ENHANCING UTILITY IN THE WATCHDOG PRIVACY MECHANISM

Mohammad Amin Zarrabian* *Ni Ding** *Parastoo Sadeghi*† *Thierry Rakotoarivelo*‡

* College of Engineering and Computer Science, Australian National University, Canberra, Australia.
*School of Computing and Information Systems, University of Melbourne, Melbourne, Australia.
†School of Engineering and Information Technology, University of New South Wales, Canberra, Australia.
‡ Data61, Commonwealth Scientific and Industrial Research Organisation, Eveleigh, Australia.
mohammad.zarrabian@anu.edu.au, ni.ding@unimelb.edu.au, p.sadeghi@unsw.edu.au, thierry.rakotoarivelo@data61.csiro.au.

ABSTRACT

This paper is concerned with enhancing data utility in the privacy watchdog method for attaining information-theoretic privacy. For a specific privacy constraint, the watchdog method filters out the high-risk data symbols through applying a uniform data regulation scheme, e.g., merging *all* high-risk symbols together. While this method entirely trades the symbols resolution off for privacy, we show that the data utility can be greatly improved by partitioning the high-risk symbols set and individually privatizing each subset. We further propose an agglomerative merging algorithm that finds a suitable partition of high-risk symbols: it starts with a singleton high-risk symbol, which is iteratively fused with others until the resulting subsets are private. Numerical simulations demonstrate the efficacy of this algorithm in privately achieving higher utilities in the watchdog scheme.

***Index Terms*—** Information-theoretic privacy; Watchdog privacy mechanism; Privacy-utility trade-off.

1. INTRODUCTION

Industries and governments are increasingly sharing data to unlock economic and societal benefits through advances in data analytics and machine learning. However, such data also contains sensitive information about individuals or businesses, which makes the privacy regulators, users, and data providers concerned about the leakage of confidential information, either explicitly or implicitly. In signal processing and information theory, data privacy is underpinned in terms of a measure called *information lift* [1, 2].

To evaluate how much private data X is informative about the confidential data S, the lift measures the change in the posterior belief $p(s|x)$ from the prior belief $p(s)$ for each instance of s and x by

$$l(s,x) = \frac{p(s|x)}{p(s)}. \tag{1}$$

$\varepsilon = 1$	utility	privacy leakage
complete merging	0.5913	0.8037
two-subset merging	0.7335	0.8488

Table 1: An example of how subset merging can enhance utility in the watchdog mechanism.

It is clear that a small lift indicates limited private information gain by the adversary, and therefore, the more private X is. The lift is the elementary measure in almost all information leakage measures, e.g., mutual information [1, 3], Sibson mutual information [4–6], α-leakage [7] and local differential privacy [8, 9]: as proved in [2], if lift is bounded, all these leakage measures are also bounded.

The existing approach to attain lift-based privacy is the watchdog method proposed in [2]. For a specific privacy constraint, i.e., a threshold ϵ on the lift, the watchdog method filters out and privatizes the high-risk symbols of X. The authors in [2] adopted a uniform approach: merging all high-risk symbols together into a 'super' symbol, which is proved in [6, 10] to be the optimal privatization scheme in attaining data privacy. However, this uniform approach neglects an important issue in data privacy: to achieve the benefits of data sharing, the privatized data should provide a satisfactory level of usefulness in X.[1] Despite the relaxation attempts in [6, 10], the *complete merging* method minimizes the resolution of high-risk symbols and significantly deteriorates data utility, which is at odds with the purpose of data sharing.

In fact, even a small alteration can greatly enhance the data utility. In Table 1, we arbitrarily cut the high-risk symbol set (of size 7) into two subsets, each of which is then privatized individually. The utility (measured by mutual information) is increased significantly without sacrificing too much data privacy, which remains below the design constraint ϵ. This not only shows that the complete merging approach is an 'overkill' in terms of data utility, but also suggests a partitioned privatization approach.

This work is partly supported by the Data61 CRP: IT-PPUB and by the ARC Future Fellowship FT190100429.

[1] The data utility is usually quantified by average performance measures such as mutual information [3], f-divergence [11], and Hamming distortion [12]. We use mutual information in this paper.

