

OptimShare: A Unified Framework for Privacy Preserving Data Sharing – Towards the Practical Utility of Data with Privacy

M.A.P. Chamikara^{1,2}[0000-0002-4286-3774], Seung Ick Jang^{1,2}, Ian Oppermann³, Dongxi Liu^{1,2}, Musotto Roberto², Sushmita Ruj⁴, Arindam Pal⁴, Meisam Mohammady⁵, Seyit Camtepe^{1,2}, Sylvia Young⁶, Chris Dorrian⁶, and Nasir David⁶

¹ CSIRO's Data61, Australia

² Cyber Security Cooperative Research Centre (CSCRC), Australia

³ Customer, Delivery and Transformation, Department of Customer Service, New South Wales, Australia

⁴ University of New South Wales, Sydney, Australia

⁵ Iowa State University of Science and Technology, Iowa, USA

⁶ Department of Health, Western Australia, Australia

Abstract. Tabular data sharing serves as a common method for data exchange. However, sharing sensitive information without adequate privacy protection can compromise individual privacy. Thus, ensuring privacy-preserving data sharing is crucial. Differential privacy (DP) is regarded as the gold standard in data privacy. Despite this, current DP methods tend to generate privacy-preserving tabular datasets that often suffer from limited practical utility due to heavy perturbation and disregard for the tables' utility dynamics. Besides, there has not been much research on selective attribute release, particularly in the context of controlled partially perturbed data sharing. This has significant implications for scenarios such as cross-agency data sharing in real-world situations. We introduce OptimShare: a utility-focused, multi-criteria solution designed to perturb input datasets selectively optimized for specific real-world applications. OptimShare combines the principles of differential privacy, fuzzy logic, and probability theory to establish an integrated tool for privacy-preserving data sharing. Empirical assessments confirm that OptimShare successfully strikes a balance between better data utility and robust privacy, effectively serving various real-world problem scenarios.

Keywords: data sharing · data privacy · tabular data sharing · privacy preserving data sharing

1 Introduction

Sharing data containing personally identifiable information (PII) may result in the exposure of sensitive personal information, thereby posing potential risks to user privacy. Data privacy, while possessing various definitions, can be characterized as “Controlled Information Release” in the context of data sharing and analysis [5]. The literature reveals several methods to ensure privacy in data sharing and analytics via “Controlled Information Release”. Among these, disclosure control has gained prominence due to its practicality [10,24]. This process entails applying various privacy preservation techniques to data prior to its release for analysis. Differential privacy (DP) is the gold standard for disclosure control mechanisms, attributed to its stringent privacy guarantees. An algorithm M adheres to differential privacy if, for every pair of neighboring datasets x and y , and all potential outputs S , the inequality $Pr[M(x) \in S] \leq \exp(\epsilon)Pr[M(y) \in S] + \delta$ holds. In this context, ϵ represents the privacy budget, indicating the privacy leak, while δ signifies the probability of model failure.

In the realm of data sharing, tabular data sharing (non-interactive data sharing) is particularly significant, as tabular data are often exchanged among agencies or released publicly in tabular format. Non-interactive data sharing poses a significant challenge due to the high degree of randomization required to maintain privacy (acceptable ϵ values), which can result in reduced utility in the shared data [7]. Despite its complexity, non-interactive data sharing is crucial for enabling various opportunities, as it allows analysts to access the entire dataset for analysis without being limited to a single query output (e.g., mean). Several differentially private (DP) approaches for non-interactive data sharing, have been proposed [8,12,22,23]. However, selecting the optimal DP approach for differentially private non-interactive data sharing is challenging due to factors such as the diversity of input datasets (e.g., statistical properties, dimensions) and the variety of applications (e.g., data clustering, deep learning) [29]. Furthermore, unanticipated data leaks may occur when privacy constraints (ϵ and δ) are relaxed to achieve higher utility [21].

Prior solutions primarily emphasize one-to-one mapping between input dataset properties (e.g., table size) and output datasets, assuming fully perturbed data can deliver sufficient utility, often diverging from real-world needs [29]. However, factors such as trustworthiness levels of third parties (e.g., fully-trusted \rightarrow fully-untrusted) and unique utility dynamics for diverse applications must be considered. Thus, investigating a partial data perturbation approach, where specific columns remain non-perturbed, is crucial. Differential privacy (DP) in non-interactive data sharing with a subset of the dataset (strategically chosen attributes) being released for mandated purposes has not been thoroughly explored. This is paramount in real-world contexts, such as cross-agency data sharing settings. Incorporating a non-perturbed vertical partition in the final dataset would enhance utility for custom query-based applications, but necessitates in-depth analysis concerning linkability and attack resilience, a problem we refer to as controlled partially perturbed non-interactive data sharing (CPNDS). A framework enabling CPNDS in an application-specific utility and privacy-preserving manner is indispensable. CPNDS challenges involve (1) the presence of various complex input data dynamics (e.g., categorical / non-categorical), (2) utility maintenance for diverse application demands, and (3) striking an appropriate privacy-utility balance. A unified framework-based solution addressing these concerns is required for CPNDS, but currently, no such comprehensive solutions exist.

In addressing this issue, we present a unified multi-criterion framework-based solution, called OptimShare, to generate a practical privacy-preserving instance of an input dataset under CPNDS. We presume OptimShare operates under a central authority (a data custodian such as a government agency, hospital, or bank) with full ownership and control over the datasets before releasing a privacy-preserving version, which is a primary requirement for CPNDS. OptimShare employs an iterative method to identify the optimal perturbed instance for release in data analytics. The empirical results demonstrate that OptimShare effectively balances utility and privacy for the selected dataset intended for release. Additionally, a comprehensive tool, available in both web-based and stand-alone versions, was developed to automate the entire CPNDS process.

2 Background

This section briefly discusses the background of methods utilized in OptimShare. These approaches include differential privacy and fuzzy logic.

2.1 Data Perturbation and Differential Privacy

OptimShare enforces data privacy through perturbation techniques, which can be classified into interactive and non-interactive approaches. Interactive approaches involve aggregated data release [14], while non-interactive methods enable the release of a perturbed, privacy-preserving version of an input dataset, such as additive perturbation [16,26], data swapping [20], Privsyn [34], PrivatePGM [25], and DP-WGAN [32]. OptimShare focuses on privacy-preserving tabular data release and employs non-interactive perturbation techniques.

Differential privacy OptimShare’s objective is to enforce differential privacy (DP) on output data. DP is the most widely accepted privacy model [15]. DP mechanisms such as Privsyn [34], PrivatePGM [25], and DP-WGAN [32] have gained interest, with this paper focusing on PrivatePGM and DP-WGAN for tabular data generation in OptimShare.

DP-WGAN, a DP data generation technique, uses the Generative Adversarial Network (GAN) framework and the DP-SGD algorithm [1] to sanitize discriminator gradients during training [18,32]. PrivatePGM [25] is a solution for privacy-preserving probabilistic graphical models (PGMs). PrivatePGM leverages differentially private algorithms to enable the analysis of sensitive data without sacrificing privacy.

Conventionally, DP uses two parameters, ϵ (the privacy budget) and δ (the model failure probability), to constraint privacy leakage [3]. A randomization algorithm (DP mechanism - M) applied to a dataset (D) is guided by these parameters [3]. For a mechanism to satisfy (ϵ, δ) -differential privacy, it must satisfy Equation (1) [3], where d , and d' are datasets differing by one record.

