tests on $k-$way CTs ($k = 2, 3,$ and 4) with Laplace noise, MCIndep (Monte Carlo independence testing), RRBA on 100 trials with $\alpha = 0.05$, $p = 0.5$, $\epsilon = 0.25$

		2-way		3-way		4-way	
		Reject H_0	Accept H_0	Reject H_0	Accept H_0	Reject H_0	Accept H_0
Laplace noise	Reject H_0	68	32	55	45	50	50
	Accept H_0	35	65	41	59	41	59
	Accuracy	66.5%		57%		54.5%	
MCIndep [13]	Reject H_0	94	6	94	6	92	8
	Accept H_0	5	95	7	93	9	91
	Accuracy	94.5%		93.5%		91.5%	
RRBR	Reject H_0	**96**	4	**94**	6	**93**	7
	Accept H_0	3	**97**	6	**94**	6	**94**
	Accuracy	**96.5%**		**94%**		**93.5%**	

Table 5: Comparison of ℓ_2 and Jensen-Shannon distance between noisy and original CTs (Survey and Alarm); noise is added using randomization protocol and Laplace noise with parameters $p = 0.4$, $\epsilon = 0.35$, $n = 8000$, and Block size $B = 250$

ℓ_2 distance						
	Survey			Alarm		
	2-way	*3-way*	*4-way*	*2-way*	*3-way*	*4-way*
RRBA	68.27	**123.89**	**140.10**	**90.36**		
Laplace	**59.58**	307.588	715.05	117.42		
Jensen-Shannon distance						
	Survey			Alarm		
	2-way	*3-way*	*4-way*	*2-way*	*3-way*	*4-way*
RRBA	**0.0104**	**0.0142**	**0.0577**	**0.0073**		
Laplace	0.0142	0.1025	0.1434	0.0153		

Table 6: Comparison of ℓ_2 and Jensen-Shannon distance between noisy and original CTs (Survey and Alarm); noise is added using the randomization protocol and Laplace noise with parameters: $p = 0.5$, $\epsilon = 0.5$, $n = 8000$, and Block size $B = 250$.

ℓ_2 distance						
	Survey			Alarm		
	2-way	*3-way*	*4-way*	*2-way*	*3-way*	*4-way*
RRBA	**71.81**	**100.70**	**111.26**	**59.58**	**102.22**	**111.15**
Laplace	109.60	154.02	372.63	61.69	162.62	427.98
Jensen-Shannon distance						
	Survey			Alarm		
	2-way	*3-way*	*4-way*	*2-way*	*3-way*	*4-way*
RRBA	**0.0107**	**0.0129**	**0.0304**	**0.0074**	**0.0156**	**0.0380**
Laplace	0.0142	0.0582	0.1417	0.0118	0.0633	0.1580

robust even in sparse data, where contingency cells often have very low or zero count values. On the contrary, Laplace and MCIndep do not produce valid results in these extreme situations, which can be a killer application.