In this paper, we propose the *subset merging* method for the watchdog mechanism to enhance the data utility. Finding the best partition of high-risk symbols set to achieve optimal utility is generally a combinatorial problem. Accordingly, we propose a heuristic greedy algorithm to find good subsets to merge, which guarantees data privacy. To do so, this greedy algorithm tries to search the finest partition of the original high-risk symbols set that ensures the resulting lift of the whole dataset does not exceed ϵ. It starts with the singleton partition of high-risk symbols (highest resolution) and iteratively merges symbols until lift values of the resulting subsets are all below ϵ. Numerical simulations show that our proposed algorithm enhances utility significantly and maintains the privacy leakage constraint.

2. SYSTEM MODEL

Denote random variables S and X the sensitive and public data, respectively. The joint distribution $p(s,x)$ describes the statistical correlation between S and X. To protect the privacy of S, we sanitize X to Y by the transition probability $p(y|x)$. Here, for each x, y, $p(y|x) = p(y|x,s), \forall s$ and therefore the Markov chain $S \to X \to Y$ is formed.

The watchdog method is based on the logarithm of the lift measure

$$i(s;x) = \log l(s;x).$$

For each $x \in \mathcal{X}$, denote the maximum symbol-wise information leakage by $\max_{s \in \mathcal{S}} |i(s,x)|$, where

$$\omega(x) = \max_{s \in \mathcal{S}} |i(s,x)|. \tag{2}$$

Applying an upper bound ϵ to $\omega(x)$ for all symbols $x \in \mathcal{X}$, the whole alphabet $\mathcal{X}$ is divided into two subsets: the low-risk subset is given by $\mathcal{X}_\varepsilon \triangleq \{x \in \mathcal{X} : \omega(x) \le \varepsilon\}$ that is safe to publish, and the high-risk symbol set

$$\mathcal{X}_\varepsilon^c = \mathcal{X} \setminus \mathcal{X}_\varepsilon \triangleq \{x \in \mathcal{X} : \omega(x) > \varepsilon\}$$

that requires some treatment before the data publishing.

The authors in [6, 10] adopt a uniform randomization scheme

$$p(y|x) = \begin{cases} 1_{\{x=y\}} & x, y \in \mathcal{X}_\varepsilon, \\ R(y) & x, y \in \mathcal{X}_\varepsilon^c, \\ 0 & \text{otherwise}, \end{cases} \tag{3}$$

where $R(y)$ is *complete merging* solution, e.g., where there is only one super symbol $y^* \in \mathcal{X}_\varepsilon^c$ such that $R(y^*) = 1$ for all $x \in \mathcal{X}_\varepsilon^c$ and $R(y) = 0$ otherwise.

After randomization, the log-lift is given by $i(s,y) = \log \frac{p(y|s)}{p(y)}$ where $p(y|s) = \sum_{x \in \mathcal{X}} p(y|x)p(x|s)$ due to the Markov property and $p(y) = \sum_{x \in \mathcal{X}} p(y|x)p(x)$. We can extend the notion of the log-lift, and $\omega(x)$ from a single $x \in \mathcal{X}$ to a subset $\mathcal{X}_Q \subseteq \mathcal{X}$ [10]:

$$i(s, \mathcal{X}_Q) = \log \frac{p(\mathcal{X}_Q|s)}{p(\mathcal{X}_Q)}, \quad \omega(\mathcal{X}_Q) = \max_{s \in \mathcal{S}} |i(s, \mathcal{X}_Q)|, \tag{4}$$

where $p(\mathcal{X}_Q|s) = \sum_{x \in \mathcal{X}_Q} p(x|s)$ and $p(\mathcal{X}_Q) = \sum_{x \in \mathcal{X}_Q} p(x)$.