Definition 1 A randomized algorithm M with domain $\mathbb{N}^{|\mathcal{X}|}$ and range R : is (ϵ, δ) -differentially private for $\delta \geq 0$ if for every adjacent datasets $d, d' \in \mathbb{N}^{|\mathcal{X}|}$ and for any subset $S \subseteq R$,

$$P[M(d) \in S] \leq e^\epsilon P[M(d') \in S] + \delta \quad (1)$$

Postprocessing invariance property of DP Postprocessing invariance refers to the ability of a differential privacy (DP) algorithm to preserve its privacy guarantee despite additional computations on its outputs. Consequently, the result of any postprocessing on an $\epsilon - DP$ output remains $\epsilon - DP$ [9].

2.2 Fuzzy Inference Systems

OptimShare employs Fuzzy Logic (FL) [19,31] to generate potential pairs of (ϵ, δ) values, conforming to the pre-established privacy requirements of a dataset. FL models imprecise definitions computationally via a fuzzy inference system (FIS) with three steps: fuzzification, rule evaluation, and defuzzification. **Fuzzification** maps crisp inputs to fuzzy values; **rule evaluation** links fuzzy input memberships to an output domain using a rule base; and **defuzzification** converts aggregated output memberships to a crisp value using methods such as the center of gravity (Equation 2) [19,31].

$$COG = \frac{\int_{min}^{max} \mu_x x dx}{\int_{min}^{max} \mu_x dx} \quad (2)$$

3 The proposed work: OptimShare

OptimShare is controlled by a central authority (a data custodian), such as a government agency, to handle Controlled Partially Perturbed Non-Interactive Data Sharing (CPNDS) with differential privacy, as illustrated in Figure 1. The objective is to create a privacy-preserving version of the existing dataset for third-party analytics utilization. For enhanced dataset security, user role management is integrated to regulate access levels. The focus of this paper is on the OptimShare central algorithm, presuming that the data curator has unrestricted access to the dataset and OptimShare for producing a privacy-preserving dataset.

3.1 Problem Definition

Given a dataset D with n tuples, m attributes, and $r (< m)$ sensitive attributes forming $S - dataset$, D_r , the remaining $(m - r)$ attributes form $D_{(m-r)}$. Applying differentially private algorithm M to D_r generates perturbed dataset D_r^p with n tuples and r attributes, privacy constrained by the privacy parameters of M . The composition of D_r^p and $D_{(m-r)}$ is released as D^p .

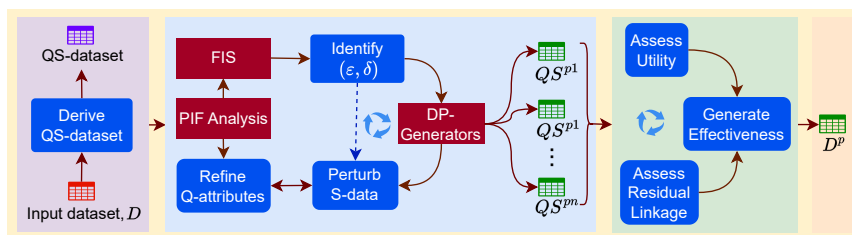


Fig. 1: The modular arrangement of the OptimShare framework. D^p represents the perturbed output dataset of the input dataset, D . FIS represents the Fuzzy Inference System. PIF represents the Personal Information Factor. QS^{pi} represents the intermediate perturbed instances of the QS dataset.

3.2 OptimShare Algorithm

Applying perturbation directly to an input dataset D using a mechanism M to create a privacy-preserving dataset, D^p , leaves certain questions unresolved. Algorithm 1 demonstrates the approach employed by OptimShare for generating privacy-preserving (perturbed) datasets, effectively addressing these concerns.

Algorithm 1: OptimShare algorithm for generating a privacy-preserving dataset

D \leftarrow input dataset
 T_ε \leftarrow threshold ε
 $TN_{\varepsilon,\delta}$ \leftarrow total (ε, δ)
 combinations
 TS \leftarrow total number of
 searches
Input: t \leftarrow perturbed instances
 per combination
 A \leftarrow application
 C \leftarrow effectiveness
 coefficient
 E^T \leftarrow effectiveness
 threshold
Output: D^p \leftarrow perturbed dataset
 of D

- 1 Identify identifiers (ID) and quasi-identifiers (Q);
- 2 Remove ID from the dataset to produce QS -dataset;
- 3 Identify tuple distribution of the QS -dataset;
- 4 Determine PIF of Q -dataset;
- 5 Determine PIF of Q -dataset conditioned to S ;
- 6 Refine the Q attributes and S attributes;
- 7 Generate $TN_{\varepsilon,\delta}$ combinations as $\{(\varepsilon_1, \delta_1), \dots, (\varepsilon_{TN}, \delta_{TN})\}$ (where $\varepsilon_i < \varepsilon_{i+1}$ and $\delta_i < \delta_{i+1}$);
- 8 **for** each $(\varepsilon_i, \delta_i)$ at $TN_{\varepsilon,\delta}$ intervals in TS **do**
- 9 Apply DP generators (DPA_1, \dots, DPA_n) to generate t perturbed instances;
- 10 Merge Q to all t perturbed instances (DP_1, \dots, DP_t);
- 11 Generate normalized utility values of all DP_i ;
- 12 Generate t residual leak normalized values;
- 13 Find effectiveness loss value el_i of each t perturbed instances for A using C and T_ε ;
- 14 Choose all DP_i that satisfy $e_i \geq E^T$ (where $e_i = 1 - el_i$);
- 15 **Return** D^p with the highest e_i ;

First, OptimShare identifies the three primary types of attributes in the input dataset, namely identifiers, quasi-identifiers, and sensitive attributes, subsequently eliminating the identifiers. Next, the distribution of tuples in the remaining dataset, referred to as the QS -dataset, is determined. The QS attributes are then further refined. The algorithm generates combinations of privacy parameters, ε and δ specific to the input dataset. Next, OptimShare employs differentially private algorithms on the sensitive portion of the dataset ($S - dataset$), leveraging each (ε, δ) combination to generate perturbed instances. The effectiveness values for the perturbed QS -datasets are then calculated. The algorithm finally returns the perturbed dataset with the highest effectiveness value.

3.3 The Main Steps of OptimShare Algorithm

Given an input dataset D with m attributes and n tuples, OptimShare identifies identifier attributes (ID) and quasi-attributes (Q) within D . To protect against direct identification, ID attributes are excluded from D based on their uniqueness. The dataset intended for publication after perturbation is formed by combining Q and the remaining vertical partition S , referred to as the QS -dataset.

Identifying initial tuple distribution of the dataset to allow M to maintain the data distribution in QS The optimal clustering dynamics are found using the k -means algorithm and Silhouette analysis [13], unless the input dataset is a classification dataset with existing class labels representing tuple distribution (refer to Algorithm 2).

Algorithm 2: Identifying original tuple distribution of the input dataset

Input: QS \leftarrow QS dataset
 cn_range \leftarrow list of cluster numbers to be searched
Output: T_s \leftarrow tuple status

- 1 **for** each $cn \in \{cn_range\}$ **do**
- 2 run k -means clustering on QS , where $k = cn$;
- 3 s_{cn} = Silhouette Coefficient of cn ;
- 4 select the cn of $maximum(s_{cn})$;
- 5 **return** T_s , which is the k -means cluster label of each tuple under $maximum(s_{cn})$;

Identification Q attributes A quasi-identifier (Q), a unique attribute set capable of distinguishing a record, could potentially facilitate linkability via auxiliary data, posing a risk to privacy leakage.

