# Comparative Study of Differentially Private Synthetic Data Algorithms and Evaluation Standards

Claire McKay Bowen[1] and Joshua Snoke[2]

[1]*Statistical Sciences Group, Los Alamos National Laboratory**, *claire.mckay.bowen@gmail.com*
[2]*RAND Corporation, jsnoke@rand.org*

**Abstract**

Differentially private synthetic data generation is becoming a popular solution that releases analytically useful data while preserving the privacy of individuals in the data. In order to utilize these algorithms for public policy decisions, policymakers need an accurate understanding of these algorithms' comparative performance. Correspondingly, data practitioners also require standard metrics for evaluating the analytic qualities of the synthetic data. In this paper, we present an in-depth evaluation of several differentially private synthetic data algorithms using the actual differentially private synthetic data sets created by contestants in the recent National Institute of Standards and Technology's (NIST) "Differentially Private Synthetic Data Challenge." We offer both theoretical and practical analyses of these algorithms. We frame the NIST data challenge methods within the broader differentially private synthetic data literature. In addition, we implement two of our own utility metric algorithms on the differentially private synthetic data and compare these metrics' results to the NIST data challenge outcome. Our comparative assessment of the differentially private data synthesis methods and the quality metrics shows the relative usefulness, general strengths and weaknesses, preferred choices of algorithms and metrics. Finally we give implications of our evaluation for policymakers seeking to implement differentially private synthetic data algorithms on future data products.

***Keywords***— differential privacy, synthetic data, utility, evaluation, statistical disclosure control

## 1 Introduction

### 1.1 Motivation

The collection and dissemination of data can greatly benefit society by enabling impactful research projects, such as the Personal Genome Project Canada database which determines Genomic variants in participants for several health problems (Reuter et al., 2018), the United Kingdom Medical Education Database "... to improve standards, facility workforce planning and support the regulation of medical education and training" (Dowell et al., 2018), and using facial recognition for identifying missing children in national missing person databases in various countries (Watts, 2019). However, there are often concerns over the privacy risks inherit in sharing sensitive research data,

*The publication has been assigned the Los Alamos National Laboratory identifier LA-UR-19-31841.

and recent misuses of data access for seemingly research purposes, such as the Facebook - Cambridge Analytica Scandal, have heightened data privacy concerns over how both private companies and government organizations gather and disseminate information (González et al., 2019; Martin et al., 2017; Tsay-Vogel et al., 2018).

Statistical disclosure control (SDC), or limitation (SDL), is a field of study that aims to develop methods for releasing high-quality data products while preserving the confidentiality of sensitive data. These techniques have existed within statistics and the social sciences since the mid-twentieth century, and they seek to balance risk against the benefit to society, also known as the utility of the data. While this field has existed for some time, over the past two decades the data landscape has dramatically changed. Data adversaries (also referred to as intruders or attackers) can more easily reconstruct data sets and identify individuals from supposedly anonymized data with the advances in modern information infrastructure and computational power. Some re-identification examples of anoymized data include the Netflix Prize data set (Narayanan and Shmatikov, 2008), the Washington State health records (Sweeney, 2013), credit card metadata (De Montjoye et al., 2015), cell phone spatial data (De Montjoye et al., 2013; Hardesty, 2013; Kondor et al., 2018), and the United States public use microdata files (Rocher et al., 2019).

One of the more recent innovations in SDC is a technique known as synthetic data generation, and it has become a leading method for releasing publicly available data that can be used for numerous different analyses (Drechsler, 2011; Little, 1993; Raab et al., 2016; Raghunathan et al., 2003; Reiter, 2005; Rubin, 1993). While this approach has been shown to offer improvements in the risk-utility trade-off compared with early methods, the main limitation with this approach remains, the same as other SDC approaches, the lack of a formal privacy quantification. SDC methods require strong assumptions concerning the knowledge and identification strategies of the attacker. The risk is then estimated by simulating these attackers. Due of the increased availability to external data files and methods of reconstructing information from data, it is less reasonable in many cases to make these assumptions, and it is harder to simulate all types of plausible adversaries. This issue is a common critique for SDC methods that rely on specific data values and assumptions on the data adversary's behavior and background knowledge (Hundepool et al., 2012; Manrique-Vallier and Reiter, 2012; Reiter, 2005).

In the past two decades, a new approach has been developed known as differential privacy (DP). This idea originated in the theoretical computer science community, and Dwork et al. (2006b) proposed the first formal definition of quantifying the privacy loss when releasing information from a confidential data set. In contrast to SDC methods, this theory does not require a simulated attacker or the same strong assumptions concerning how much information an intruder may have or what kind of disclosure is occurring. At a high level, DP links the potential for privacy loss to how much the answer of a statistic (or query) is changed given the absence or presence of the most extreme possible person in the population within the data. The level of protection required is proportional to this maximum potential change that could be made of the person. For further details, Dwork and Roth (2014) provides a rigorous mathematical review of DP while Nissim et al. (2017) and Snoke and Bowen (2019) describe DP for a non-technical, general audience. Since its conception, DP has created an entire new field of research with applications in Bayesian learning (Wang et al., 2015), data mining (Mohammed et al., 2011), data streaming (Dwork et al., 2010), dimension reduction (Chaudhuri et al., 2012), eye tracking (Liu et al., 2019), genetic associate tests (Yu et al., 2014), various statistical analyses (Karwa et al., 2016; Wasserman and Zhou, 2010), power grid obfuscation (Fioretto et al., 2019), and recommender systems (Friedman et al., 2016) to list a few.

Within the large body of DP literature, the combination of DP and data synthesis has more recently been considered a solution to releasing analytically useful data while preserving the privacy of individuals in the data. Applications include binary data (Charest, 2011; McClure and Reiter, 2012), categorical data (Abowd and Vilhuber, 2008; Hay et al., 2010), continuous data (Bowen and Liu, 2016; Snoke and Slavković, 2018; Wasserman and Zhou, 2010), network data (Karwa et al., 2017, 2016), and Poisson distributed data (Quick, 2019). In order to utilize these algorithms for public policy decisions, policymakers need an accurate understanding of these algorithms' comparative performance. Correspondingly, data practitioners also require standard metrics for evaluating the analytic qualities of the synthetic data.

### 1.2 Contribution

In this paper, we provide an in-depth assessment of various differentially private data synthesis methods from the 2018 National Institute of Standards and Technology's (NIST) "Differentially Private Synthetic Data Challenge" (Vendetti, 2018). The challenge consisted of three "Marathon Matches," which spanned from November 2018 to May 2019. Each match provided the contestants with a data set to train and develop their DP methods that were identical in structure and variables to the test data used for final scoring. Contestants were also given details regarding how their methods would be evaluated. Participants had 30 days from the start of each match to develop and submit their differentially private synthetic data approaches. Over the 30 day period, a panel of subject matter experts reviewed and verified that the submitted methods satisfied DP. If approved, the differentially private synthetic data methods were applied to the test data for final scoring.

The challenge called for researchers to develop practical and viable differentially private data synthesis methods that were then tested for accuracy on summary statistics, classification tasks, and regression models. Detailed proofs and code were required for the submissions, and the highest scoring submissions received cash prizes. Accordingly, we evaluate the approaches based on their performance in the challenge, their ease of implementation, and their current standing as formalized concepts in the literature. We provide recommendations for which methods are the best candidates for future use, with details descriptions of their strengths and weaknesses.

Additionally, using the actual synthetic data sets generated by contestants in the challenge, which were made available to us by NIST, we implement two utility metric algorithms on the differentially private synthetic data to compare if our metric results will predict the NIST Data Challenge outcome. Our evaluation of the differentially private synthetic data algorithms and metric standards will cover the relative usefulness (ranging from specific measures of model accuracy to general measures of distributional similarity), general strengths and weaknesses, preferred choices of the algorithms and metrics, as well as implications of our evaluation for policymakers seeking to implement differentially private synthetic data algorithms on future data products.

We organize the remainder of the paper as follows. Section 2 reviews the definitions and concepts of differential privacy and common differentially private mechanisms. Section 3 summarizes the differentially private data synthesis methods ranked in the NIST Data Challenge, and Section 4 describes the quality metrics we developed to compare the NIST Data Challenge outcome. Section 5 evaluates and compares the quality metric results to the how the NIST Data Challenge ranked the competitors. Concluding remarks and suggestions for future work are given in Section 6.

## 2 Differential Privacy

In contrast to past SDC methods, differential privacy (DP) offers privacy protection with a provable and quantifiable amount, colloquially referred to as the privacy-loss budget. It is important to note that DP is a statement about the algorithm (or mechanism), *not* a statement about the data. Rather than stating that the output data meets privacy requirements, DP requires that the *algorithm* which produces the output provably meets the definitions. Accordingly, algorithms which satisfy the definitions are referred to as differentially private algorithms.

In this section, we reproduce the pertinent definitions and theorems of DP with the following notation: $X \in \mathbb{R}$ is the original data set with dimension $n \times q$ and $X^*$ is the private version of $X$ with dimension $n^* \times q$. We also define a statistical query as a function $u : \mathbb{R}^{n \times q} \to \mathbb{R}^k$, where the function maps the possible data sets of $X$ to $k$ real numbers.

### 2.1 Definitions and Theorems

**Definition 1.** ***Differential Privacy*** *(Dwork et al., 2006b): A sanitization algorithm, $\mathcal{M}$, gives $\epsilon$-DP if for all subsets $S \subseteq Range(\mathcal{M})$ and for all $X, X'$ such that $d(X, X') = 1$,*

$$\frac{\Pr(\mathcal{M}(X) \in S)}{\Pr(\mathcal{M}(X') \in S)} \leq \exp(\epsilon) \tag{1}$$

where $\epsilon > 0$ is the privacy-loss budget and $d(X, X') = 1$ represents the possible ways that $X'$ differs from $X$ by one record. We define this difference as a presence or absence of a record, but note that some definitions of DP has this difference as a change, where $X$ and $X'$ have the same dimensions.

One concern about algorithms that satisfy $\epsilon$-DP is they tend to inject a large amount of noise to statistical query results to attain a strong privacy guarantee. Several relaxations of $\epsilon$-DP have been developed such as approximate DP (Dwork et al., 2006a), probabilistic DP (Machanavajjhala et al., 2008), and concentrated DP (Dwork and Rothblum, 2016). These are called relaxations because, while still formal, they offer slightly weaker privacy guarantees. In return, they typically lessen the amount of noise required. We will only cover approximate DP, also known as $(\epsilon, \delta)$-DP, since the NIST Data Challenge required the differentially private data synthesis submissions to satisfy $(\epsilon, \delta)$-DP rather than strict $\epsilon$-DP.