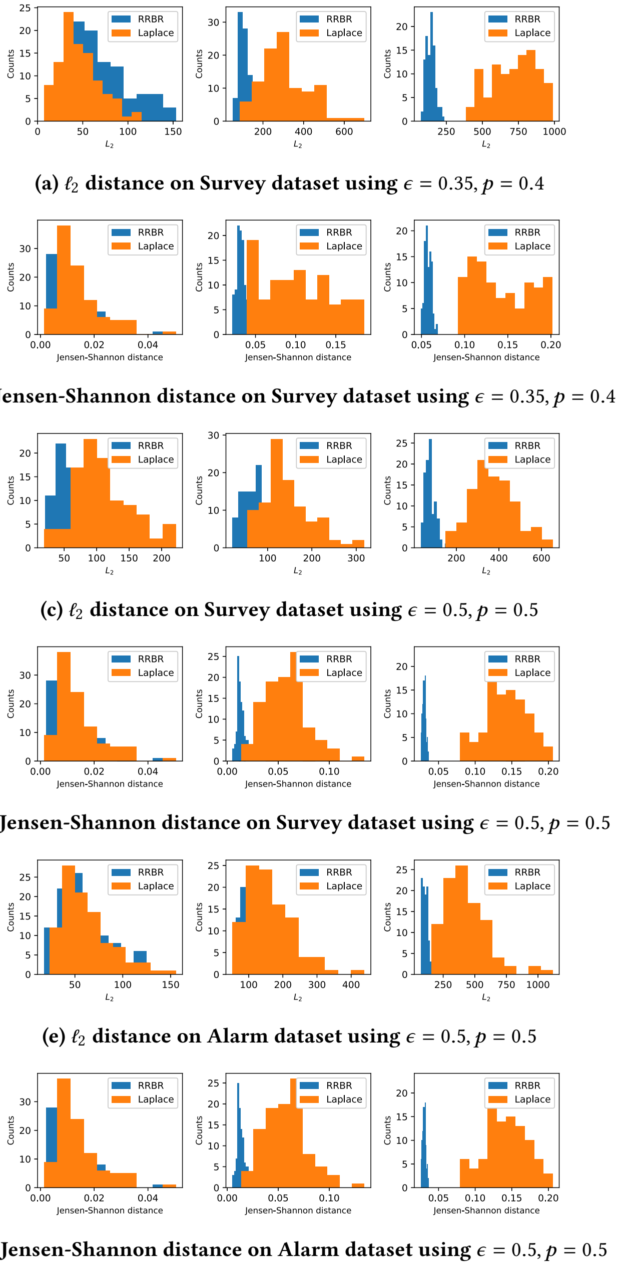
7.4 Performance using ℓ_2 norm and Jensen-Shannon distance

We evaluate the performance of the proposed randomization protocol using ℓ_2 norm. For evaluation purposes, we use the noisy 2, 3, 4-way CTs that are compared with the ground truth. The Laplace noise is drawn from $Lap\,(0, b)$ with zero mean and a scale that depends on the privacy budget $b = \frac{2|T_{r,c}|}{\epsilon}$. We performed 100 trials on *Survey* and *Alarm* datasets and reported the average performance in Table 5 and Table 6. Figure 7 shows the distribution of the performance metrics.

From Table 5 and Table 6, the proposed randomization protocol has the lowest average ℓ_2 distance on Survey and Child datasets. The proposed protocol has the lowest average distance on higher dimensional tables when the noise variance is large $\epsilon = 0.35$ and $p = 0.4$. When $\epsilon = 0.5$ and $p = 0.5$ our protocol wins on all CTs. These tables also conclude that our proposed randomization model compared with Laplace noise has a lower distance on the Jensen-Shannon distance scale (a lower scale means the noisy distribution is similar to the ground truth). The results from the experiments (performance metric using independence test, ℓ_2 distance, and Jensen-Shannon divergences) show that the proposed randomization method wins over Laplace noise. The proposed privacy protocol maximizes utility in the released CTs while ensuring ϵ-differential privacy.

8 CONCLUSION

In this work, we systematically explore the problem of collecting and analyzing data from smart devices under $\epsilon-$local differential privacy, in which the aggregator/server is honest-but-curious, has access to randomized responses from users, and reconstructs statistical models based on perturbed data. The server computes accurate statistics from the released joint distributions. With the experiments, we showed that our protocol achieves high utility in reconstructing the probabilities of attribute values, committing a low error bound. In future work, we will use the hash function to store CTs to reduce computation and communication overheads.



(a) ℓ_2 distance on Survey dataset using $\epsilon = 0.35, p = 0.4$

(b) Jensen-Shannon distance on Survey dataset using $\epsilon = 0.35, p = 0.4$

(c) ℓ_2 distance on Survey dataset using $\epsilon = 0.5, p = 0.5$

(d) Jensen-Shannon distance on Survey dataset using $\epsilon = 0.5, p = 0.5$

(e) ℓ_2 distance on Alarm dataset using $\epsilon = 0.5, p = 0.5$

(f) Jensen-Shannon distance on Alarm dataset using $\epsilon = 0.5, p = 0.5$

Figure 7: Randomization and Laplace noise performance histograms on the noisy $2-$way table (left), $3-$way (middle), $4-$way (right) with ℓ_2 and Jensen-Shannon distance. The average performance is in Table 5 and Table 6. Block size $B = 250$, records $n = 8000$

REFERENCES

[1] Alan Agresti. 2018. *An introduction to categorical data analysis*. John Wiley & Sons.