Declaring Q attributes Selecting data-specific Q attributes is challenging due to variable definitions of sensitive attributes (i.e., domain specific). OptimShare addresses this by using a global set of common Q attributes (GQ). These attributes are then refined based on their distinguishability using the personal information factor (PIF), a measure that gauges record indistinguishability.

Cell surprise factor (CSF) and personal information factor (PIF) Ian et al. defined PIF using entropy-based KL-divergence [27,28]. We extend the idea and propose CSF as a bounded measure for assessing attribute impact on record indistinguishability. CSF is computed using Equations 3, 4, and 5. The CSF provides a unique method to assess how the indistinguishability of records is affected by the introduction of a specific attribute or set of attributes. PIF, bounded by $[0,1]$, encapsulates the attribute's CSF distribution (Definition 1).

$$\text{Prior}(X) : \text{Prior}(X) = P(X = x) = \frac{|x|}{|X|} \quad (3)$$

$$\text{Posterior}(X) : \text{Posterior}(X) = P(X = x|Y = y) = \frac{|x, y|}{|y|} \quad (4)$$

$$\text{CSF Definition} : \text{CSF} = |\text{Prior}(X) - \text{Posterior}(X)| \quad (5)$$

Note that CSF is upper bounded by $Posterior(X)$ as OptimShare only looks at the increase in indistinguishability. Hence, in all cases interested, $Prior(X) \leq Posterior(X)$.

Definition 1 (PIF). Let x_i be the CSF value bins (bounded by $[0,1]$) of an attribute, where h_i is the number of occurrences of each x_i .

Then,

$$PIF = \frac{\sum_{i=1}^n x_i h_i}{\sum_{i=1}^n h_i} \quad (6)$$

Application of perturbation on the QS -dataset The perturbation of the QS -dataset is a four-step process: (1) Conduct the PIF analysis on the Q attributes, (2) Refine the Q and S attributes per PIF outcomes, (3) Define the privacy parameters (ϵ and δ) for the S -dataset through the PIF analysis, and (4) Implement perturbation on S data and determine the optimal perturbed instance for sharing.

Assessing the Q attributes using PIF Calculate $QPIF_i$ (i.e., PIF) for all Q attributes in the Q -dataset. Determine $QSPIF_i$ (i.e., PIF) for all Q attributes in the QS -dataset to evaluate the influence of S attributes on each Q attribute. The difference between $QPIF_i$ and $QSPIF_i$ indicates the independence of a specific Q attribute's data distribution from the S attributes. The inequality $\Delta PIF_i \geq \alpha QPIF_i$ determines the extent of PIF change, where $\Delta PIF_i = QSPIF_i - QPIF_i$ and α is the sensitivity coefficient. If $\alpha = 1$, it means the PIF leak from Q_i in the QS dataset is exactly $QPIF_i$, implying that $QPIF_i < 0.5$. Attributes that satisfy $\Delta PIF_i \geq QPIF_i$ are moved to the S -dataset for perturbation, as their distribution is significantly altered by S attributes, which could otherwise risk personal information leakage.

Next, as the initial step to determine the privacy requirements of the S -dataset, we calculate the PIF (PIF_{Thresh}) of the QS dataset using Equation 7. In the equation, $QSMaXPIF$ is the maximum PIF value returned by the QS dataset.

$$PIF_{Thresh} = \begin{cases} QSMaXPIF & \text{if } QSMaXPIF < 1 \\ 1 & \text{otherwise} \end{cases} \quad (7)$$

Developing a link between PIF and (ϵ, δ) A link between PIF and (ϵ, δ) in terms of enforcing differential privacy can be modeled as follows:

The definition of (ϵ, δ) -differential privacy characterizes the probabilistic bounds for a randomized algorithm or statistical mechanism M . For every pair of neighboring datasets d and d' (that differ by a single individual's data) and for every possible subset of the output space $S \subseteq Range(M)$, this model ensures that:

$$P[M(d) \in S] \leq e^\epsilon P[M(d') \in S] + \delta \quad (8)$$

where $P[M(d) \in S]$ denotes the probability that the mechanism M produces an output in set S with input dataset d .

Here, ε signifies the privacy parameter (the privacy budget), and δ is a negligible quantity representing the probability of the privacy mechanism potentially violating the ε -privacy condition. As ε approaches zero and δ is sufficiently small, a higher degree of privacy protection is conferred. Hence, we can define a privacy metric $f(\varepsilon, \delta) = (1 - \exp(-\varepsilon)) + \delta$, which serves as a suitable gauge for quantifying privacy levels. Consequently, a decrease in the value of $f(\varepsilon, \delta)$ indicates an enhanced privacy protection.

One essential property of differential privacy is its postprocessing invariance, implying that if a random mechanism M guarantees (ε, δ) -differential privacy, then any post-processing function g applied to the output of M also maintains the (ε, δ) -differential privacy. Formally, if M ensures (ε, δ) -differential privacy, then the composed mechanism $g \circ M$ is also (ε, δ) -differentially private for all functions g .

In the non-interactive privacy-preserving data publishing paradigm, a data curator generates a differentially private version of a dataset D using a differentially private mechanism M . In this setting, $f(\varepsilon, \delta)$ acts as an upper bound for privacy loss, ensuring that privacy loss does not exceed $(1 - \exp(-\varepsilon)) + \delta$.

Examining a particular attribute $A \in D$, we define the ‘‘Personal Information Factor’’ (PIF_A) that quantifies the attribute-specific distinguishability level. For each attribute A , we define Δ_A as the increase in indistinguishability, which can be represented as:

$$\Delta_A = Posterior(A) - Prior(A) \quad (9)$$

The relationship between PIF_A and Δ_A is given by:

$$PIF_A = \frac{\sum_{i=1}^n \Delta_{A_i} h_i}{\sum_{i=1}^n h_i} \quad (10)$$

where Δ_{A_i} represents the increase in indistinguishability for the attribute A in the i -th bin with h_i occurrences.

Utilizing PIF_A for each attribute, we can introduce the privacy measure f_A as follows:

$$f_A(PIF_A, \delta) = PIF_A + \delta. \quad (11)$$

Consequently, we can derive a privacy measure for the entire dataset D using the maximum Personal Information Factor (PIF_{Thresh}) over all attributes in D . Hence, the privacy measure for the dataset can be defined as:

$$f_D(\varepsilon, \delta) = PIF_{Thresh} + \delta. \quad (12)$$

$f_D(\varepsilon, \delta)$ signifies an upper bound to privacy loss upon the release of the dataset and provides a quantitative control mechanism balancing data utility and privacy protection. $PIF_{Thresh} = \max(PIF_{A_i})$ signifies the maximum PIF across all attributes, indicating the dataset’s potential to satisfy privacy parameters without any attribute surpassing this threshold. A fuzzy model can now be utilized to represent this relationship between PIF_{Thresh} and (ε, δ) .

Determination of the privacy parameters (ε and δ) for S -dataset perturbation Optimshare employs a fuzzy inference system (FIS) for determining suitable ε and δ inputs for the S -dataset from PIF_{Thresh} . Higher PIF (PIF_{Thresh}) suggests enhanced distinguishability of the QS dataset and, consequently, greater privacy need for the S data via increased perturbation. We model an FIS to encapsulate the relationship between PIF , ε , and δ . Each of the three fuzzy variables have three Gaussian-shaped membership functions (LOW , $MEDIUM$, $HIGH$) signifying different input value ranges and facilitating a gradual shift between functions for a broader value spectrum (see Figure 2a). The mean (μ) and standard deviation (σ) for LOW , $MEDIUM$, and $HIGH$ are respectively set as $(\mu = 0, \sigma = 1)$, $(\mu = 0.5, \sigma = 1)$, and $(\mu = 1, \sigma = 1)$.