**Definition 2.** $(\epsilon, \delta)$***-Differential Privacy*** *(Dwork et al., 2006a): A sanitization algorithm $\mathcal{M}$ gives $(\epsilon, \delta)$-DP if for all $X, X'$ that are $d(X, X') = 1$,*

$$\Pr(\mathcal{M}(X) \in S) \leq \exp(\epsilon) \Pr(\mathcal{M}(X') \in S) + \delta \tag{2}$$

*where $\delta \in [0, 1]$. Pure $\epsilon$-DP is a special case of $(\epsilon, \delta)$-DP when $\delta = 0$.*

The parameter $\delta$ adds a small probability that the bound given in Definition 1 does not hold, which can be useful when dealing with extreme yet very unlikely cases.

In general, if the data maintainer (or curator) "spends" more of the privacy-loss budget (runs the algorithm with a larger value of $\epsilon$), the data practitioners (those who access the noisy data or statistics) should gain more accurate information from the data. With DP, $\epsilon$ formally parameterizes the trade-off between accuracy and privacy-loss. For a given algorithm, greater accuracy leads to less privacy guarantee due to information being "leaked" from the data. Inversely, the data maintainer could spend a smaller amount of the privacy-loss budget, which will result in less accurate information and greater privacy protection. The goal in devising differentially private

algorithms is to optimize this trade-off in order to offer the best accuracy for a given amount of privacy-loss.

Although DP has existed for over a decade, there is no general consensus on what value of $\epsilon$, the amount of the privacy-loss budget, should be used for practical implementation. The decision is considered a social and policy question that will need stakeholders to shoulder the responsibility of the decision. In addition, the opinions of those involved in creating the DP algorithms (e.g., statisticians or computer scientists) and the participants in the data will have their own opinions on what an appropriate privacy-loss budget is. A full discussion on this topic is beyond the scope of this paper, but we will examine the affects of $\epsilon$ on accuracy within the context of the NIST Data Challenge.

Many DP algorithms have multiple outputs, which is common for dealing with high-dimensional data or a query system that can be repeatedly utilized. Each time a statistic or output is released, data information is being "leaked", and therefore needs protected. This is done by splitting the amount $\epsilon$ used for each output, and the composition theorems formalize how to protect multiple statistics.

**Theorem 1.** ***Composition Theorems*** *(McSherry, 2009):*

*Suppose a mechanism, $\mathcal{M}$, provides $\epsilon_j$-DP for $j = 1, \ldots, k$.*

*a)* ***Sequential Composition:***
*The sequence of $\mathcal{M}_j(X)$ applied on the same $X$ provides $(\sum_j \epsilon_j, \sum_j \delta_j)$-DP.*

*b)* ***Parallel Composition:***
*Let $D_j$ be disjoint subsets of the input domain $D$. The sequence of $\mathcal{M}_j(X \cap D_j)$ provides $\max(\epsilon_j, \delta_j)$-DP.*

To put it more generally, suppose there are $k$ many statistical queries on $X$. These theorems state that the data maintainer must allocate a portion of the overall desired level of $\epsilon$ to each statistic by Sequential Composition. A typical appropriation is equally dividing up $\epsilon$ by $k$. Another situation for Sequential Composition is when the data maintainer might want to make multiple draws of the statistic of interest to generate multiple differentially private synthetic data sets. In the case of Parallel Composition, if the $k$ statistical queries are applied to a disjoint set of $X$, then there is no additional privacy-loss. An example is a histogram, where the bins are disjoint subsets of the data, so noise can be added to each of them without needing to split $\epsilon$.

Another important theorem is the post-processing theorem that states that any function applied a differentially private output is also differentially private.

**Theorem 2.** ***Post-Processing Theorem*** *(Dwork et al., 2006b; Nissim et al., 2007):*

*If $\mathcal{M}$ be a mechanism that satisfies $\epsilon$-DP, and $g$ be any function, then $g(\mathcal{M}(X))$ also satisfies $\epsilon$-DP.*

This theorem is particularly useful for differentially private synthetic data algorithms, since many of those algorithms focus on perturbing the parameters of the distribution from which the synthetic data will be drawn. Because of post-processing, it is sufficient to perturb the parameters, and any data drawn based on the noisy parameters will also be differentially private. Additionally, post-processing is vital for many algorithms to improve the accuracy of the final output. For example, enforcing structural aspects of the data, such as not outputing negative values for people's ages, can be handled by post-processing. We will refer to post-processing for almost every contestants' algorithm described in Section 3.

## 2.2 Differentially Private Mechanisms

Algorithms which add sufficient noise to the released data or queries such that they satisfy the differential privacy definitions are commonly referred to as mechanisms. In this section, we present some building-block mechanisms which are used in the $\epsilon$-DP and $(\epsilon, \delta)$-DP algorithms developed by the competitors for the NIST Data Challenge.

For a given value of $\epsilon$ (the privacy-loss budget), an algorithm that satisfies DP or approximate DP will adjust the amount of noise being added to the data based on the maximum possible change, given two databases that differ by one row, of the statistic or data that the data practitioner wants released. This value is commonly referred to as the global sensitivity (GS), given in Definition 3.

**Definition 3.** *$l_1$-Global Sensitivity (Dwork et al., 2006b): For all $X, X'$ such that $d(X, X') = 1$, the global sensitivity of a function $u$ is*

$$\Delta_1 u = \sup_{d(X,X')=1} \|u(X) - u(X')\|_1 \tag{3}$$

We can calculate this sensitivity under different norms, for example $\Delta_2 u$ represents the $l_2$ norm global sensitivity, $l_2$-GS, of the function $u$. Another way of thinking about this value is that it measures how robust the statistical query is to outliers.

The most basic mechanism satisfying $\epsilon$-DP is the Laplace Mechanism, given in Definition 4, which was first introduced by Dwork et al. (2006b).

**Definition 4.** *Laplace Mechanism (Dwork et al., 2006b): The Laplace Mechanism satisfies $\epsilon$-DP by adding noise to $u$ that are drawn from a Laplace distribution with the location parameter at 0 and scale parameter of $\Delta_u \epsilon^{-1}$ such that*

$$u^*(X) = u(X) + Laplace\left(0, \Delta_1 u \epsilon^{-1}\right) \tag{4}$$

Mean zero noise is added to $u(X)$, and the variance of that noise is increased as $\epsilon$ decreases or $\Delta_1 u$ increases. In other words, more noise is added for using a smaller privacy-loss budget or for releasing information that is less robust to outliers. Another popular mechanism is the Gaussian Mechanism that satisfies $(\epsilon, \delta)$-DP, given in Definition 5, but relies on the $l_2$-GS of the statistical query.

**Definition 5.** *Gaussian Mechanism (Dwork and Roth, 2014): The Gaussian Mechanism satisfies $(\epsilon, \delta)$-DP by adding Gaussian noise with zero mean and variance, $\sigma^2$, such that*

$$u^*(X) = u(X) + N\left(0, \sigma^2 I\right) \tag{5}$$

*where $\sigma = \Delta_2 u \epsilon^{-1} \sqrt{2 \log(1.25/\delta)}$.*

Both the Laplace and Gaussian Mechanisms are simple and quick to implement, but only apply to numerical values (without additional post-processing, Theorem 2). A more general $\epsilon$-DP mechanism is the Exponential Mechanism, given in Definition 6, which allows for the sampling of values from a noisy distribution rather than adding noise directly. This mechanism was not used by any of the top ranking participants.

**Definition 6.** *Exponential Mechanism (McSherry and Talwar, 2007): The Exponential mechanism releases values with a probability proportional to*

$$\exp\left(\frac{\epsilon u(X, \theta)}{2\Delta_u}\right) \tag{6}$$

*and satisfies $\epsilon$-DP, where $u(X, \theta)$ is the score function that measures the quality of all the possible outputs of $\theta$ on $X$.*

Although the Exponential Mechanism has the nice flexibility that it can apply to any type of statistic, many theoretical algorithms using the Exponential Mechanism are computationally infeasible for practical applications without limiting the possible outputs of $\theta$.

## 3 Differentially Private Data Synthesis Algorithms

In this section, we review the top ranking differentially private data synthesis algorithms from the NIST Data Challenge. Hay et al. (2016) and Bowen and Liu (2016) offer in-depth evaluations and assessments of differentially private data synthesis methods beyond what we will cover in this paper, so we direct any interested readers to these papers for more information.

For the NIST Data Challenge, competitors were given a public data set to help develop their algorithms before submitting their code for final scoring on the test data. The public and challenge data had identical data structures and variables, so the code could be adequately prepared to run without error. Both Matches #1 and #2 used the San Francisco Fire Departments (SFFDs) Call for Service data, but at different years. These data sets contained a total of 32 categorical and continuous variables with roughly 236,000 to 314,000 observations respectively. Some of the variables are *Call Type Group*, *Number of Alarms*, *City*, *Zip Code of Incident*, *Neighborhood*, *Emergency Call Received Date and Time*, *Emergency Call Response Date and Time*, *Supervisor District*, and *Station Area*. For Match #3, challenge participants trained their methods on the Colorado Public Use Microdata Sample (PUMS) data, and their methods were evaluated on the Arizona and Vermont PUMS data for final scoring. All three PUMS data sets had 98 categorical and continuous variables with the number of observations ranging from about 210,000 to 662,000. *Gender*, *Race*, *Age*, *City*, *City Population*, *School*, *Veteran Status*, *Rent*, and *Income Wage* are a few of the 98 variables. We discuss how the NIST Differentially Private Synthetic Data Challenge executed their scoring in Section 4.

We categorize the differentially private data synthesis methods from the challenge into the same two categories used in Bowen and Liu (2016), non-parametric and parametric approaches. We define non-parametric approaches as differentially private data synthesis methods that generate data from an empirical distribution, and we define parametric approaches as methods that generate the synthetic data from a parametric distribution or a generative model.

### 3.1 Non-parametric Data Synthesis

Most non-parametric differentially private synthetic data techniques work by sanitizing the cell counts or proportions of a cross-tabulation of the data (e.g., the full cross-tabulation of all variables). For categorical data, the noisy tables can be released. To provide a synthetic microdata file, or when the original data has continuous variables, the non-parametric approaches will sample data from a uniform distribution for each of the discretized bins. The bounds for the discretization of continuous variables must be selected in a differentially private manner or by leveraging public information to satisfy DP and is often a tricky part of these methods. The majority of the teams who developed non-parametric data synthesis methods focused on reducing the number of cells to sanitize by using techniques such as clustering variables (i.e. creating multiple cross-tabulations on a subset of the variables), modeling highly correlated marginals, or using the privacy budget asymmetrically across cells. In Table 1, we list and summarize the non-parametric differentially private data synthesis methods covered in this section.