Algorithm 1: Make a refinement of $\mathcal{X}_\varepsilon^c$

```
Input: $\mathcal{X}, \varepsilon, p(s,x)$
Output: $\mathcal{G}_{\mathcal{X}_\varepsilon^c} = \{\mathcal{X}_1, \mathcal{X}_2, \cdots \mathcal{X}_g\}$
Initialize: Obtain $\{\mathcal{X}_\varepsilon, \mathcal{X}_\varepsilon^c\}$, $\mathcal{X}_Q \leftarrow \mathcal{X}_\varepsilon^c$, and $g = 1$
while $|\mathcal{X}_Q| > 0$ do
    $\mathcal{X}_g = \arg\max_{x \in \mathcal{X}_Q} \omega(x)$, and $\mathcal{X}_Q \leftarrow \mathcal{X}_Q \setminus \mathcal{X}_g$;
    while $\omega(\mathcal{X}_g) > \varepsilon$ & $|\mathcal{X}_Q| > 0$ do
        $x^* = \arg\min_{x \in \mathcal{X}_Q} \omega(\mathcal{X}_g \cup \{x\})$
        $\mathcal{X}_g \leftarrow \mathcal{X}_g \cup \{x^*\}$ and $\mathcal{X}_Q \leftarrow \mathcal{X}_Q \setminus \{x^*\}$;
    end
    $\mathcal{G}_{\mathcal{X}_Q} = \{\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_g\}$, and $g \leftarrow g + 1$
end
while $\omega(\mathcal{X}_g) > \varepsilon$ & $|\mathcal{G}_{\mathcal{X}_Q}| > 1$ do
    $\mathcal{X}_k = \arg\min_{1 \le i < g} \omega(\mathcal{X}_g \cup \mathcal{X}_i)$;
    $\mathcal{X}_g \leftarrow \mathcal{X}_g \cup \mathcal{X}_k$; for $k + 1 \le i \le g$ update the
      index of $\mathcal{X}_i$'s to $\mathcal{X}_{i-1}$
    $\mathcal{G}_{\mathcal{X}_Q} = \{\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_g\}$
end
```

Applying $p(y|x)$, the value of $\max_{y \in \mathcal{Y}} \omega(y)$ as the upper bound on privacy leakage after randomization is given by [10]

$$\max_{y \in \mathcal{Y}} \omega(y) = \max\{\max_{y \in \mathcal{X}_\varepsilon} \omega(y), \omega(\mathcal{X}_\varepsilon^c)\}. \tag{5}$$

It is obvious that $\max_{y \in \mathcal{X}_\varepsilon} \omega(y) \le \varepsilon$, so it attains the privacy constraint. However, the value of $\omega(\mathcal{X}_\varepsilon^c)$ is variable and depends on the joint probability distribution $p(s,x)$.

Example 1 *Let* $\mathcal{X} = \{x_1, x_2, \cdots, x_5\}$, $\mathcal{S} = \{s_1, s_2, s_3\}$, *and* $\varepsilon = 0.8$. *We randomly generate a joint distribution* $p(s,x)$ *and the resulting maximum symbol-wise leakage and utility are* $\omega(x) = [1.3515, 1.6458, 0.9295, 0.8161, 0.2608]$ *and* $H(X) = 1.6034$*, respectively. For the given* ε*, the low-risk and high-risk subsets are given by* $\mathcal{X}_\varepsilon = \{x_5\}$ *and* $\mathcal{X}_\varepsilon^c = \{x_1, x_2, x_3, x_4\}$*, respectively. After randomization, assume high-risk symbols are mapped to* y^* *where* $y^* = y_1 = x_1$ *and* $y_2 = x_5$*, then the maximum symbol-wise leakage and utility are given by* $\omega(y) = [0.0627, 0.2608]$ *and* $I(X;Y) = 0.5269$*, respectively.*

In Example 1, the leakage in the high-risk subset after randomization is $\omega(\mathcal{X}_\varepsilon^c) = 0.0627$, which is an order of magnitude smaller than the original privacy constraint $\varepsilon = 0.8$, based on which $\mathcal{X}_\varepsilon^c$ was obtained. Although this small leakage guarantees a very high level of privacy, it damages utility drastically, the utility decreases from $H(X) = 1.6034$ to $I(X;Y) = 0.5269$. On the other hand, when a threshold ε is given as the privacy constraint, it is acceptable to just keep the privacy leakage less than ε, even if it becomes very close to