Rule 1: *IF*($\varepsilon = LOW$) *THEN* ($PIF = HIGH$)

Rule 2: *IF*($\delta = LOW$) *THEN* ($PIF = HIGH$)

Rule 3: *IF*($\varepsilon = MEDIUM$ AND $\delta = MEDIUM$) *THEN* ($PIF = MEDIUM$) (13)

Rule 4: *IF*($\varepsilon = HIGH$) *THEN* ($PIF = LOW$)

Rule 5: *IF*($\delta = HIGH$) *THEN* ($PIF = LOW$)

Figure 2a depicts the fuzzification of variables ε , δ , and PIF , with the y-axis quantifying their degree of membership. A fuzzy rule base, providing the foundation for fuzzy inference, is established

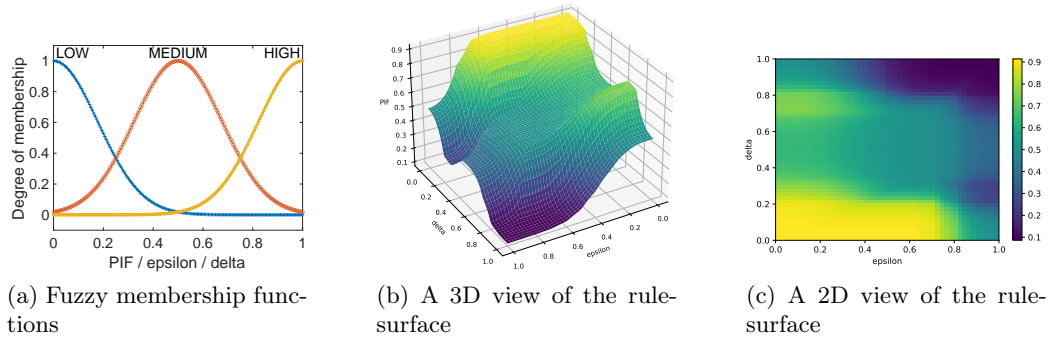


Fig. 2: The mapping between the three fuzzy variables and the change of PIF against the changes of δ and ϵ .

next. Equation 13 represents the proposed FIS rules, defined by the IF-THEN convention (e.g., $IF(\epsilon = MEDIUM \text{ AND } \delta = HIGH) \text{ THEN } (PIF = MEDIUM)$). The FIS rule evaluation step fuses fuzzy conclusions into one via the fuzzy rule base, applying $MAX - MIN$ (OR for MAX and AND for MIN) operation. The minimum among membership levels is considered for each rule, while the maximum fuzzy value from all rule outputs determines the value conclusion.

Figure 2 depicts the rule surface between the three fuzzy variables. As shown in the rule surface, higher values of PIF correspond to lower values for ϵ and δ . The final step of the FIS is the defuzzification based on the rule aggregated shape of the output function. We use the centroid-based technique to obtain the final defuzzified output value, where $x = output$ and $\mu_x = \text{degree of membership of } x$. As depicted in the fuzzy-rule surface (refer to Figure 2), a single PIF value corresponds to a collection of (ϵ, δ) combinations.

Application of perturbation on the S -dataset OptimShare generates a list of $(\epsilon$ and $\delta)$ combinations matching the input dataset's PIF_{Thresh} . With a specific $(\epsilon$ and $\delta)$ pair, it perturbs the S -dataset, generating a set number of perturbed instances that reflect the data distributions (refer to Section 3.3). Each perturbed version is $min - max$ rescaled to the original attribute $min - max$ values, then merged with the Q -dataset to create perturbed QS datasets.

Privacy analysis Our threat model assumes the worst-case scenario, with the attacker having full knowledge of the Q attributes in the perturbed $QS - dataset$, to assess residual linkage risk. We define a similarity group, SG_k , as a collection of identical records (Q) in the QS dataset. We compute the cosine similarity (CS_r^i) between original and perturbed S attributes for each record (r_i) in each SG_k . The worst-case record linkability is then defined as per Definition 2.

The Threat Model

The adversary has a complete knowledge (e.g., record order, attribute domain) of the Q attributes. This assumption leads to a worst-case linkage risk by enabling the adversary to explore the linkability of the records through Q (and perturbed S) attributes based on the tuple similarity. The knowledge acquired will subsequently be leveraged by the adversary to extract the sensitive information of individuals.

Definition 2 (Record linkability).

Let R be the set of all rows in the perturbed (P) and original (D) datasets. If $q^\alpha = q^\beta$ for some $\alpha, \beta \in R$ and $q \in Q$, take $(q^\alpha, s^\alpha) \in SG$. For each $SG_k \in SG$ compute CS_k^i for some $i \in R_{SG_k}$, where R_{SG_k} is all records in SG_k . If $CS_k^i \leq CS_k^j \forall j \in R_{SG_k}$, then $CS_k^i \in L$, where L is the set of linkable records.

Theorem 1. For any $\alpha, \beta \in R$ such that $q^\alpha = q^\beta$ for some $q \in Q$, the probability that (q^α, s^α) and (q^β, s^β) are in the same similarity group and (q^α, s^α) is linkable is small. Refer to Section 8.1, Proof 1, for the proof.

Theorem 2. OptimShare framework satisfies ϵ -differential privacy when the following inequality holds. Refer to Section 8.1, Proof 2, for the proof.

$$\frac{P[(q^\alpha, s^\alpha) \in SG \wedge CS_k^i \leq CS_k^j \forall j \in R_{SG_k}]}{P[(q^\alpha, s^\beta) \in SG \wedge CS_k^i \leq CS_k^j \forall j \in R_{SG_k}]} \leq e^\varepsilon$$

Analysis of Utility and Effectiveness in Data Perturbation The utility can be measured based on any measurement such as accuracy, precision, recall, and ROC area (KL -divergence for generic scenarios) normalized within $[0,1]$. Consider KL_x as the KL -divergence between a perturbed attribute, $x_i^p \in S$, and its unperturbed version, x_i . The maximum KL_x is the dataset’s KL -divergence, indicating the highest distribution difference. The utility loss U_l quantifies the utility reduction resulting from data perturbation, given an original utility U_o and a utility U_p after the perturbation.

The effectiveness of perturbation is gauged by the normalized residual linkage leak P_N and the ε -threshold T_ε set by the OptimShare curator. The dataset is not suitable for release if P_N is too high, which is calculated as $\frac{\varepsilon L}{T_\varepsilon}$ if $T_\varepsilon > \varepsilon L$, or 1 otherwise, where L represents linkable records.

The effectiveness loss (E_l) of a perturbed dataset is defined as a weighted measure of U_l and P_N , calculated by $E_l = C U_l + (1 - C) P_N$. Here, C determines the emphasis on linkage protection (high C) versus utility preservation (low C). The ranges of E_l are dependent on P_N and U_l values: For Low P_N and Low U_l : E_l is in $[0, C]$. For High P_N , low U_l : E_l is in $[C, 1]$. For Low P_N , high U_l : E_l is in $[1 - C, 1]$. For High P_N and High U_l : E_l is in $[C, 1]$. In our study, we set C to 0.5 to treat residual linkability leak and utility as equally crucial.

4 Results and Discussion

This section outlines the process of implementing OptimShare as a live tool (a usable product in the real world) and setting up the experiments. Additionally, we discuss the intermediate steps and dynamics of OptimShare.

4.1 Implementation

We developed two versions of OptimShare (using Python 3.8): a server-based for large-scale settings and a stand-alone for single-computer use. Figure 8 and 9 show the screen captures of the stand-alone and the server version.