*Table 1: Summary of the non-parametric differentially private synthetic data approaches discussed in Section 3.1.*

| Team | Computation | Off-the-Shelf vs. Hand-Coding | Pre- and Post-Processing |
|---|---|---|---|
| Team DPSyn (Sec. 3.1.1) | light to moderate computational complexity | some hand-coding due to identifying marginals for pre-processing, *Python* code available on GitHub | **pre-processing:** identify marginals from public data; **post-processing:** adjust noisy marginals to be consistent and change counts to zero below a threshold |
| Team Gardn999 (Sec. 3.1.2) | simplest and fastest method | some hand-coding due to identifying marginals for pre-processing, *Java* code available on GitHub | **pre-processing:** identify marginals from public data; **post-processing:** adjust the overall counts based on a threshold to avoid a large number of non-zero bins |
| Team pfr (Sec. 3.1.3) | simple and quick after pre-processing | hand-coding required, no public code available | **pre-processing:** identify marginals from public data; **post-processing:** reduce the number of non-empty cells from sanitization by modeling the noisy cell counts |

### 3.1.1 Team DPSyn

DPSyn consistently performed well throughout the entire NIST Data Challenge, placing second in all three matches. Team DPSyn's method (DPSyn) works by clustering the variables (c similar variables) and perturbing the cell counts of the joint histograms for each cluster. This lessens the noise necessary because it reduces the total number of cells, but it pays for that gain by sacrificing the correlations between variables in different clusters. Identifying the correlations from the public data, DPSyn constructs the 1-, 2-, and 3-way marginals for all variables in each cluster, and sanitizes the counts via the Gaussian Mechanism. For post-processing, these noisy marginals are then constrained using techniques from Qardaji et al. (2014) to be consistent with one another. These techniques essentially check for mutual consistency among the totals of the multi-way marginals (altering the counts to be consistent) and reduce the noisy counts to zero when they are below a threshold. Finally, DPSyn generates the synthetic data by sampling from the noisy marginals of the joined clusters.

The algorithm is straightforward, and a data maintainer could implement DPSyn fairly easily given the simplicity and because Li et al. (2019) provided the source code (`Python`) and full documentation on GitHub. The main difficulty would be selecting the variable groups for the pre-processing step, which could be daunting for an inexperienced data maintainer or someone without familiarity of the data set. This method is fairly novel, only being published recently, so it has yet to gain wide acceptance in the field. That being said, it's simplicity and good performance is likely to lead to others implementing it.

### 3.1.2 Team Gardn999

Team Gardn999 developed the simplest method, DPFieldGroups, out of all the NIST Data Challenge entrants while still performing well. They placed fifth and fourth place in Matches #2 and #3, respectively, while they did not participate in Match #1. DPFieldGroups sanitizes the original data cell counts via the Laplace Mechanism. Similarly to DPSyn, the cells are first clustered by identifying the highly correlated variables from the public, training data set. In addition, Team Gardn999 conducts post-processing by reducing noisy counts to zero if they were below a threshold that is calculated from $\epsilon$ and the $\log_{10}$ number of bins in the particular marginal histogram. DPFieldGroups generates the synthetic data by randomly sampling the sanitized observations from each of the marginal histograms with a weighted probability proportional to the noisy counts.

Similarly to DPSyn, the pre-processing step for DPFieldGroups relies on the data maintainer to cluster highly correlated variables based on public data. Once the variables are grouped, the data maintainer can execute the `Java` code hosted on GitHub (Gardner, 2019). The post-processing step is less involved than DpSyn, only adjusting some counts down to avoid a large number of non-zero bins. The strength of this approach lies in its simplicity. On the other hand, it has not been published as a novel method and relies only on relatively simple DP steps. This method forms a good case study for the performance of a simple application of differential privacy.

### 3.1.3 Team pfr

Team pfr placed first in Matches #1 and #2, but did not compete in Match #3. Their lack of participation might be due to them initially designing their algorithm based on how Match #1 scored the similarity of the original and synthetic data sets. Specifically, they targeted maximizing accuracy on the 3-way marginal counts, and the variables that involve the Emergency Call Data and Time information. Before sanitizing the 3-way marginals, their pre-processing step depends on the data maintainer to establish a list of:

1. variables that could be computed deterministically from other variables and therefore did not need to be encoded. e.g., *City* variable computed deterministically from *Neighborhood* variable.
2. histogram queries, which the data maintainer may optionally identify which variables were correlated to determine a subset of the domain. e.g., *Supervisor District* is correlated with *Station Area*.
3. data set size threshold for certain queries, where the query is discarded and the corresponding output is replaced by a uniform distribution.

They then roughly cluster the variables into three disjoint groups (Spatial, Temporal, and Call-Information groups), and they identify within each group which variables could be computed deterministically from other variables and which variables were highly correlated. Clustering the variables in this manner, they reduce the possible combinations of cells that needed to be sanitized. For the sanitizing step, the cell counts in each group of variables are sanitized separately via the Laplace mechanism. The privacy budget is allocated proportionally to the number of variables in each group. e.g., *Call-Information* had 10 variables out of 32 possible, so a total of $\approx 0.31\epsilon$ privacy budget. Additionally, pfr "denoise" the counts based on modeling the number of empty and non-empty cells to reduce the excessive non-zero counts created during the sanitizing step. Any variables that are not grouped with others are generated by sampling from a uniform distribution before all counts are normalized to the desired synthetic data set size.

Overall, the typical data maintainer would have to hand-code the pfr approach given the lack of open source code (the team did not share their code on GitHub). Also, pfr depends heavily on the publicly available information for query selection to improve accuracy, so pfr would perform poorly on data sets with no associated public knowledge. Without the ability to generalize the algorithm, it is unlikely pfr would ever publish or get credit for these ideas, but Team pfr demonstrated how simpler DP methods that leverage on public or domain knowledge can perform as well as more complicated methods.

### 3.2 Parametric Data Synthesis

Parametric differentially private synthetic data methods rely on using or learning an appropriate distribution based on the original data and sampling values from a noisy version of that distribution. One of the concerns when applying a parametric approach is distribution or model selection itself might violate privacy. Either the data maintainer has to use a separate public data set to test what model is appropriate or leverage public knowledge on what model should be used to avoid a privacy violation. If this is not possible, the data maintainer may apply a differentially private model selection method (Lei et al., 2018). These methods are also generally much more computationally demanding than the non-parametric methods. Table 2 lists the methods used in the NIST Data Challenge.

*Table 2: Summary of the parametric differentially private synthetic data approaches discussed in Section 3.2.*

| **Team** | **Computation** | **Off-the-Shelf vs. Hand-Coding** | **Pre- and Post-Processing** |
|---|---|---|---|
| Team PrivBayes (Sec. 3.2.1) | more computationally complex compared to the other methods | off-the-shelf via `Python` code on GitHub | **pre-processing:** Bayesian network to determine which variables are highly correlated or not; **post-processing:** enforcing consistency among the marginals |
| Team RMcKenna (Sec. 3.2.2) | light to moderate computational complexity | some hand-coding due to identifying marginals for pre-processing, `Python` code on GitHub | **pre-processing:** identify marginals from public data |
| Team UCLANESL (Sec. 3.2.3) | the most computationally complex method; requires more RAM memory | off-the-shelf via `Python` code on GitHub | none |

#### 3.2.1 Team PrivBayes

Team PrivBayes has a well developed DP approach fully detailed in their PrivBayes paper (Zhang et al., 2017). They placed fifth in Match #1 and third in Matches #2 and #3. Simply put, PrivBayes uses a Bayesian network with binary nodes and low-order interactions among the nodes to release high-dimensional data that satisfies differential privacy. Specifically, PrivBayes creates

a Bayesian network by first scoring each pair of possible attributes that indicates if the attributes are highly correlated or not. These scores are sanitized via the Gaussian Mechanism and then used to create the Bayesian network. When the attributes are continuous values, PrivBayes must discretize the values to create the Bayesian network. Using the differentially private Bayesian network, PrivBayes approximates the distribution of the original data with a set of $P$ many low-dimensional marginals. These $P$ many marginals are then santitized via the Gaussian Mechanism, which are used with the Bayesian network to reconstruct an approximate distribution of the original data set. PrivBayes then generates the synthetic data by sampling tuples from this approximated distribution. Post-processing involves enforcing consistency on the noisy marginals in three parts: 1. marginal set of attributes, 2. attribute hierarchy, and 3. overall consistency.

PrivBayes performs fairly well while not requiring public data for a pre-processing step such as the non-parametric approaches described in Section 3.1. Additionally, there exists PrivBayes `Python` code on GitHub (dataresponsibly.com, 2018), allowing data maintainers to easily apply PrivBayes to their data. However, the complexity of PrivBayes due to constructing the differentially private Bayesian network and enforcing consistency among the noisy marginals increases the computational burden compared to the other methods. A data maintainer might be limited in implementing PrivBayes depending on computational resources and the size of the target data set. The complexity of the approach also means that it is harder to diagnose potential issues that may lead to inaccurate synthesis. While the empirical methods are easy to tune to public data, PrivBayes represents essentially a black-box method. That being said, the well founded theory and representation in the literature is a boon to its potential use.

### 3.2.2 Team RMcKenna

Team RMcKenna's performed third in Match #1, fourth in Match #2, and first in Match #3. Their approach is the most similar of the parameteric approaches to a non-parametric approach, since the algorithm focuses on determining a smaller set of cells to perturb and then sampling data from these noisy marginals. As a first step, Team RMcKenna's method uses a similar pre-processing step to the non-parametric methods by first identifying the 1-, 2-, 3-way marginals of the highly correlated variables on a public data set to avoid splitting the privacy-loss budget. Similarly to DPSyn, the marginal counts are then sanitized via the Gaussian Mechanism, but RMcKenna also utilizes the Moments Accountant, a privacy-loss tracking technique that tightens the privacy bound for the Gaussian Mechanism better than Theorem 1, resulting in less noise on the marginals (Abadi et al., 2016). Based on the sanitized marginals, Team RMcKenna uses graphical models to determine a model for the data distribution, capturing the relationships among the variables and generating the synthetic data (McKenna et al., 2019).