[2] Boaz Barak, Kamalika Chaudhuri, Cynthia Dwork, Satyen Kale, Frank McSherry, and Kunal Talwar. 2007. Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 273–282.

[3] Raef Bassily, Kobbi Nissim, Uri Stemmer, and Abhradeep Thakurta. 2017. Practical locally private heavy hitters. *arXiv preprint arXiv:1707.04982* (2017).

[4] T. Bedford and R. M. Cooke. 2001. Probability density decomposition for conditionally dependent random variables modeled by vines. *Annals of Mathematics and Artificial intelligence* (2001), 245–268.

[5] Ingo A Beinlich, Henri Jacques Suermondt, R Martin Chavez, and Gregory F Cooper. 1989. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *AIME 89*. Springer, 247–256.

[6] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

[7] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. 2017. Collecting telemetry data privately. *arXiv preprint arXiv:1712.01524* (2017).

[8] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*. Springer, 265–284.

[9] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science* 9, 3-4 (2014), 211–407.

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

[11] Liyue Fan and Hongxia Jin. 2015. A practical framework for privacy-preserving data analytics. In *Proceedings of the 24th International Conference on World Wide Web*. 311–321.

[12] Giulia Fanti, Vasyl Pihur, and Úlfar Erlingsson. 2016. Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries. *Proceedings on Privacy Enhancing Technologies* 2016, 3 (2016), 41–61.

[13] Marco Gaboardi, Hyun Lim, Ryan Rogers, and Salil Vadhan. 2016. Differentially private chi-squared hypothesis testing: Goodness of fit and independence testing. In *International conference on machine learning*. PMLR, 2111–2120.

[14] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. 2014. Differentially private multi-party computation: Optimality of non-interactive randomized response. *arXiv preprint arXiv:1407.1546* (2014).

[15] Vishesh Karwa, Aleksandra Slavković, et al. 2016. Inference using noisy degrees: Differentially private β-model and synthetic graphs. *Annals of statistics* 44, 1 (2016), 87–112.

[16] Jaewoo Lee, Yue Wang, and Daniel Kifer. 2015. Maximum likelihood postprocessing for differential privacy under consistency constraints. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 635–644.

[17] Todd K Leen, Thomas G Dietterich, and Volker Tresp. 2001. Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO.

[18] Bo-Cheng Lin, Shang-Hong Wu, Yao-Tung Tsou, and Yennun Huang. 2018. PPDCA: Privacy-preserving crowdsensing data collection and analysis with randomized response. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 1–6.

[19] Jianhua Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information theory* 37, 1 (1991), 145–151.

[20] Wahbeh Qardaji, Weining Yang, and Ninghui Li. 2014. Priview: practical differentially private release of marginal contingency tables. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. 1435–1446.

[21] Zhan Qin, Yin Yang, Ting Yu, Issa Khalil, Xiaokui Xiao, and Kui Ren. 2016. Heavy hitter estimation over set-valued data with local differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 192–203.

[22] Marco Scutari and Jean-Baptiste Denis. 2014. *Bayesian networks: with examples in R*. CRC press.

[23] David J Spiegelhalter, A Philip Dawid, Steffen L Lauritzen, and Robert G Cowell. 1993. Bayesian analysis in expert systems. *Statistical science* (1993), 219–247.

[24] Differentially Privacy Team. 2017. Learning with Privacy at Scale Differential Privacy Team. "https://docs-assets.developer.apple.com/ml-research/papers/learning-with-privacy-at-scale.pdf"

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

[26] Stanley L Warner. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *J. Amer. Statist. Assoc.* 60, 309 (1965), 63–69.

[27] Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)* 67, 2 (2005), 301–320.