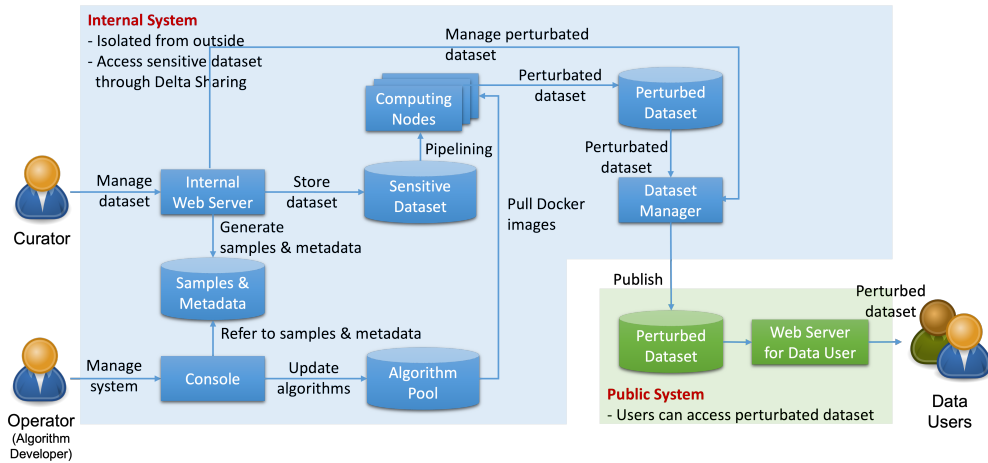


Fig. 3: System design of the OptimShare server-based version

OptimShare server-based version Figure 3 outlines a server-based system design with three user roles: curator (the data custodian), operator (admin), and data user, each with distinct privileges. Curators own and manage original datasets, applying OptimShare data perturbation, auditing, and publishing perturbed datasets for data users. Operators, as administrators, manage the algorithms while being restricted from accessing the original datasets. Data users consume the perturbed datasets approved by curators. The system ensures security and data privacy by allowing dataset owners exclusive control and

isolating servers from external access. OptimShare uses Docker containers to store the privacy-preserving algorithm for scalability and continuous integration and deployment (CI/CD). The dataset manager then pushes the published datasets to the public system, where data users can only access approved, perturbed datasets.

4.2 Experiments

This section discusses the generation of perturbed privacy-preserving datasets using the datasets and configurations mentioned in Table 1. The experiments were performed on an Apple MacBook Pro with an M1 Max and 32GB of RAM, with all plots generated automatically by our live tool (see Section 4.1).

Table 1: Datasets used for the experiments. Note: all the datasets are tabular.

Dataset	Abbr.	Records	Attributes	Classes	Global Q Attributes
NHANES diabetes Kaggle ¹	NHDS	4,412	17	2	'BPQ020', 'RIAGENDR', 'ALQ120Q', 'LBXTC'
Wine Quality ²	WQDS	4,898	12	7	'free sulfur dioxide', 'total sulfur dioxide'
Page Blocks Classification ³	PBDS	5,473	11	5	'at1', 'at2', 'at10'
Letter Recognition ⁴	LRDS	20,000	17	26	'lettr', 'x-box', 'y-box', 'width', 'high', 'xy2br'
Statlog (Shuttle) ⁵	SSDS	58,000	9	7	'b', 'd', 'i'
Credit Score Kaggle ⁶	CSDS	150,000	11	2	'ID', '#ofOCLL', '#ofT90DL', '#RELL', '#ofT60DPDNW', '#ofDependents'

The configurations of OptimShare In the experiments, the primary parameters for OptimShare were set as follows: $T_\epsilon = 8$, $P_l = 0.01\%$ ($\delta = (1/(100 \times \text{number of rows of } D)) \times P_l$), $TN_{\epsilon,\delta} = 12$, $t = 4$, $A = \text{"classification - GaussianNB"}$, $C = 0.5$, $E^T = 0.5$ (see Section 3 for parameter details). Global Q attributes used for each dataset are provided in Table 1. All settings remained constant in all experiments, ensuring uniformity for unbiased results. DP-WGAN (focusing non-categorical attributes) and PrivatePGM (focusing categorical attributes) were used for S data perturbation.

Table 2: Experiment results

Dataset	# of Records	Total Processing Time (sec)	Record Processing Time (ms)	Average Utility	Average Effectiveness
NHDS	4,412	517.6	117.3	0.784	0.725
WQDS	4,898	471.1	96.2	0.801	0.900
PBDS	5,473	546.6	99.9	0.717	0.859
LRDS	20,000	4600.9	230.0	0.942	0.971
SSDS	58,000	11,579.3	199.6	0.925	0.962
CSDS	150,000	49,867.2	332.4	0.806	0.903

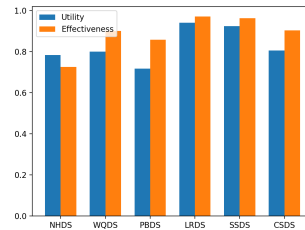


Fig. 4: Average utilities and effectivenesses

Results Table 2 displays the processing time for each dataset and the averages of utilities and effectivenesses for the privacy-preserving datasets generated by OptimShare, while Figure 4 plots the averages of utilities and effectivenesses. The datasets exhibit high effectiveness due to high utility and minimal residual data linkability. The average time complexity of the OptimShare algorithm is $O(nl)$, where n is the number of records and l is the product of TS and t . The training time of DP-WGAN and PrivatePGM models increases according to the number of records. Thus, as the number of records increased, the total processing time increased accordingly.

4.3 Dynamics of OptimShare Algorithmic Steps

In this section, we discuss the intermediate steps involved in OptimShare’s process for generating and releasing a private dataset. By understanding these steps, we can gain a comprehensive understanding of the experimental dynamics behind the process. As discussed in Section 3, one of the fundamental

¹<https://www.kaggle.com/cdc/national-health-and-nutrition-examination-survey>

²<https://archive.ics.uci.edu/ml/datasets/Wine+Quality>

³<https://archive.ics.uci.edu/ml/datasets/Page+Blocks+Classification>

⁴<https://archive.ics.uci.edu/ml/datasets/Letter+Recognition>

⁵<https://archive.ics.uci.edu/ml/datasets/Statlog+%28Shuttle%29>

⁶<https://www.kaggle.com/c/GiveMeSomeCredit/data?select=cs-training.csv>

components of OptimShare is the determination of privacy requirements. This is done through *PIF* analysis, as explained in Section 3. As shown in Figure 5a, the NHDS dataset (refer to Table 1) shows extreme *CSF* values (represented by dark red in the heatmap) in certain attributes (e.g., BMXBMI, BMXHT), whereas certain other attributes such as BPQ020 shows lower *CSF* values (represented by green). This is due to the introduction of BMXBMI drastically reducing the overall indistinguishability of the tuples in the dataset. However, BPQ020, among other attributes in the dataset, has much less impact on reducing the tuple indistinguishability. Hence, the comparison between Figures 5a and 5b provides a clear indication of the intuition behind the *PIF* value generation. As shown in Figure 5b, higher *PIF* values indicate higher levels of distinguishability (or *PIF* leak) compared to the other attributes.

The separate analysis of the *Q* attributes (represented by the red bars in Figure 5b) provides a clearer understanding of their impact on *PIF* values compared to when they are introduced to the *S* attributes, as demonstrated in Figures 5c and 5d. It is clear that *PIF* values of the attributes LBXTC and ALQ120Q significantly increase when they are introduced to the *S* attributes.