Team RMcKenna's method is fairly easy to understand, resembling the implementation steps for DPSyn and DPFieldGroups, while utilizing some more advance techniques for splitting the privacy budget across cells and sampling from the noisy marginals. The combination of the parametric and non-parametric ideas offers a unique approach among the competitors. The algorithm is also straightforward to implement. The data maintainer must first select the highly correlated variables for the low dimensional marginals before executing the `Python` code from McKenna (2019) on GitHub. This approach is fairly novel, but given its performance and the fact that it builds on previous work, it will likely gain acceptance in the literature.

### 3.2.3 Team UCLANESL

Team UCLANESL placed fourth in Match #1 and fifth in Match #3, and they did not compete in Match #2. They based their DP approach on the Wasserstein generative adversarial network (WGAN) training algorithm along with the Gaussian Mechanism and the Moment Accountant technique to ensure DP. First, WGAN trains two competing models: the *generator*, a model that learns to generate synthetic data from the target data, and the *discriminator*, a model that attempts to differentiate between observations from the training data and the *generator* created synthetic data. The *generator* creates fake observations that mimic ones from the target data by taking in a noisy vector sampled from a prior distribution such as a normal or uniform distribution. These fake observations attempt to confuse the *discriminator*, reducing the model's ability to distinguish the target and synthetic data sets. For the models to be differentially private, the *discriminator* gradient updates are pertubed using the Gaussian mechanism. Essentially, the *discriminator* gradient updates are first "clipped" to ensure a bounded $l_2$ sensitivity before adding noise from the Gaussian Mechanism. These sanitized gradient updates are then used on the *discriminator* model weights, which means the *generator* model also satisfies DP since it relies on the feedback from the *discriminator*. The Moment Account technique comes in to track the privacy-loss budget and will abort the WGAN training if the privacy budget has been reached.

For further details, Alzantot and Srivastava (2019) provides a full technical report with proofs in addition to their `Python` code. As a published paper with publicly available code, it is possible this method could become commonly implemented. However, Team UCLANESL's method is the most computationally intense out of all the competitors. One reason is their method consumes a lot of RAM memory. Team UCLANESL's `Python` code includes the TensorFlow library, a GPU-accelerated deep learning framework, that they report significantly reduces the computational time when the code runs on a GPU-powered machine. For this reason, we suspect the average data maintainer will have extreme difficulties implementing the DP WGAN method given the computational power required.

## 4 Metrics to Evaluate the Synthetic Data Quality

In this section, we describe the scoring methods used for the NIST Differentially Private Synthetic Data Challenge and detail our quality metrics we used for additional evaluation of the resulting data sets.

### 4.1 NIST Differentially Private Synthetic Data Challenge Scoring

*Table 3: NIST Differentially Private Synthetic Data Challenge Marathon Match Information.*

| Match | Training Data | Scoring Data | Analyses |
|---|---|---|---|
| 1 | 2017 SFFD's Call for Service Data | 2016 SFFD's Call for Service Data | Clustering |
| 2 | 2016 SFFD's Call for Service Data | 2006, 2017 SFFDs Call for Service Data | Clustering and Classification |
| 3 | Colorado PUMS | Arizona and Vermont PUMS | Clustering, Classification, and Regression |

We summarize the data and scoring analyses used for the three "Marathon Matches" in Table 3

from the NIST Differentially Private Synthetic Data Challenge. For each match, the final scores were progressively evaluated based on the clustering, classification, and regression analyses the NIST Data Challenge created. This means Match #1 had only the clustering analysis, Match #2 had the clustering and classification analyses, and Match # 3 used all three analyses. The clustering analysis compared the 3-way marginal density distributions between the original and synthetic data sets, where the utility score was the absolute difference in the density distributions. This calculation was repeated 100 times on randomly selected variables, and then averaged for the final clustering score. The classification analysis first randomly picked 33% of the variables. If a particular variable was categorical, a subset of the possible variable values were randomly picked, where as, if the variable was continuous, a range of values were randomly picked. In either case, these selected values were used to calculate how many of the observations in the synthetic and original data matched the specific variable subset. The synthetic data matched counts were then subtracted from the original data matched counts before taking the natural log. This natural log difference was computed over 300 repeats, where the final classification score was the root mean-squared on the repeats divided by $ln(10^{-3})$. The term classification is slightly misleading, given that this was essentially testing similarity between the original and synthetic data in randomly selected subsets of the joint distributions. Lastly, the regression analysis used a two part score system. The first score calculated the mean-square deviation of the Gini indices in the original and synthetic data sets for every city, and then those values were averaged over the total number of cities in the original data. The second score compared how the cities in the original and the synthetic data sets were ranked on gender pay gap. The rank differences were also calculated by the mean-square deviation. These two scores were averaged for the overall regression analysis score. Again, the term regression is slightly misleading given that this was not a comparison of regression coefficients, as is commonly seen in the literature.

### 4.2 Discriminant-based Quality Metric Algorithms

The NIST algorithms were devised specifically for the challenge. We now describe more standard approaches that have been used frequently in the literature. When assessing differentially private mechanisms, researchers typically report the $l_1$ distance between the original values and released queries or data. Alternatively, some papers may assess how well the differentially private synthetic data performed on a specific statistical analysis such as linear regression, also using the $l_1$ distance between original and noisy coefficients. For assessing the accuracy of an entire synthetic data set, these evaluations are not sufficient since the $l_1$ distance does not indicate how well complex relationships between variables are preserved. Additionally, it is impossible to predict every analysis a data practitioner may wish to implement using the data. Expecting the data maintainer to exhaustively investigate all possible statistical tests a data practitioner might want to conduct would be impractical and impossible.

To resolve these issues, one area of SDC and DP research is the development of a quality or utility metric to measure how "close" the synthetic data is to the original data. We focus here on a few approaches that utilize the concept of propensity scores as discriminates between the original and synthetic data, with utility metrics calculated in different ways using the estimated propensity scores. These methods were first developed on traditional synthetic data, but they apply as well to DP synthetic data. The basic idea behind this approach is to train a classifier to discriminate between two data sets. The worse it is able to perform, the more similar the data sets are on a distributional level.

Woo et al. (2009) first proposed using propensity scores (or predicted probabilities) and summarized

them into a utility metric by calculating the mean-square difference between the propensity score estimates and the true proportion the synthetic data within the total combined data set. This value was later coined the propensity score mean-squared error (pMSE). Sakshaug and Raghunathan (2010) applied a Chi-squared test on the discretized estimated propensity scores. Snoke et al. (2018) improved on the pMSE by deriving its theoretical expected value and standard deviation under certain null conditions and using these to create standardized versions of the statistic called the pMSE-ratio and standardized pMSE. Bowen and Liu (2018) developed SPECKS (**S**ynthetic data generation; **P**ropensity score matching; **E**mpirical **C**omparison via the **K**olmogorov-**S**mirnov distance), which applies the Kolmogorov-Smirnov (KS) distance to the predicted probabilities as the utility metric. A small KS distance would indicate that the original and synthetic empirical CDFs are indistinguishable.

In many ways, these approaches are more formalized versions of the scoring metrics devised for the NIST Data Challenge, which also sought to assess distributional similarity in various ways. Because of this, we expect to see general agreement between our metrics and the results from the NIST Data Challenge. For data maintainers, the formally developed methods that we present will be easier to implement than metrics tailored specifically to the NIST Data Challenge.

**Algorithm 1** General Utility Measure based on Propensity Score

1: Apply $V$ fold cross-validation on $X$ and $X^*$, where $X_j$ and $X_j^*$ are the $j$th cross validation fold for $j = 1, ..., V$.
2: **for** $j = 1$ to $V$ **do**
3: Divide $X_j$ and $X_j^*$ into training and test data with split of $s$.
4: Combine the training and test data sets of the original and synthetic data, respectively, such that $X_j^{train}$ has dimension $N_{train} \times q$ and $X_j^{test}$ has dimension $N_{test} \times q$, where $N_{train} = s(n + n^*)/V$ and $N_{test} = (1 - s)(n + n^*)/V$.
5: Add an indicator variable $T$ to $X_j^{train}$ and $X_j^{test}$ such that $T_i = 1$ if record $x_i \in X^*$ and $T_i = 0$ otherwise for all $i = 1, \ldots, N_{train}$ and $i = 1, \ldots, N_{test}$.
6: Train model from the desired classification method on $X_j^{train}$ (see Remark 1).
7: Calculate the propensity scores for each record $x_i \in X_j^{test}$, $\hat{p}_i = Pr(T_i = 1|x_i)$ by applying the trained model in the previous step for all $i = 1, \ldots, N_{test}$.
8: Combine the propensity scores across all $V$ folds.
9: Apply the desired utility statistic.

**Remark 1.** Various classification methods can be used such as a non-parametric classification and regression trees (CART), logistic regression, and support vector machine (SVM) for calculating the propensity scores in Step 7 of Algorithm 1.

For our analysis in Section 5, we implement the pMSE-ratio and SPECKS. Algorithm 1 outlines the common steps for calculating these metrics until Step 9. Both approaches require training and fitting classifiers to the combined original and synthetic data, with a binary indicator labeling the data set each row. We obtain the predicted probabilities of this binary label, and for the pMSE-ratio, we compute the pMSE using

$$pMSE = \frac{1}{N} \sum_{i=1}^{N} (\hat{p}_i - c)^2 \tag{7}$$

where $N = n + n^*$ is the total number of observations from both the synthetic and original data; and $c = n^*/N$ is the proportion of observations from the synthetic data out of the total. This values is typically 0.5 because synthetic data is often generated with the same number of rows as the original data. If the classifier we use is a parametric model, we can use the theorems derived in Snoke et al. (2018) for the expected value under the null hypothesis that the original and synthetic data were sampled from the same generative distribution. We calculate the expected null mean using

$$\mathbb{E}(pMSE) = (k-1)\frac{(1-c)^2 c}{N} \tag{8}$$

such that $k$ is the number of parameters from the selected model. In the case of using a non-parametric classifier, we can approximate the null expected value using resampling techniques such as permutating the rows and reestimating the pMSE. After calculating the null mean pMSE, we obtain the pMSE-ratio by dividing the observed mean pMSE calculated on the original and synthetic data by the null mean. A pMSE-ratio value close to 1 indicates that the synthetic data is very similar to the original data. For a more in-depth discussion of this method, its strengths and weaknesses, please see Snoke et al. (2018).

For SPECKS, we determine the empirical CDFs of the propensity scores for the original and synthetic data sets separately, and then apply the Kolmogorov-Smirnov (KS) distance on the two empirical CDFs (Bowen and Liu, 2018). The KS distance is the maximum distance of two empirical CDFs, where the synthetic and the original data have the largest separation. A smaller KS distance (close to 0) indicates that the synthetic data preserved the original data well whereas a larger KS distance (close to 1) means the synthetic data differs a lot from the original data. Again, for more on this method please see Bowen and Liu (2018).