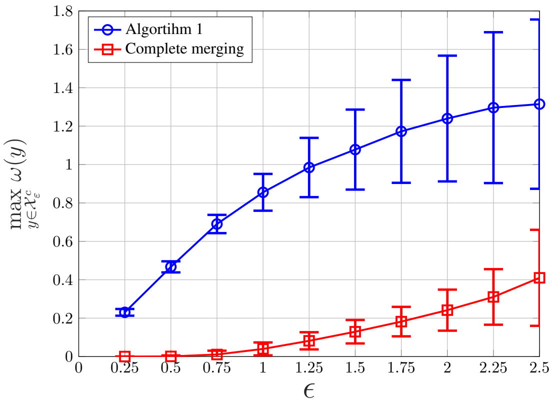

Fig. 1: Privacy leakage for different values of ε: The mean values of $\max_{y\in\mathcal{X}_\varepsilon^c} \omega(y)$ are shown with standard deviation. Algorithm 1 increases privacy leakage in $\mathcal{X}_\varepsilon^c$ in comparison with complete merging, however, in all cases it is still below the constraint ε.

this threshold. Thus, in the next section, we propose a subset merging method to enhance utility where the privacy leakage increases, but remains under ε to the extent possible.

3. ENHANCING UTILITY

In this section, we introduce an approach to enhance utility while maintaining a set privacy constraint. We measure utility by mutual information which for the bi-partition $\{\mathcal{X}_\varepsilon, \mathcal{X}_\varepsilon^c\}$ and complete merging randomization $\mathcal{X}_\varepsilon^c$ is given by [10]

$$I(X;Y) = H(X) + \sum_{x\in\mathcal{X}_\varepsilon^c} p(x) \log \frac{p(x)}{p\left(\mathcal{X}_\varepsilon^c\right)}. \tag{6}$$

Clearly, the utility depends on $p(x)$ for $x \in \mathcal{X}_\varepsilon^c$ and the overall $p(\mathcal{X}_\varepsilon^c)$. Our proposed approach to enhance data utility is through increasing data resolution. That is, we propose to randomize subsets of $\mathcal{X}_\varepsilon^c$ separately rather than the complete merging of the whole set $\mathcal{X}_\varepsilon^c$.

Let $[g] = \{1, 2, \cdots, g\}$ and consider a bi-partition $\{\mathcal{X}_\varepsilon, \mathcal{X}_\varepsilon^c\}$, a further partitioning of elements in $\mathcal{X}_\varepsilon^c$ denoted by $\mathcal{G}_{\mathcal{X}_\varepsilon^c} = \{\mathcal{X}_1, \cdots, \mathcal{X}_g\}$, and complete merging randomizations $R_i(y), i \in [g]$ where $\sum_{y\in\mathcal{X}_i} R_i(y) = 1$. In other words, we partition $\mathcal{X}_\varepsilon^c$ to subsets $\mathcal{X}_i$, so $\mathcal{X}_\varepsilon^c = \cup_{i=1}^g \mathcal{X}_i$ and each subset $\mathcal{X}_i$ is randomized by the corresponding randomization $R_i(y)$. Note that for $y \in \mathcal{X}_i$ we have $p(y) = \sum_{x\in\mathcal{X}_i} p(y|x)p(x) = p(\mathcal{X}_i)R_i(y)$. The resulting mutual information $I(X;Y)$ is

$$I(X;Y) = H(X) + \sum_{i=1}^{g} \sum_{x\in\mathcal{X}_i} p(x) \log \frac{p(x)}{p\left(\mathcal{X}_i\right)}. \tag{7}$$

Then the normalized mutual information-loss for partition $\mathcal{G}_{\mathcal{X}_\varepsilon^c}$ on $\mathcal{X}_\varepsilon^c$ is defined as

$$\text{NMIL}(\mathcal{G}_{\mathcal{X}_\varepsilon^c}) = \frac{H(X) - I(X;Y)}{H(X)}. \tag{8}$$

Since $p(\mathcal{X}_i) \leq p(\mathcal{X}_\varepsilon^c)$ for $i \in [g]$ the data resolution increases for each $x \in \mathcal{X}_i$ and this can results in a larger mutual information and hence a lower utility loss. The following definition and derivations formalize this observation.