Figure 6 shows the *CSF* and *PIF* dynamics of the refined set of *Q* attributes. As depicted by the plots, OptimShare has identified that LBXTC and ALQ120Q should be removed from the set of *Q* attributes as they leak too much information when released with no perturbation. Hence, LBXTC and ALQ120Q are automatically considered as sensitive attributes and moved to the set of *S* attributes. As shown in the plots (refer to Figure 6), the refined *Q* attributes show minimal data distinguishability, producing more homogeneity in the refined *Q*-dataset tuples. This result, in turn, supports the application of less perturbation on the *S*-dataset compared to the previous non-refined *Q* attribute set.

Figure 7 shows the utility and effectiveness variations of the 12 datasets produced for the twelve ϵ, δ combinations ($TN_{\epsilon, \delta} = 12$). As Figures 7a and 7b show, the utility and effectiveness of the dataset are almost similar. This is due to the corresponding datasets producing much lower normalized residual linkage leak (P_N) than the utility values. This also suggests that OptimShare effectively refined the *Q* attribute, so the datasets can still maintain a lower residual linkage leak.

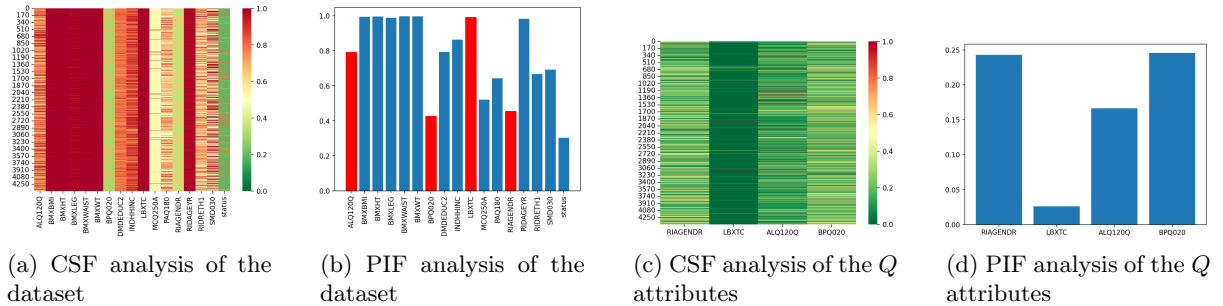


Fig. 5: The CSF and PIF analysis of the input dataset and the CSF and PIF analysis of the *Q* attributes. Note: The red bars in (b) represent the *Q* attributes.

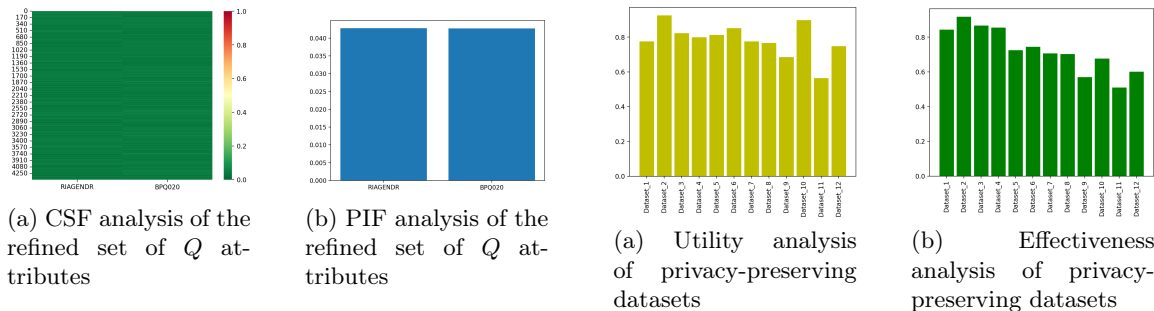


Fig. 6: The CSF and PIF analysis of the refined set of *Q* attributes

Fig. 7: A comparison between utility and effectiveness of the privacy-preserving datasets generated by OptimShare

We compared DP-WGAN and OptimShare using the NHDS dataset, applying the configurations from Section 4.2 to generate privacy-preserving datasets. Upon comparing these with the original dataset, the mean and standard deviation of two Q -attributes (BPQ020 and RIAGENDR) were evaluated. DP-WGAN’s perturbation of the entire dataset largely destroys individual statistics for these attributes, whereas OptimShare preserves them, yielding nearly identical utility to the original except for a small discrepancy, mostly due to replacing missing values. We also measured the Naïve Bayes classification performance across the three datasets using RMSE, precision, and recall. Interestingly, DP-WGAN outperforms the original dataset due to the beneficial effect of perturbation on the dataset’s distribution. However, OptimShare still delivers good performance on all three metrics, benefiting from perturbation and the preservation of certain attributes. This indicates OptimShare’s high utility and potential applicability in scenarios requiring accurate global statistics, such as tracking exact COVID-19 case distribution across postcodes, assuming postcode attributes meet OptimShare’s indistinguishability requirements.

Table 3: Comparison of results of OptimShare against direct perturbation by DP-WGAN. The following experiments were carried out under the same configurations explained in Section 4.2. NB represents Naïve Bayes classification, and std. represents the standard deviation.

	Original	DP-WGAN	OptimShare
BPQ020_mean	0.0117	6.72105	0.0079
RIAGENDR_mean	1.30235	1.66805	1.35855
BPQ020_std.	0.6667	0.97205	0.6667
RIAGENDR_std.	0.45205	0.1451	0.47925
NB_RMISE	0.4015	0.4125	0.4153
NB_precision	0.760	0.780	0.811
NB_recall	0.789	0.794	0.798

5 Related Works

Literature shows a few attempts to utilize data perturbation to solve privacy issues in tabular data-sharing (non-interactive data sharing) for different application-specific scenarios. Two of the primary advantages of data perturbation against cryptographic protocols are efficiency and scalability. Examples of data perturbation techniques include additive perturbation, random rotation, geometric perturbation, randomized response, random projection, microaggregation, hybrid perturbation, data condensation, data wrapping, data rounding, and data swapping [11,16,20]. However, the utilization of these perturbation techniques is often intended for one application (e.g., histogram analysis, deep learning), which restricts the utility of the corresponding perturbation approach to one dedicated task. Hence, the generalizability of a perturbation mechanism has not been of fundamental focus, and the practicality of these approaches for real-world applications has been a challenge. The literature does not show many mechanisms that have been developed to investigate the tradeoff between utility and privacy [33] related to tabular data perturbation, towards supporting practical utility (i.e., not restricting the utility to one application). Bertino et al.’s framework for evaluating privacy-preserving data mining algorithms is one of the few approaches developed to evaluate the balance between privacy and utility. However, their approach is more of a perturbation quality evaluation approach than an approach to improve the usability (practical utility) of data perturbation approaches [4]. FRAPP is another solution that provides matrix-theoretic framework-based solutions for random perturbation schemes [2]. Thuraisingham et al. attempted to develop insights into balancing privacy and utility during privacy preservation [30]. Although these approaches are insightful, they did not specifically answer the usability aspect of a perturbation mechanism in the real-world setting. Although a few other framework-based solutions, such as PSI (Ψ [17]), investigate the generalizability of data sharing with high privacy, they often focus only on the interactive data sharing setting. Hence, it is essential to develop a unified framework-based solution to improve the practical utility of non-interactive privacy preservation mechanisms.