Both of these methods can be used with either simple parametric models, such as logistic regression, or more complex non-parametric models, such as classification and regression trees (CART). For any given model, we can compare various synthetic data sets using these metrics. One issue is that we may obtain different rankings if we use different classifiers. This is due to the fact that models with varying complexity are actually measuring different types of accuracy. A logistic regression with only main effects, for example, is only measuring the accuracy of the first-order marginal distributions. A more complex CART model on the other hand is likely measuring high-order interactions in the data. As was recommended in previous work, we use different classifiers and compare relative rankings from each set of models. This gives a holistic view of the utility of the data.

Future work could explore and compare various classification methods (Step 6 in Algorithm 1) as well as other loss functions for the utility statistic (Step 9 in Algorithm 1). We discuss this topic further in Section 6.

## 5 Results

In this section, we report the results of the NIST Differentially Private Synthetic Data Challenge, and we compare these scores to the results we get from calculating the pMSE-ratio and SPECKS metrics on the same synthetic data generated from the competition. We were given access by NIST to the original synthetic data generated by the competitors listed in Table 4. We make this comparison for two primary reasons. First, to determine whether alternative metrics would have changed the ranking of the challenge, and second we wish to provide a case study for data maintainers on potential utility evaluations of differentially private synthetic data. The metrics we present are more broadly applicable than those used in the challenge and should be easily adopted

by an interested data maintainer.

### 5.1 NIST Differentially Private Synthetic Data Challenge

For Matches #1 and #2, the privacy-loss budget was set at $\epsilon = \{0.01, 0.1, 1\}$ and $\delta = 0.001$ (if contestants chose to use it), whereas for Match #3, $\epsilon$ was set at higher levels of $\{0.3, 1, 8\}$ and $\delta$ was kept the same. The stated reason for the increase in $\epsilon$ values was due to the increased complexity of the PUMS data used for Match #3. Additionally, contestants were asked to generate multiple differentially private data sets for each match. Matches #1 and #2 had three synthetic data sets while Match #3 had five synthetic data sets. All contestants divided $\epsilon$ equally across each data set, which, per Theorem 1, resulted in the $\epsilon$ values used for each single data set being the total divided by the number of multiple synthetic data sets. Accordingly, the $\epsilon$ used *per* data set in Matches #1 annd #2 was $\{0.00\overline{3}, 0.0\overline{3}, 0.\overline{3}\}$ and for Match #3 the per data set $\epsilon$ was $\{0.06, 0.2, 1.6\}$.

*Table 4: NIST Differentially Private Synthetic Data Challenge Results from the three Marathon Matches.*

| Rank | Match #1 | Match #2 | Match #3 |
|---|---|---|---|
| 1 | Team pfr (Sec. 3.1.3) | Team pfr (Sec. 3.1.3) | Team RMcKenna (Sec. 3.2.2) |
| 2 | Team DPSyn (Sec. 3.1.1) | Team DPSyn (Sec. 3.1.1) | Team DPSyn (Sec. 3.1.1) |
| 3 | Team RMcKenna (Sec. 3.2.2) | Team PrivBayes (Sec. 3.2.1) | Team PrivBayes (Sec. 3.2.1) |
| 4 | Team UCLANESL (Sec. 3.2.3) | Team RMcKenna (Sec. 3.2.2) | Team Gardn999 (Sec. 3.1.2) |
| 5 | Team PrivBayes (Sec. 3.2.1) | Team Gardn999 (Sec. 3.1.2) | Team UCLANESL (Sec. 3.2.3) |

Table 4 lists the team ranks while Figure 1 shows the numerical results of the NIST Differentially Private Synthetic Data Challenge matches. There are a total of six teams that ranked in the Top 5 throughout the matches. Note that Teams Gardn999, UCLANESL, and pfr did not compete in Matches #1, #2, and #3, respectively, so their absence from the Top 5 for each match was not due to scoring lower than rank 5. Overall, we recommend methods proposed by DPSyn and RMcKenna based on their NIST Data Challenge ranks, computational burden and complexity, and the ease of implementation for the average data maintainer (see our reviews in Sections 3.1 and 3.2). Team DPSyn placed second throughout the entire competition with their method while RMcKenna placed third, fourth, and then first for each the matches, respectively. Their methods require some pre-processing by identifying the marginals from public data, but the methods are easy to execute with open source code and are not as computationally demanding.

Figure 1 displays the numeric scores of the NIST Data Challenge for Matches #2 and #3. We focus on Matches #2 and #3 since Match #1 used the same data type as Match #2. Examining the results, Match #2 shows fairly "flat" scores whereas Match #3 scores slightly increased with larger $\epsilon$ (except for Team UCLANESL on Vermont). The lack of any trend in Match #2 for increasing $\epsilon$ is mostly likely due to the small $\epsilon$ values used for scoring. The values are very small after dividing them over multiple synthetic data sets, so the differentially private synthetic data sets have lower utility for the higher privacy guarantee. For future differentially private data challenges, a wider range of $\epsilon$ should be applied to both verify empirically if the methods satisfy DP and demonstrate

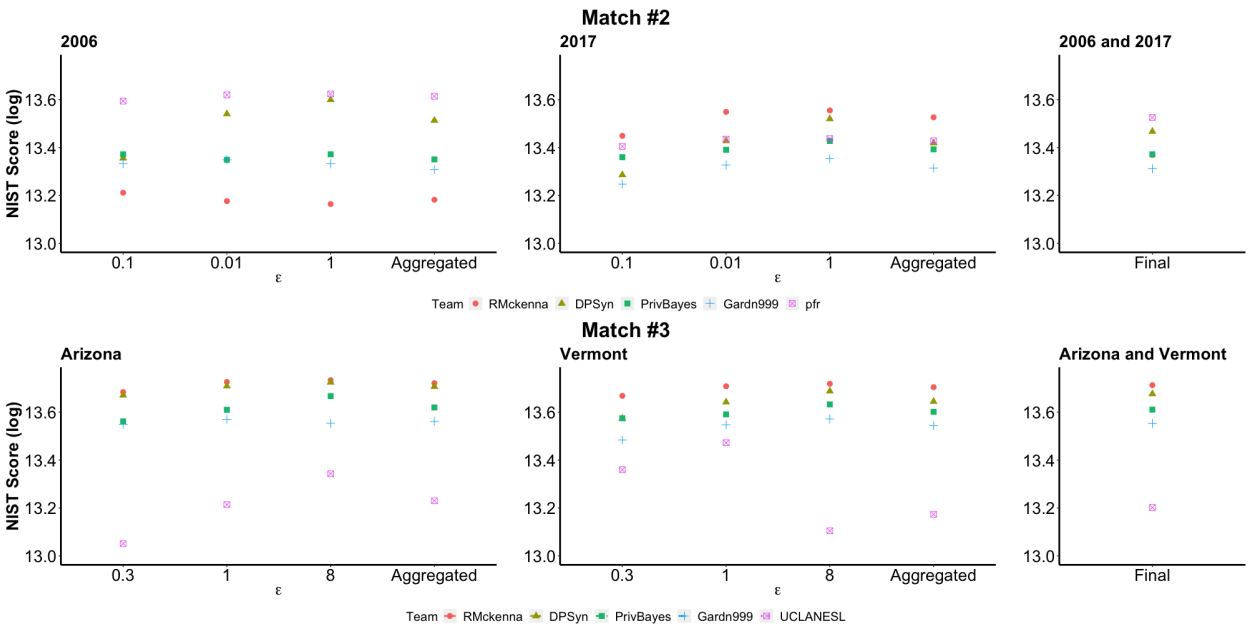


*Figure 1: The average NIST Differential Privacy Synthetic Data Challenge score results on a log scale for Matches #2 (2006 and 2017) and #3 (Arizona and Vermont).*

a trend towards higher accuracy as $\epsilon$ grows. Algorithms which do not eventually asymptote at maximal accuracy as privacy-loss goes to infinity are inherently flawed and should not be utilized. Testing methods at a wide enough range of privacy-loss budget values is needed for the potential data practitioner to understand which algorithms are most suitable.

## 5.2 Metric Evaluation

When applying the metric algorithms from Section 4, we implemented both CART and logistic regression with all main effects of the variables for estimating predicted probabilities (see Remark 1). We used the `R` package `rpart` for CART with a complexity parameter (CP) chosen by cross-validation. We applied a 5-fold cross-validation with a 70-30 train-test split on both classifiers and produced the estimated predicted probabilities (propensity scores) from the test predictions. Since there were multiple synthetic data sets, we calculated the pMSE-ratio and KS distance for each data set (using the same best performing CP values across all data sets) and then averaged the results. For pMSE-ratio with the CART models, we generated 10 permutations to estimate the null mean pMSE.

Figures 2 and 3 show the results of pMSE-ratio (log scale) and SPECKS trained on the CART and logistic regression models, respectively, for Match #2. We applied a log scale on the pMSE-ratio results so that the ideal value for pMSE-ratio was the same as SPECKS, which is 0. The figures show the utility metrics, and they also show the estimated cross-validation error of the best performing model and its respective CP value. These plots are roughly the inverse of the utility plots, and they help to understand how the utility metric is calculated. A higher error in the classifier implies the data sets are harder to discriminate, and this corresponds to a smaller (or better) utility score.