Definition 1 *Assume two partitions $\mathcal{G}_{\mathcal{X}_\varepsilon^c} = \{\mathcal{X}_1, \cdots, \mathcal{X}_g\}$ and $\mathcal{G}'_{\mathcal{X}_\varepsilon^c} = \{\mathcal{X}'_1, \cdots, \mathcal{X}'_{g'}\}$. We say $\mathcal{G}'_{\mathcal{X}_\varepsilon^c}$ is a refinement of $\mathcal{G}_{\mathcal{X}_\varepsilon^c}$ and $\mathcal{G}_{\mathcal{X}_\varepsilon^c}$ is an aggregation of $\mathcal{G}'_{\mathcal{X}_\varepsilon^c}$ [13], if for every $i \in [g]$, $\mathcal{X}_i = \cup_{j\in J_i} \mathcal{X}'_j$ where $J_i \subseteq [g']$, and we have $p(\mathcal{X}_i) = \sum_{j\in J_i} p(\mathcal{X}'_j)$.*

If $\mathcal{G}'_{\mathcal{X}_\varepsilon^c}$ is a refinement of $\mathcal{G}_{\mathcal{X}_\varepsilon^c}$ and $\mathcal{G}_{\mathcal{X}_\varepsilon^c}$ is an aggregation of $\mathcal{G}'_{\mathcal{X}_\varepsilon^c}$ then $\text{NMIL}(\mathcal{G}'_{\mathcal{X}_\varepsilon^c}) \leq \text{NMIL}(\mathcal{G}_{\mathcal{X}_\varepsilon^c})$. This is because

$$H(X) \times \text{NMIL}(\mathcal{G}_{\mathcal{X}_\varepsilon^c}) = \sum_{i=1}^{g} \sum_{x\in\mathcal{X}_i} p(x) \log \frac{p(\mathcal{X}_i)}{p(x)} \tag{9}$$

$$\geq \sum_{i=1}^{g'} \sum_{x\in\mathcal{X}_j} p(x) \log \frac{p(\mathcal{X}_j)}{p(x)} = H(X) \times \text{NMIL}(\mathcal{G}'{}_{\mathcal{X}_\varepsilon^c}).$$

Generally, finding an optimal partition $\mathcal{G}_{\mathcal{X}_\varepsilon^c}$ that maximizes utility while maintaining the privacy constraint is combinatorial since it depends on the high-risk symbol probabilities $p(x)$, $x \in \mathcal{X}_\varepsilon^c$ and the joint probability distribution $p(s, x)$. However, we can use ε as a stop criteria to make a heuristic agglomerative algorithm for obtaining a refinement of $\mathcal{X}_\varepsilon^c$ to enhance utility as much as possible.

3.1. A greedy algorithm to refine the high-risk subset $\mathcal{X}_\varepsilon^c$

Knowing that the aggregation of the partition of $\mathcal{X}_\varepsilon^c$ reduces the symbol resolution and data utility, we propose a heuristic greedy algorithm that determines the most refined partition of $\mathcal{X}_\varepsilon^c$ that satisfies the data privacy constraint specified by ϵ. This is a bottom-up algorithm, which bootstraps from the most refined (singleton-element) partition of $\mathcal{X}_\varepsilon^c$. This starting point provides the highest resolution/utility but results in a lowest data privacy level. We let the subsets in the singleton partition merge with each other to reduce the log-lift measure $\omega(\mathcal{X}_i), \mathcal{X}_i \in \mathcal{G}_{\mathcal{X}_\varepsilon^c}$ until the log-lift of all subsets is reduced below ϵ. To achieve a low complexity, but effective procedure that results in a finer partition of $\mathcal{X}_\varepsilon^c$, we implement the subset merging in order. A good candidate to begin with is the most challenging symbol with the highest log-lift leakage.

The pseudo-code of our method is shown in Algorithm 1. To find each subset $\mathcal{X}_i \subseteq \mathcal{X}_\varepsilon^c$, we start with $\arg\max_{x\in\mathcal{X}_\varepsilon^c} \omega(x)$ as the symbol with the highest risk in $\mathcal{X}_\varepsilon^c$ (line 4) and then to make the leakage of the subset $\omega(\mathcal{X}_i)$ less than ε we merge another symbol to it that minimizes $\omega(\mathcal{X}_i)$ in each iteration (lines 5-8). Each symbol that is added to $\mathcal{X}_i$ is removed from $\mathcal{X}_\varepsilon^c$ (line 7). When a subset $\mathcal{X}_i$ is made, we add it to the partition set $\mathcal{G}_{\mathcal{X}_\varepsilon^c}$ and repeat the same process for the remaining $x \in \mathcal{X}_\varepsilon^c$. After making all subsets, there is