6 Conclusion

This paper introduces OptimShare, a unified framework-based solution for privacy-preserving tabular data sharing. Unlike existing methods that concentrate on one problem (e.g., histogram analysis), OptimShare caters to a wide range of use cases, resolving privacy and utility issues more effectively. OptimShare uniquely identifies the privacy requirements of a specific dataset through a novel approach

called the Personal Information Factor (PIF) and allows a carefully selected limited set of raw attributes to be released, adhering to differential privacy principles. OptimShare achieves this by a rigorous iterative privacy enforcement mechanism, yielding a perfect balance between privacy and utility. This is verified by the empirical evidence produced by OptimShare. Lastly, we developed both web-based and stand-alone versions of OptimShare. In particular, the web-based system focuses more on security by isolating raw datasets according to roles, and the system enables scalability and CI/CD using Docker containers.

Despite OptimShare's effective approach to handling controlled partially perturbed non-interactive data sharing (CPNDS), CPNDS still introduces new challenges around maintaining a proper balance between utility and privacy. This arises largely from the complexity of input data that presents unlimited potential scenarios, suggesting avenues for future work. We continuously examine these dynamics as part of our development and strive for ongoing OptimShare improvements.

7 Acknowledgment

The work has been supported by the Cyber Security Research Centre Limited whose activities are partially funded by the Australian Government's Cooperative Research Centres Programme.

8 Appendices

8.1 proofs

Proof 1

Proof. Consider D as an original dataset with n tuples and m attributes. Define S and Q as sets of sensitive and non-sensitive attributes in D respectively. Assume the adversary possesses complete knowledge of Q in perturbed dataset, D^p .

We define record linkability as follows. Consider R as the collection of all records in D and D^p . If $q^\alpha = q^\beta$ for some $q \in Q$ and $\alpha, \beta \in R$, then (q^α, s^α) and (q^β, s^β) are part of the same similarity group, SG . Compute the cosine similarity, CS_k^i , between original and perturbed S attributes of each record r_i in SG_k . A record is linkable if $CS_k^i \leq CS_k^j$ for all $i \in R_{SG_k}$, for some $j \in R_{SG_k}$. Denote linkable records set as L .

ε -differential privacy is satisfied if for any datasets D_1 and D_2 differing by at most one record, and any outcome o of a randomized algorithm M , the following inequality holds:

$$\frac{P[M(D_1) = o]}{P[M(D_2) = o]} \leq e^\varepsilon \quad (14)$$

Take D_1 as the original dataset and D_2 as the dataset identical to D_1 but with modified sensitive attributes in one record. Then, we can apply ε -differential privacy, showing the adversary's successful record linkage probability is minimal.

Calculate the probabilities in the inequality's numerator and denominator. The numerator's probability is the chance that D^p contains a record (q^α, s^α) in the same SG as (q^β, s^β) , and (q^α, s^α) is linkable. This is:

$$P[M(D_1) = o] = P[(q^\alpha, s^\alpha) \in SG \wedge CS_k^i \leq CS_k^j \forall j \in R_{SG_k}] \quad (15)$$

For the denominator, the probability is the chance that D^p contains a record (q^α, s^β) in the same SG as (q^β, s^β) , and (q^α, s^β) is linkable:

$$P[M(D_2) = o] = P[(q^\alpha, s^\beta) \in SG \wedge CS_k^i \leq CS_k^j \forall j \in R_{SG_k}] \quad (16)$$

Substituting into Equation 14, we get:

$$\frac{P[(q^\alpha, s^\alpha) \in SG \wedge CS_k^i \leq CS_k^j \forall j \in R_{SG_k}]}{P[(q^\alpha, s^\beta) \in SG \wedge CS_k^i \leq CS_k^j \forall j \in R_{SG_k}]} \leq e^\varepsilon \quad (17)$$

This suggests the adversary's successful record linking probability is limited, fulfilling the ε -differential privacy requirement.

Proof 2

Proof. The proof of Theorem 2 requires demonstrating the numerator and denominator of the Theorem’s Equation are small, indicating the probability of a record in a similarity group being linkable is minimal. This necessitates verifying that the perturbations on D^p ’s sensitive attributes suffice to deter successful record linking by an adversary.

This is feasible by ensuring the cosine similarity between the original and perturbed sensitive attributes of all D^p records is minimal. Lower cosine similarity complicates record linking for the adversary as it dictates the record’s linkability probability. Compliance with the privacy budget demands a negligible change in a specific outcome’s probability when a record is added or deleted, which is achievable by applying DP noise to sensitive attributes during perturbation.

The sufficiently small cosine similarity between original and perturbed attributes can be upper-bounded using record linkability (Definition 2), computing the cosine similarity for each dataset record. Complying with the privacy budget involves bounding the change in a specific outcome’s probability upon record addition or deletion.

Considering two records, (q_1, s_1) and (q_2, s'_1) , which have identical quasi-identifiers, and sensitive attributes s_1 and s'_1 , (where s'_1 is the perturbed version of s_1 , generated using an (ϵ, δ) -differentially private generator), we can compute the cosine similarity of original and perturbed sensitive attributes, showing the insignificant change in a specific outcome’s probability with record addition or deletion.

The cosine similarity between s_1 and s'_1 is calculated as:

$$CS = \frac{s_1 \cdot s'_1}{|s_1||s'_1|} \quad (18)$$

We can use the Cauchy-Schwarz inequality [6] to show that:

$$s_1 \cdot s'_1 \leq |s_1||s'_1| \quad (19)$$

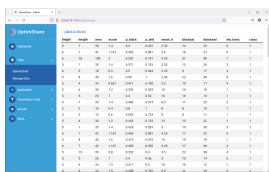
Given the constraints set by $\frac{\epsilon L}{T_\epsilon}$ (where L represents the set of linkable records), we can establish an upper bound for $|s'_1|$ to ensure that the cosine similarity is small.

For $\frac{\epsilon L}{T_\epsilon} \leq 1$, we can ensure that the added noise is within the acceptable range defined by ϵ . This limits the denominator of the cosine similarity expression to a value that’s consistent with the privacy budget, ϵ .

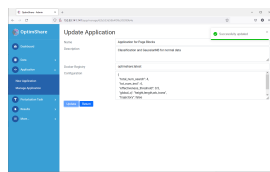
Therefore, the cosine similarity between the original and perturbed sensitive attributes is upper-bounded by a value that complies with the privacy budget ϵ , which confirms that the OptimShare framework satisfies ϵ -differential privacy.

8.2 Different Interface Views of the OptimShare Live Tool

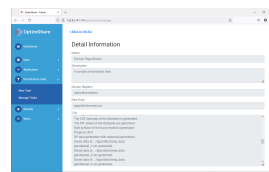
OptimShare web-based and stand-alone live tool Figure 8 and Figure 9 show the screenshots of the two versions (web-based and stand-alone) of the OptimShare live tool.



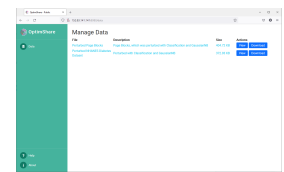
(a) View of a dataset on the server



(b) Details of a privacy-preserving algorithm



(c) Detail of a perturbation task



(d) Published dataset to data users

Fig. 8: Screenshots of the server-based OptimShare and the web server for Data Users.