Similar to the NIST scores from Figure 1, the utility metrics estimated from both classifiers for Match #2 provide relatively "flat" values due to the very small and narrow range of $\epsilon$ values. When

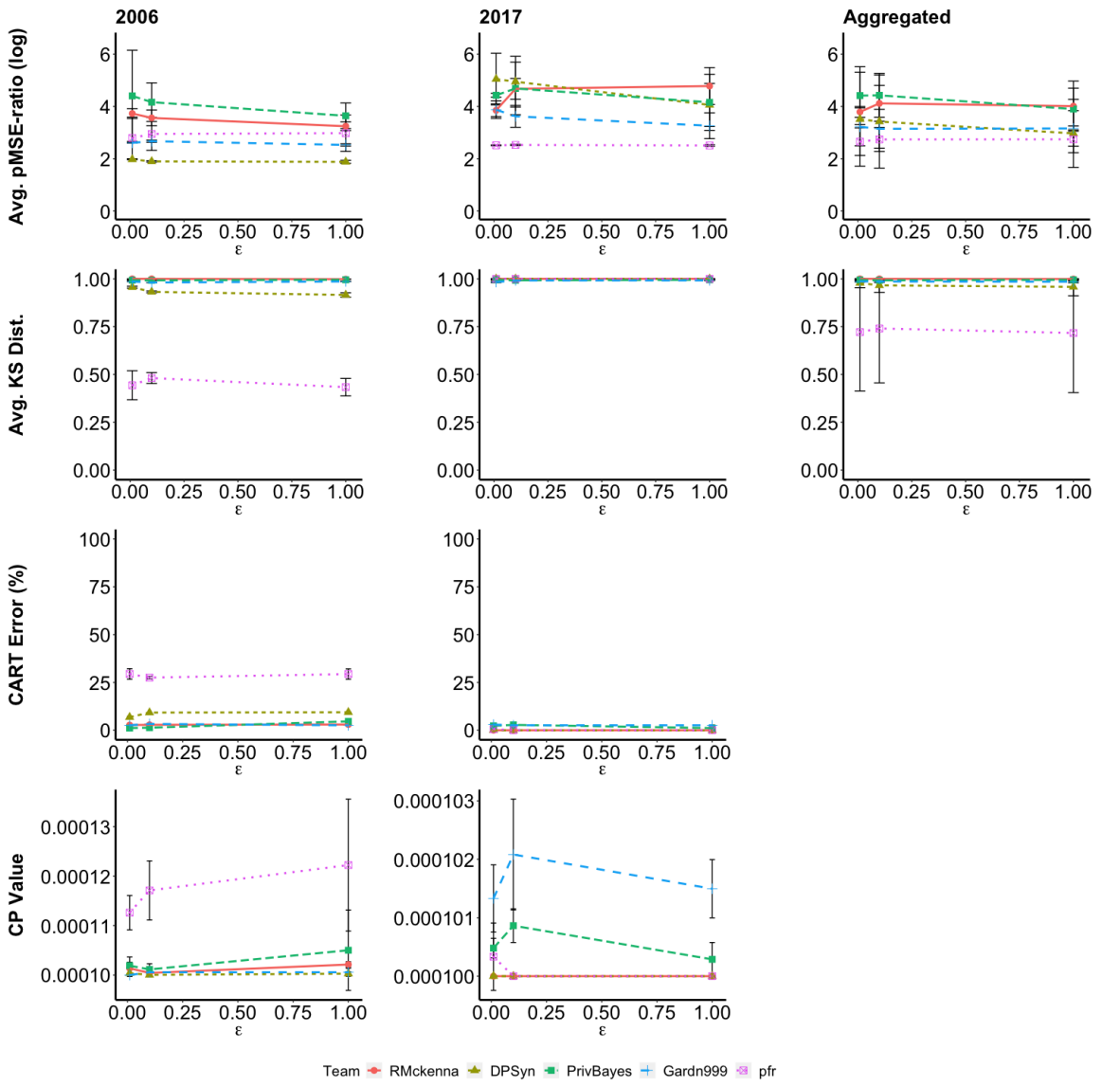


*Figure 2: The average pMSE-ratio, KS distances, error rate, and optimal complex parameter (CP) values from the trained CART model on the Match #2 data.*

using the CART model, both quality metrics correctly predicted that Team pfr's synthetic data was the closest to the original data. SPECKS barely ranked Team DPSyn better than the remaining methods that are considered too noisy to differentiate one another with KS distance values close to 1. Unlike SPECKS, pMSE-ratio clearly separated the remaining teams and identified Team DPSyn as one of the better performing methods as well. With the logistic regression model, both pMSE-ratio and SPECKS inaccurately predicted that Team PrivBayes generated the most similar synthetic data to the original data. Our utility metrics only correctly ranked Team RMcKenna as fourth place compared to the NIST Data Challenge final outcome. This larger discrepancy in predicting the NIST Data Challenge results is likely due the logistic regression measuring the accuracy of the marginal distributions rather than what was scored in Match #2, which is more similar to what the CART model may be measuring. Table 5 summarizes the utility metric results compared to the NIST Data Challenge outcome.

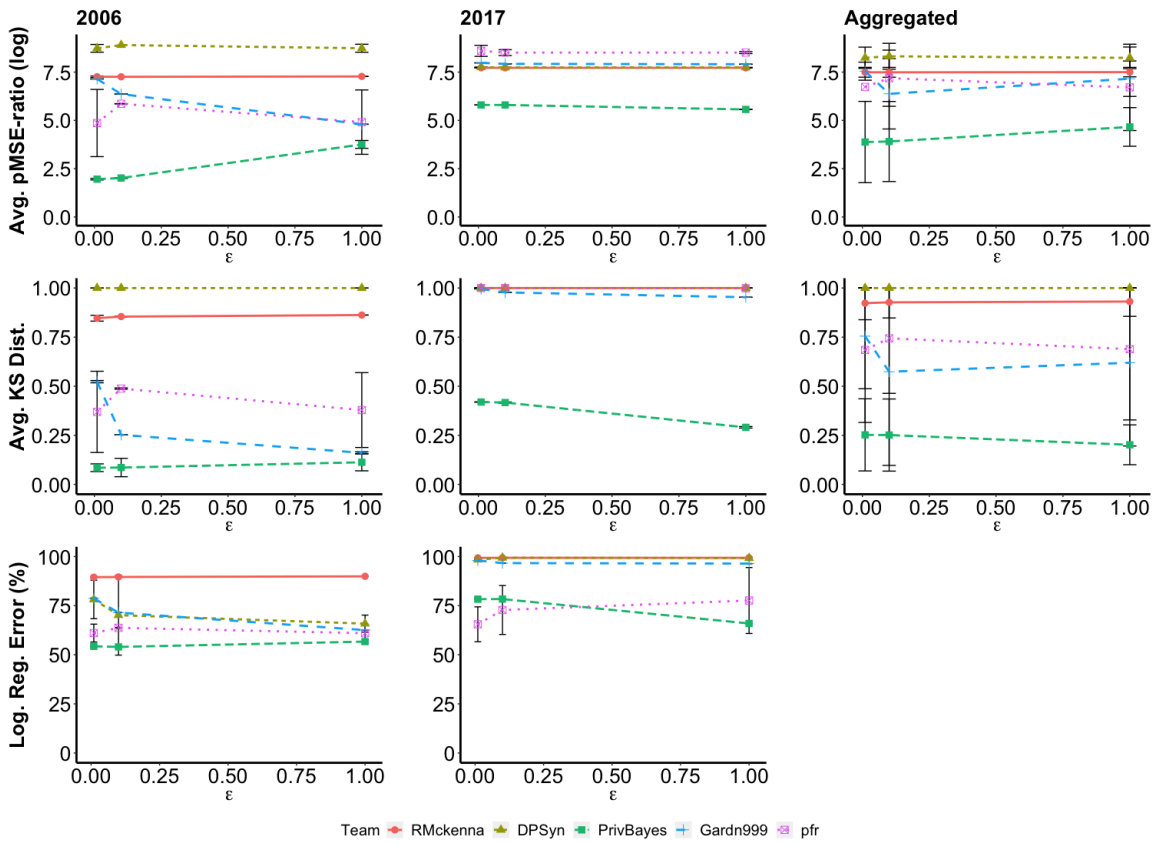


*Figure 3: The average pMSE-ratio and KS distances trained on the Match #2 data with the logistic regression model.*

*Table 5: Summary of the pMSE-ratio and SPECKS ranking results on Match #2. The team names that are in bold indicate that the utility metric matches the team placement in the NIST Differentially Private Synthetic Data Challenge.*

| | **CART** | | **logistic regression** | |
|---|---|---|---|---|
| **Rank** | **pMSE-ratio** | **SPECKS** | **pMSE-ratio** | **SPECKS** |
| 1 | **Team pfr** | **Team pfr** | Team PrivBayes | Team PrivBayes |
| 2 | Team Gardn999 | **Team DPSyn** | Team Gardn999 | Team Gardn999 |
| 3 | Team DPSyn | Team Gardn999 | Team pfr | Team pfr |
| 4 | Team RMcKenna | Team PrivBayes | **Team RMcKenna** | **Team RMcKenna** |
| 5 | Team PrivBayes | Team RMcKenna | Team DPSyn | Team DPSyn |

Figures 4 and 5 display the results of pMSE-ratio (log scale) and SPECKS trained on the CART and logistic regression models, respectively, for Match #3. With a wider range of $\epsilon$ values than Match #2, the results from the quality metrics generally decrease as $\epsilon$ increases. This trend provides some empirical evidence that the pMSE-ratio and SPECKS approaches are good metrics for measuring the similarity between differentially private synthetic data and original data. Under the CART model, pMSE-ratio predicted none of the teams correctly. SPECKS only identified the