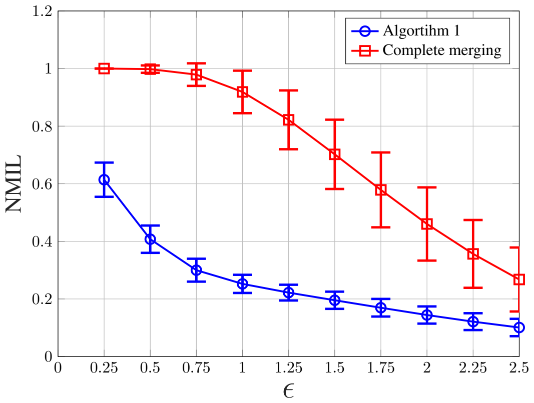

Fig. 2: Normalized Utility loss (NMIL) for different values of ε: The mean values of NMIL are shown with standard deviation. Algorithm 1 reduces NMIL in comparison with complete merging.

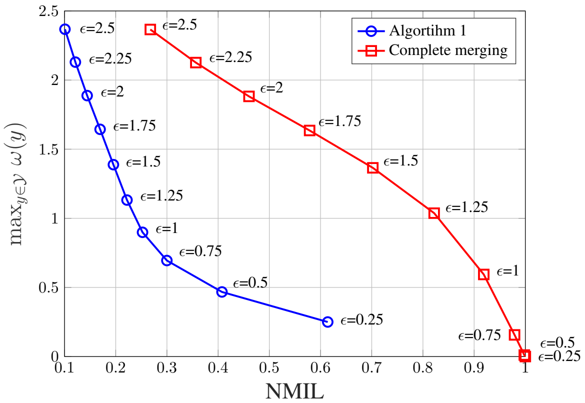

Fig. 3: Privacy Utility trade-off: For each value of ε, algorithm 1 reduces NMIL substantially while maintain privacy leakage below ε.

a possibility that for the last subset $\mathcal{X}_g$ the leakage is greater than ε. That is, $\omega(\mathcal{X}_g) > \varepsilon$. If this happens, while $\omega(\mathcal{X}_g) > \varepsilon$ we make an agglomerate $\mathcal{X}_g$ by merging a subset to it that minimizes $\omega(\mathcal{X}_g)$ (lines 11-15). A complete merging is a special output of our algorithm if no better finer partition can be found that maintains privacy.

It can be verified that the complexity of Algorithm 1 is $O(|\mathcal{X}_\varepsilon^c|^2)$. The reason is that the size of $\mathcal{X}_\varepsilon^c$ decreases after making each subset.

4. EXPERIMENTS

To make a comparison between Algorithm 1 and complete merging, we randomly generated 1000 joint distributions $p(s,x)$ where $|\mathcal{X}| = 20$ and $|\mathcal{S}| = 13$. For each distribution, after randomization we obtained maximum privacy leakage of high-risk symbols $\max_{y\in\mathcal{X}_\varepsilon^c} \omega(y)$, maximum overall privacy leakage $\max_{y\in\mathcal{Y}} \omega(y)$, and the utility loss NMIL under Algorithm 1 and complete merging for different values of $\varepsilon \in \{0.25, 0.5, 0.75, \cdots, 2.25, 2.5\}$. Then for each ε, we derived the mean value and standard deviation of $\max_{y\in\mathcal{X}_\varepsilon^c} \omega(y)$, $\max_y \omega(y)$, and NMIL across these 1000 distributions.

In Fig. 1 the mean value of $\max_{y\in\mathcal{X}_\varepsilon^c} \omega(y)$ is depicted for each ε, as well as its standard deviation (shown as tolerance bars). As expected, complete merging makes a strong guarantee on privacy leakage and keeps both mean and standard deviation much less than ε in all cases. In contrast, algorithm 1 increases the privacy leakage and lets it be closer to ε compared to complete merging, but crucially it still keeps the mean value and the corresponding deviation less than ε in all cases. As the value of ε increases, the standard deviation is also increased. This is because when ε increases, the privacy constraint is less strict and consequently, the size of $\mathcal{X}_\varepsilon^c$ decreases. As a result, the sample size to calculate the mean value of privacy leakage reduces, which causes a larger deviation.