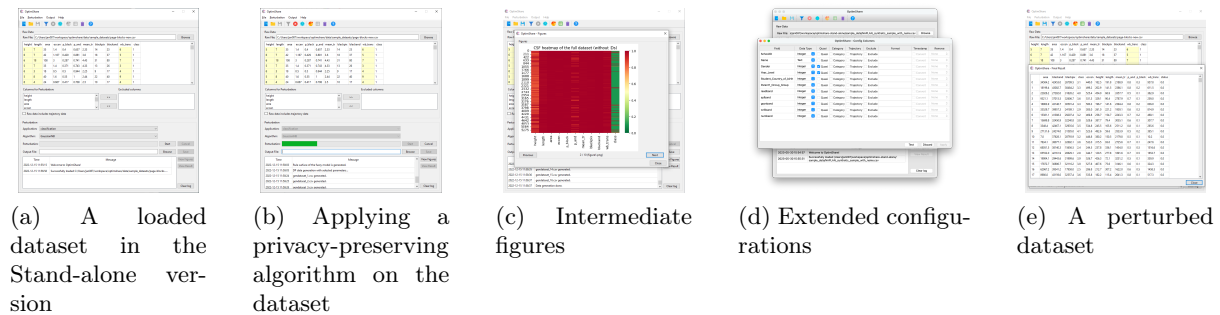


Fig. 9: Screenshots of the OptimShare stand-alone version.

References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. pp. 308–318 (2016)
2. Agrawal, S., Haritsa, J.R.: A framework for high-accuracy privacy-preserving mining. In: Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on. pp. 193–204. IEEE (2005)
3. Arachchige, P.C.M., Bertok, P., Khalil, I., Liu, D., Camtepe, S., Atiquzzaman, M.: Local differential privacy for deep learning. *IEEE Internet of Things Journal* **7**(7), 5827–5842 (2019)
4. Bertino, E., Fovino, I.N., Provenza, L.P.: A framework for evaluating privacy preserving data mining algorithms. *Data Mining and Knowledge Discovery* **11**(2), 121–154 (2005)
5. Bertino, E., Lin, D., Jiang, W.: A survey of quantification of privacy preserving data mining algorithms. In: Privacy-preserving data mining, pp. 183–205. Springer (2008)
6. Bhatia, R., Davis, C.: A cauchy-schwarz inequality for operators with applications. *Linear algebra and its applications* **223**, 119–129 (1995)
7. Bindschaedler, V., Shokri, R., Gunter, C.A.: Plausible deniability for privacy-preserving data synthesis. *Proceedings of the VLDB Endowment* **10**(5) (2017)
8. Blum, A., Ligett, K., Roth, A.: A learning theory approach to noninteractive database privacy. *Journal of the ACM (JACM)* **60**(2), 1–25 (2013)
9. Bun, M., Steinke, T.: Concentrated differential privacy: Simplifications, extensions, and lower bounds. In: *Theory of Cryptography Conference*. pp. 635–658. Springer (2016)
10. Chamikara, M.A.P., Bertók, P., Khalil, I., Liu, D., Camtepe, S.: Privacy preserving face recognition utilizing differential privacy. *Computers & Security* **97**, 101951 (2020)
11. Chamikara, M.A.P., Bertók, P., Liu, D., Camtepe, S., Khalil, I.: Efficient privacy preservation of big data for accurate data mining. *Information Sciences* **527**, 420–443 (2020)
12. Day, W.Y., Li, N.: Differentially private publishing of high-dimensional data using sensitivity control. In: *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*. pp. 451–462 (2015)
13. Dinh, D.T., Fujinami, T., Huynh, V.N.: Estimating the optimal number of clusters in categorical data clustering by silhouette coefficient. In: *International Symposium on Knowledge and Systems Sciences*. pp. 1–17. Springer (2019)
14. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) *Automata, Languages and Programming*. pp. 1–12. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
15. Dwork, C.: The differential privacy frontier. In: *Theory of Cryptography Conference*. pp. 496–502. Springer (2009)
16. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* **9**(3–4), 211–407 (2014)
17. Gaboardi, M., Honaker, J., King, G., Murtagh, J., Nissim, K., Ullman, J., Vadhan, S.: PSI ($\{\Psi\}$): a private data sharing interface. *arXiv preprint arXiv:1609.04340* (2016)
18. Ganey, G., Oprisanu, B., De Cristofaro, E.: Robin hood and matthew effects: Differential privacy has disparate impact on synthetic data. In: *International Conference on Machine Learning*. pp. 6944–6959. PMLR (2022)
19. Gupta, Y., Saini, A., Saxena, A.: A new fuzzy logic based ranking function for efficient information retrieval system. *Expert Systems with Applications* **42**(3), 1223–1234 (2015)
20. Hasan, A., Jiang, Q., Luo, J., Li, C., Chen, L.: An effective value swapping method for privacy preserving data publishing. *Security and Communication Networks* **9**(16), 3219–3228 (2016)
21. Jayaraman, B., Evans, D.: Evaluating differentially private machine learning in practice. In: *28th USENIX Security Symposium (USENIX Security 19)*. pp. 1895–1912 (2019)

22. Jordon, J., Yoon, J., Van Der Schaar, M.: Pate-gan: Generating synthetic data with differential privacy guarantees. In: International Conference on Learning Representations (2018)
23. Li, H., Xiong, L., Jiang, X.: Differentially private synthesization of multi-dimensional data using copula functions. In: Advances in database technology: proceedings. International conference on extending database technology. vol. 2014, p. 475. NIH Public Access (2014)
24. Mahawaga Arachchige, P.C., Liu, D., Camtepe, S., Nepal, S., Grobler, M., Bertok, P., Khalil, I.: Local differential privacy for federated learning. In: Computer Security–ESORICS 2022: 27th European Symposium on Research in Computer Security, Copenhagen, Denmark, September 26–30, 2022, Proceedings, Part I. pp. 195–216. Springer (2022)
25. McKenna, R., Sheldon, D., Miklau, G.: Graphical-model based estimation and inference for differential privacy. In: International Conference on Machine Learning. pp. 4435–4444. PMLR (2019)
26. Muralidhar, K., Parsa, R., Sarathy, R.: A general additive data perturbation method for database security. *management science* **45**(10), 1399–1415 (1999)
27. Oppermann, I.: Privacy-preserving data sharing frameworks. Australian Computer Society (ACS) (2019), <https://www.acs.org.au/insightsandpublications/reports-publications/privacy-preserving-data-sharing-frameworks.html>
28. Oppermann, I., Nabaglo, J., Henecka, W.: A measure of personal information in mobile data. In: 2020 2nd 6G Wireless Summit (6G SUMMIT). pp. 1–6. IEEE (2020)
29. Tao, Y., McKenna, R., Hay, M., Machanavajjhala, A., Miklau, G.: Benchmarking differentially private synthetic data generation algorithms. arXiv preprint arXiv:2112.09238 (2021)
30. Thuraisingham, B., Kantarcioglu, M., Bertino, E., Clifton, C.: Towards a framework for developing cyber privacy metrics: A vision paper. In: (BigData Congress), 2017 IEEE International Congress on Big Data. pp. 256–265. IEEE (2017)
31. Tran, V.X., Tsuji, H.: QoS based ranking for web services: Fuzzy approaches. In: 2008 4th International Conference on Next Generation Web Services Practices. pp. 77–82. IEEE (2008)
32. Xie, L., Lin, K., Wang, S., Wang, F., Zhou, J.: Differentially private generative adversarial network. arXiv preprint arXiv:1802.06739 (2018)
33. Xu, L., Jiang, C., Chen, Y., Ren, Y., Liu, K.R.: Privacy or utility in data collection? a contract theoretic approach. *IEEE Journal of Selected Topics in Signal Processing* **9**(7), 1256–1269 (2015)
34. Zhang, Z., Wang, T., Li, N., Honorio, J., Backes, M., He, S., Chen, J., Zhang, Y.: Privsyn: Differentially private data synthesis. In: 30th USENIX Security Symposium (USENIX Security 21) (2021)