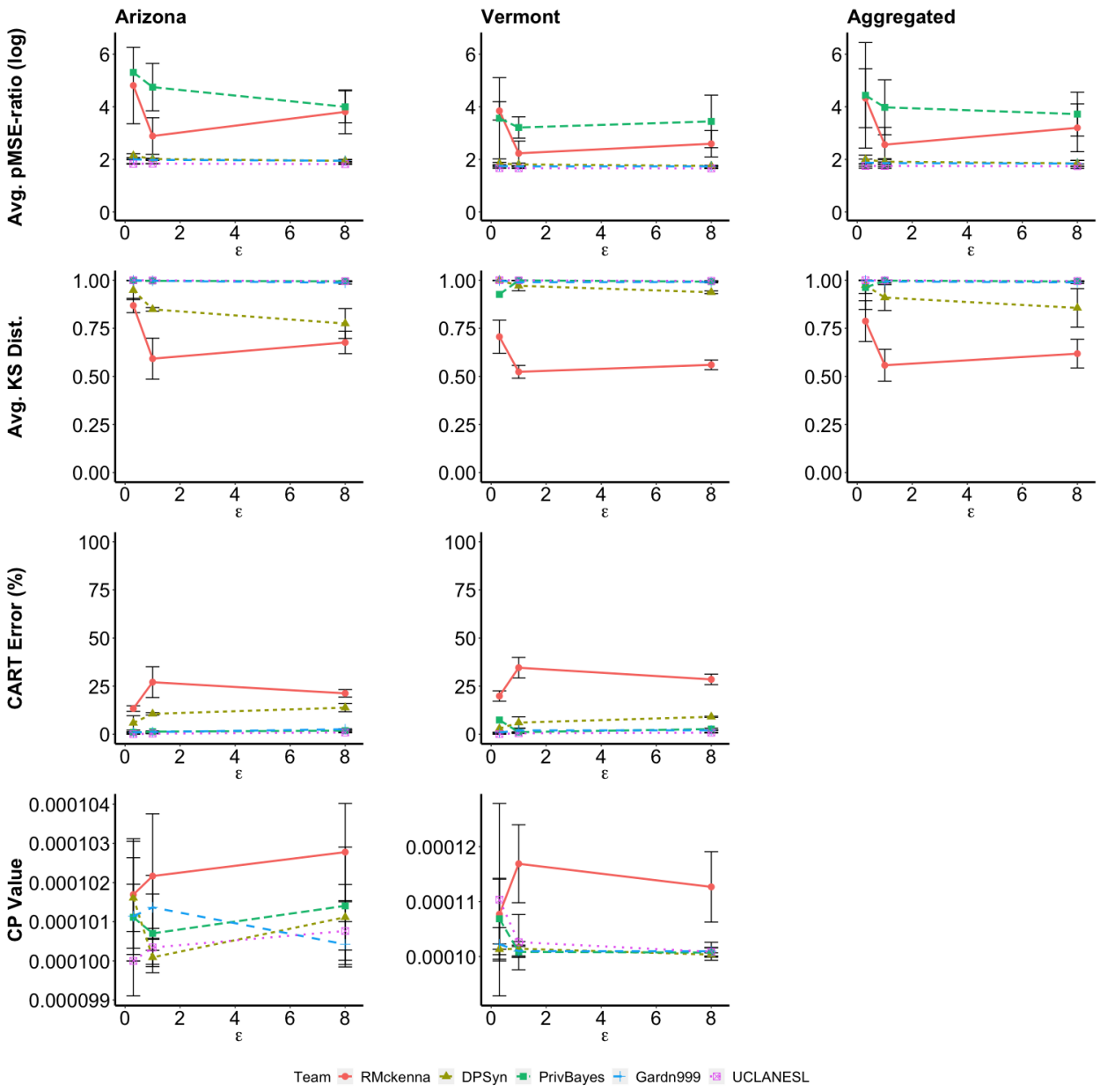


*Figure 4: The average pMSE-ratio, KS distances, error rate, and optimal complex parameter (CP) values from the trained CART model on the Match #3 data.*

top two teams (RMcKenna and DPSyn) accurately, but, again, could not differentiate the remaining teams since the values were too close to 1. Using the logistic regression model, both utility metrics correctly determined that Team RMcKenna's synthetic data are closer to the original data and that Team UCLANESL performed the worst. pMSE-ratio also placed Team Gardn999 correctly while SPECKS incorrectly ranked the remaining teams. Unlike Match #2, the logistic regression model more accurately predicted the NIST Data Challenge outcome than the CART model, probably due to the additional regression analysis for the NIST Data Challenge scoring. Table 6 lists how the utility metric rank the teams compared to the NIST Data Challenge outcome.

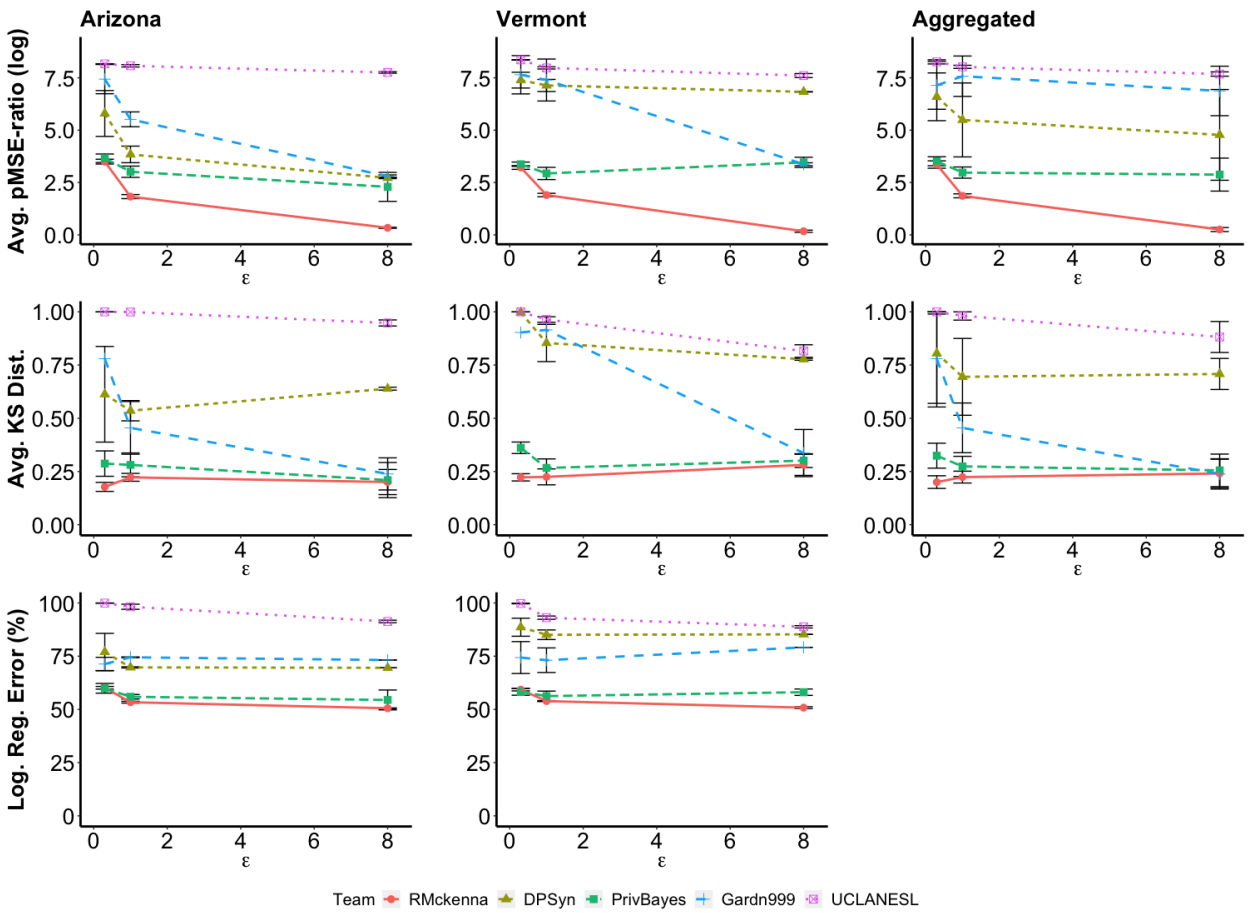


*Figure 5: The average pMSE-ratio and KS distances trained on the Match #3 data with the logistic regression model.*

*Table 6: Summary of the pMSE-ratio and SPECKS ranking results on Match #3. The team names that are in bold indicate that the utility metric matches the team placement in the NIST Differentially Private Synthetic Data Challenge.*

| | **CART** | | **logistic regression** | |
|---|---|---|---|---|
| **Rank** | **pMSE-ratio** | **SPECKS** | **pMSE-ratio** | **SPECKS** |
| 1 | Team UCLANESL | **Team RMcKenna** | **Team RMcKenna** | **Team RMcKenna** |
| 2 | Team Gardn999 | **Team DPSyn** | Team PrivBayes | Team PrivBayes |
| 3 | Team DPSyn | **Team PrivBayes** | Team DPSyn | Team Gardn999 |
| 4 | Team RMcKenna | Team UCLANESL | Team Gardn999 | Team DPSyn |
| 5 | Team PrivBayes | Team Gardn999 | **Team UCLANESL** | **Team UCLANESL** |

Overall, the quality metrics approaches predicted the NIST Data Challenge outcome from poorly to moderately well, depending on the choice of classifier. This difference was likely due primarily to two reasons. First, that our metrics were measuring different qualities of the data than those

used in the NIST challenge, and second that many of the participants' data sets were very close in terms of accuracy. On the latter point, slight changes or even randomness could lead to a difference in ranking. Both of these reasons suggest that it is often hard to determine the "best" synthetic data set, and in a competition much depends on the choice of metrics.

Data maintainers wishing to select algorithms for releasing data should use a suite of metrics, as they are best informed by an ensemble of utility metrics. For the discriminator approaches, one should select a classifier based on what measure of accuracy is desired from the synthetic data compared to the original data. For instance, in the context of the NIST Data Challenge, the CART model might be the most informative for Match #2 and the logistic regression model for Match #3.

## 6 Conclusions and Future Work

In this paper, we reviewed and evaluated the top NIST Differentially Private Synthetic Data Challenge methods along with implementing two of our own utility metric algorithms. This is the first comparative work, to the best of our knowledge, that assess a variety of DP synthetic data generation mechanisms applied to complex real-world data sets and gives recommendations concerning their accuracy and ease of implementation. We highlight the methods of Team RMcKenna and Team DPSyn as the overall best in terms of performance, ease of implementation, and current public availability in the literature.

For future DP data competitions, we recommend a wider range of the privacy-loss budget to be explored. As seen in Match #2, there was a lack of an asymptotic trend as $\epsilon$ increased, which made it difficult to learn much from this match. Additionally, the NIST data sets used in the competition are very complex compared to what is typically seen in literature, and these results suggest such complx data require more privacy-loss budget for greater accuracy. Elements such as structural missing values in the Match #2 data (e.g., some emergency calls do not have an officer dispatched to the location) or the large number of variables in Match #3 greatly increase the difficulty of providing accurate differentially private synthetic data.

Our utility metric algorithms' showed mixed accuracy in predicting the NIST Data Challenge outcome depended on the choice of classifier and the NIST evaluation standards used in different matches. This points to the difference in what many utility metrics measure, and it suggests that a suite of metrics is the most informative approach for data maintainers wishing to evaluate DP mechanisms. Future work on discriminator approaches to utility should investigate additional loss functions and classification methods (e.g., SVM), conducting a thorough investigation of what data qualities and features are being measured. However, there is a concern for using certain classifiers such as SVM that are computationally more expensive. Since most DP methods are computationally intense, the utility metrics should ideally be calculated using very little the computational resources, which might eliminate SVM and other classifiers in practice. Also, besides the quality metrics we described in Section 4, there are other ways to extensively evaluate differentially private method such as DPBench (Hay et al., 2016). This paper is a first step in developing the field of knowledge on evaluation metrics to compare DP synthetic data, and further applications will continue to be highly beneficial for data maintainers in deciding a DP synthetic data approach for future data products.

## Acknowledgments

This research is supported by the National Institute for Standards and Technology Public Safety Communications Research Division.

## References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM.

Abowd, J. M. and Vilhuber, L. (2008). How protective are synthetic data? In *International Conference on Privacy in Statistical Databases*, pages 239–246. Springer.

Alzantot, M. and Srivastava, M. (2019). Differential Privacy Synthetic Data Generation using WGANs.