Fig. 2 shows the normalized mutual information loss under Algorithm 1 and complete merging. It demonstrates that Algorithm 1 enhances utility substantially for each value of ε.

Finally, to have a clear privacy-utility trade-off (PUT) comparison, we present PUT curves for both Algorithm 1 and complete merging in Fig. 3. For each ε, the mean value of overall privacy leakage $\max_{y\in\mathcal{Y}} \omega(y)$ is shown versus the corresponding mean value of NMIL. Clearly, Algorithm 1 enhances utility significantly while satisfying the privacy leakage constraint. For a very strict constraint on privacy ($\varepsilon \leq 0.5$), complete merging results in perfect privacy with total utility loss where NMIL $= 1$. However, Algorithm 1 keeps the average utility loss less than 1.

5. CONCLUSION

In this paper, we introduced a method to enhance utility in the watchdog privacy mechanism. We showed that it is possible to maintain the privacy constraint and improve utility. In our approach, instead of randomizing the whole high-risk partition, we randomized subsets of high-risk symbols separately. Then we proposed a heuristic greedy algorithm to find subsets of high-risk elements and at the same time keep the leakage of each subset less than the privacy constraint, ε. The simulation results showed substantial utility enhancement and preservation of the privacy constraint.

In future research, it would be interesting to consider how to develop more efficient subset merging algorithms. Investigation of effective parameters like subset size and different privacy and utility measures could be helpful. It would be also beneficial to apply the proposed algorithm to real-world data sets and measure actual practical utility.

6. REFERENCES

[1] F. d. P. Calmon and N. Fawaz, "Privacy against statistical inference," in *Proc. Annu. Allerton Conf. Commun., Control, and Comput.*, Monticello, IL, 2012, pp. 1401–1408.

[2] H. Hsu, S. Asoodeh, and F. d. P. Calmon, "Information-theoretic privacy watchdogs," in *Proc. IEEE Int. Symp. Inf. Theory*, Paris, France, 2019, pp. 552–556.

[3] A. Makhdoumi, S. Salamatian, N. Fawaz, and M. Médard, "From the information bottleneck to the privacy funnel," in *Proc. IEEE Inf. Theory Workshop*, Hobart, TAS, 2014, pp. 501–505.

[4] R. Sibson, "Information radius," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, vol. 14, no. 2, pp. 149–160, 1969.

[5] S. Verdú, "α-mutual information," in *Proc. Information Theory and Applications Workshop (ITA)*, San Diego, CA, 2015, pp. 1–6.

[6] N. Ding, M. A. Zarrabian, and P. Sadeghi, "α-information-theoretic privacy watchdog and optimal privatization scheme," in *Proc. IEEE Int. Symp. Inf. Theory*, 2021, pp. 2584–2589.

[7] J. Liao, O. Kosut, L. Sankar, and F. d. P. Calmon, "Tunable measures for information leakage and applications to privacy-utility tradeoffs," *IEEE Trans. Inf. Theory*, vol. 65, no. 12, pp. 8043–8066, 2019.

[8] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proc. IEEE 54th Annu. Symp. Found. Comput. Sci.*, 2013, pp. 429–438.

[9] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?," *SIAM Journal on Computing*, vol. 40, no. 3, pp. 793–826, 2011.

[10] P. Sadeghi, N. Ding, and T. Rakotoarivelo, "On properties and optimization of information-theoretic privacy watchdog," in *Proc. IEEE Inf. Theory Workshop*, 2020.

[11] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," in *Adv. Neural Inf. Process. Syst*, jan 2014, vol. 4, pp. 2879–2887.

[12] A. D. Sarwate and L. Sankar, "A rate-disortion perspective on local differential privacy," in *Proc. Annu. Allerton Conf. Commun., Control, and Comput.*, 2014, pp. 903–908.

[13] Y. Liu, P. Sadeghi, F. Arbabjolfaei, and Y. H. Kim, "Capacity theorems for distributed index coding," *IEEE Trans. Inf. Theory*, vol. 66, no. 8, pp. 4653–4680, 2020.