Bowen, C. M. and Liu, F. (2016). Comparative study of differentially private data synthesis methods. *arXiv preprint arXiv:1602.01063*.

Bowen, C. M. and Liu, F. (2018). Statistical election to partition sequentially (steps) and its application in differentially private release and analysis of youth voter registration data. *arXiv preprint arXiv:1803.06763*.

Charest, A.-S. (2011). How can we analyze differentially-private synthetic datasets? *Journal of Privacy and Confidentiality*, 2(2).

Chaudhuri, K., Sarwate, A., and Sinha, K. (2012). Near-optimal differentially private principal components. In *Advances in Neural Information Processing Systems*, pages 989–997.

dataresponsibly.com (2018). Datasynthesizer. https://github.com/DataResponsibly/DataSynthesizer.

De Montjoye, Y.-A., Hidalgo, C. A., Verleysen, M., and Blondel, V. D. (2013). Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3:1376.

De Montjoye, Y.-A., Radaelli, L., Singh, V. K., et al. (2015). Unique in the shopping mall: On the reidentifiability of credit card metadata. *Science*, 347(6221):536–539.

Dowell, J., Cleland, J., Fitzpatrick, S., McManus, C., Nicholson, S., Oppé, T., Petty-Saphon, K., King, O. S., Smith, D., Thornton, S., et al. (2018). The UK medical education database (ukmed) what is it? why and how might you use it? *BMC Medical Education*, 18(1):6.

Drechsler, J. (2011). *Synthetic datasets for statistical disclosure control: theory and implementation*, volume 201. Springer Science & Business Media.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. (2006a). Our data, ourselves: Privacy via distriuted noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006b). Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, pages 265–284. Springer.

Dwork, C., Naor, M., Pitassi, T., Rothblum, G. N., and Yekhanin, S. (2010). Pan-private streaming algorithms. In *ICS*, pages 66–80.

Dwork, C. and Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407.

Dwork, C. and Rothblum, G. N. (2016). Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*.

Fioretto, F., Mak, T. W., and Van Hentenryck, P. (2019). Differential privacy for power grid obfuscation. *arXiv preprint arXiv:1901.06949*.

Friedman, A., Berkovsky, S., and Kaafar, M. A. (2016). A differential privacy framework for matrix factorization recommender systems. *User Modeling and User-Adapted Interaction*, 26(5):425–458.

Gardner, J. (2019). Differential Privacy Synthetic Data Challenge Algorithm. https://github.com/gardn999/PrivacyEngCollabSpace/tree/master/tools/de-identification/Differential-Privacy-Synthetic-Data-Challenge-Algorithms/DPFieldGroups.

González, F., Yu, Y., Figueroa, A., López, C., and Aragon, C. (2019). Global reactions to the cambridge analytica scandal: An inter-language social media study. *WWW '19 Companion Proceedings of The 2019 World Wide Web Conference*, pages 799–806.

Hardesty, L. (2013). How hard is it to'de-anonymize'cellphone data. *MIT News*, 27:2013.

Hay, M., Machanavajjhala, A., Miklau, G., Chen, Y., and Zhang, D. (2016). Principled evaluation of differentially private algorithms using dpbench. In *Proceedings of the 2016 International Conference on Management of Data*, pages 139–154. ACM.

Hay, M., Rastogi, V., Miklau, G., and Suciu, D. (2010). Boosting the accuracy of differentially-private queries through consistency. In *36th International Conference on Very Large Databases (VLDB)*. Citeseer.

Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Nordholt, E. S., Spicer, K., and De Wolf, P.-P. (2012). Statistical disclosure control. *John Wiley & Sons*.

Karwa, V., Krivitsky, P. N., and Slavković, A. B. (2017). Sharing social network data: differentially private estimation of exponential family random-graph models. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 66(3):481–500.

Karwa, V., Slavković, A., et al. (2016). Inference using noisy degrees: Differentially private $\beta$-model and synthetic graphs. *The Annals of Statistics*, 44(1):87–112.

Kondor, D., Hashemian, B., de Montjoye, Y.-A., and Ratti, C. (2018). Towards matching user mobility traces in large-scale datasets. *IEEE Transactions on Big Data*.

Lei, J., Charest, A.-S., Slavkovic, A., Smith, A., and Fienberg, S. (2018). Differentially private model selection with penalized and constrained likelihood. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 181(3):609–633.

Li, N., Zhang, Z., and Wang, T. (2019). DPSyn. https://github.com/usnistgov/PrivacyEngCollabSpace/tree/master/tools/de-identification/Differential-Privacy-Synthetic-Data-Challenge-Algorithms/DPSyn.

Little, R. J. (1993). Statistical analysis of masked data. *Journal of Official Statistics*, 9(2):407.

Liu, A., Xia, L., Duchowski, A., Bailey, R., Holmqvist, K., and Jain, E. (2019). Differential privacy for eye-tracking data. *arXiv preprint arXiv:1904.06809*.

Machanavajjhala, A., Kifer, D., Abowd, J., Gehrke, J., and Vilhuber, L. (2008). Privacy: Theory meets practice on the map. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, pages 277–286. IEEE Computer Society.

Manrique-Vallier, D. and Reiter, J. P. (2012). Estimating identification disclosure risk using mixed membership models. *Journal of the American Statistical Association*, 107(500):1385–1394.

Martin, K. D., Borah, A., and Palmatier, R. W. (2017). Data privacy: Effects on customer and firm performance. *Journal of Marketing*, 81(1):36–58.

McClure, D. and Reiter, J. P. (2012). Differential privacy and statistical disclosure risk measures: An investigation with binary synthetic data. *Trans. Data Privacy*, 5(3):535–552.

McKenna, R. (2019). rmckenna - Differential Privacy Synthetic Data Challenge Algorithm. https://github.com/usnistgov/PrivacyEngCollabSpace/tree/master/tools/de-identification/Differential-Privacy-Synthetic-Data-Challenge-Algorithms/rmckenna.

McKenna, R., Sheldon, D., and Miklau, G. (2019). Graphical-model based estimation and inference for differential privacy. *arXiv preprint arXiv:1901.09136*.

McSherry, F. and Talwar, K. (2007). Mechanism design via differential privacy. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 94–103. IEEE.

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

Mohammed, N., Chen, R., Fung, B., and Yu, P. S. (2011). Differentially private data release for data mining. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 493–501. ACM.

Narayanan, A. and Shmatikov, V. (2008). Robust de-anonymization of large datasets (how to break anonymity of the netflix prize dataset). *University of Texas at Austin*.

Nissim, K., Raskhodnikova, S., and Smith, A. (2007). Smooth sensitivity and sampling in private data analysis. In *Proceedings of the Thirty-ninth annual ACM Symposium on Theory of Computing*, pages 75–84. ACM.

Nissim, K., Steinke, T., Wood, A., Altman, M., Bembenek, A., Bun, M., Gaboardi, M., OBrien, D. R., and Vadhan, S. (2017). Differential privacy: A primer for a non-technical audience. In *Privacy Law Scholars Conf*.

Qardaji, W., Yang, W., and Li, N. (2014). Priview: practical differentially private release of marginal contingency tables. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1435–1446. ACM.

Quick, H. (2019). Generating poisson-distributed differentially private synthetic data. *arXiv preprint arXiv:1906.00455*.

Raab, G. M., Nowok, B., and Dibben, C. (2016). Practical data synthesis for large samples. *Journal of Privacy and Confidentiality*, 7(3):67–97.

Raghunathan, T. E., Reiter, J. P., and Rubin, D. B. (2003). Multiple imputation for statistical disclosure limitation. *Journal of Official Statistics*, 19(1):1.

Reiter, J. P. (2005). Using cart to generate partially synthetic public use microdata. *Journal of Official Statistics*, 21(3):441.

Reuter, M. S., Walker, S., Thiruvahindrapuram, B., Whitney, J., Cohn, I., Sondheimer, N., Yuen, R. K., Trost, B., Paton, T. A., Pereira, S. L., et al. (2018). The personal genome project canada: findings from whole genome sequences of the inaugural 56 participants. *CMAJ*, 190(5):E126–E136.

Rocher, L., Hendrickx, J. M., and De Montjoye, Y.-A. (2019). Estimating the success of re-identifications in incomplete datasets using generative models. *Nature Communications*, 10(1):3069.

Rubin, D. B. (1993). Statistical disclosure limitation. *Journal of Official Statistics*, 9(2):461–468.

Sakshaug, J. W. and Raghunathan, T. E. (2010). Synthetic data for small area estimation. In *International Conference on Privacy in Statistical Databases*, pages 162–173. Springer.

Snoke, J. and Bowen, C. M. (2019). Differential privacy: What is it? *AMSTAT news: the membership magazine of the American Statistical Association*, pages 26–28.

Snoke, J., Raab, G. M., Nowok, B., Dibben, C., and Slavkovic, A. (2018). General and specific utility measures for synthetic data. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 181(3):663–688.

Snoke, J. and Slavković, A. (2018). pmse mechanism: Differentially private synthetic data with maximal distributional similarity. In *International Conference on Privacy in Statistical Databases*, pages 138–159. Springer.

Sweeney, L. (2013). Matching known patients to health records in washington state data. *Available at SSRN 2289850*.

Tsay-Vogel, M., Shanahan, J., and Signorielli, N. (2018). Social media cultivating perceptions of privacy: A 5-year analysis of privacy attitudes and self-disclosure behaviors among facebook users. *New Media & Society*, 20(1):141–161.

Vendetti, B. (2018). 2018 differential privacy synthetic data challenge. https://www.nist.gov/communications-technology-laboratory/pscr/funding-opportunities/open-innovation-prize-challenges-1.

Wang, Y.-X., Fienberg, S., and Smola, A. (2015). Privacy for free: Posterior sampling and stochastic gradient monte carlo. In *International Conference on Machine Learning*, pages 2493–2502.

Wasserman, L. and Zhou, S. (2010). A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389.

Watts, R. (2019). Facial recognition as a force for good. *Biometric Technology Today*, 2019(3):5–8.

Woo, M.-J., Reiter, J. P., Oganian, A., and Karr, A. F. (2009). Global measures of data utility for microdata masked for disclosure limitation. *Journal of Privacy and Confidentiality*, 1(1).

Yu, F., Fienberg, S. E., Slavković, A. B., and Uhler, C. (2014). Scalable privacy-preserving data sharing methodology for genome-wide association studies. *Journal of Biomedical Informatics*, 50:133–141.

Zhang, J., Cormode, G., Procopiuc, C. M., Srivastava, D., and Xiao, X. (2017). Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4):25.