An Exploration of the Methods and Challenges of Differential Privacy

Harshit Agarwal (hagarwa3) and Zach Kimberg (kimberg2)

Abstract—In order to gain the maximum benefits from the large amounts of data generated by modern software, it is important that the data be shared with researchers and those best able to gain and use the insights it can generate. But, this runs into the problems of privacy. Even with names removed, it is quite possible, and increasingly likely, that the data can be matched with other public data sets and de-anonymized. When the data set contains sensitive data, such as medical, this can be problematic. Differential privacy is the most common technique used now to prevent this de-anonymization and help people feel secure in sharing their information. It works by adding random noise to the data to obscure individual data points, but the effect averages out over large amounts of data. For the survey, we will research the creation of differentially private data sets, mining with these data sets, and the possibility of using machine learning to undermine the privacy guarantees. We will look into the state of the art of differential privacy, and how novel techniques have been used for the same, including the role that neural networks can now play. In the process, we would also like to see what sort of applications certain methods are best suited for, and if there are any commonly exploitable weaknesses in the many methods used. We hope to understand the state of the art in creating data sets that are secure even against adversarial attacks, and gaining insights from them quickly and efficiently.

Index Terms—Differential privacy, data mining, privacy, anonymity

I. INTRODUCTION

The problem of privacy-preserving data analysis has a long history spanning multiple disciplines. There is a tremendous growth in digital information about individuals, data is becoming increasingly detailed, and technology now enables ever more powerful collection and curation of these data. With this, the need increases for a robust, meaningful, and mathematically rigorous definition of privacy, together with a computationally rich class of algorithms that satisfy this definition. Differential Privacy is such a definition.

The amount and variety of collected data, paired with the incredible adoption of cloud computing, presents previously impossible opportunities to create new applications and enables research that can change our lives for the better. However, like all coins, this has a flip side too. There is now a very real threat to privacy because data in raw form often contain sensitive information about individuals. Privacy-preserving data publishing (PPDP) studies how to transform raw data into a version that is immunized against privacy attacks but that still supports effective data mining tasks. In this survey, we discuss various strategies for PPDP, as well as common methods of privacy preserving data mining (PPDM), tying both of these with the ideas of differential privacy that are now a critical part of good storage of public data.

Defining privacy is a difficult task. One of the key challenges is how to model the background knowledge of an adversary. It would be reasonable to ask, why is it insufficient to anonymize data? Removing names and other obviously identifiable information would seem sufficient to the layman. However, simply removing explicit identifiers (e.g., name) does not preserve privacy, given that the adversary has some background knowledge about the victim. In a well-known case, the personal health information of Massachusetts governor William Weld was discovered in a supposedly anonymized public database [1]. By merging overlapping records between the health database and a voter registry, researchers were able to identify the personal health records of the governor, among others. In fact, Sweeney also shows [1] how almost 87% of Americans can be identified based on their 5-digit zip code, gender, and date of birth.

To combat such scenarios, multiple different more robust definitions of privacy were put forward, such as k-anonymity[1], t-closeness[2], and l-diversity[3]. These definitions relied on grouping samples based on their sensitive information, and the group was published if it was big enough. Provided a large enough group was released, it would seem tough for an attacker to identify individual samples. And while this certainly was better than no privacy at all, it still had its shortcomings. If the attacker somehow had information available about some of these sensitive features from another source, they could go about distinguishing and identifying individual samples.

Consider an example of a database that holds the grades and departments of 4 students, and publishes data using 3-anonymity. According to 3-anonymity, if 3 students are in the same department, their department and grades are available in response to a query. If an attacker knows that two students are in Computer Science, and one student is in Economics, then if the database does not respond to a query, the attacker will also know that the last student is not in Computer Science. And this is just an oversimplified example of a case where k-anonymity could fail us.

Other methods often include the use of aggregated statistics, or generalization. Generalization is when specific values in the database are replaced with more generalized value, thus changing the information, while still maintaining the truthfulness of the dataset. However, studies have shown that both of these methods can often fail at guaranteeing data privacy in certain real life scenarios [4].

Differential privacy ([5], [6]), which is introduced in the next section and discussed in detail in the remainder of this paper, uses random noise to ensure that the publicly visible
information doesn't change much if one individual in the dataset changes. Since output will not be significantly affected by any one individual sample, it is impossible for an attacker to confidently infer any private information corresponding to any individual sample.

A. Motivating High Profile Examples

There are many cases where anonymity wasn’t enough to preserve privacy, and differential privacy was either used or the need for differential privacy was shown. These examples serve as motivation to better understand and explore the various facets of differential privacy. Some of these examples are discussed below:

1) Netflix Dataset: The work shown and discussed in [7], [8] and [9] explores the massive public dataset released by Netflix, that was seemingly anonymous. The research shows that many individuals were identified from the dataset, when their data was compared to public posts in the IMDB dataset. Combining viewing times and ratings across the two datasets, allowed many individuals in the dataset to be de-anonymized. In the study, for a few people, the researchers were able to uncover apparent political preferences and other potentially sensitive information. This served as a very big example of a case where differential privacy could have made a huge difference.

2) Apple and Differential Privacy: Recently, Apple made a huge announcement to its users. With the launch of iOS 10, they would now be using Differential Privacy to collect user specific data for larger scale analytics[10]. This would be used to understand content browsing habits and the correlation with battery statistics, keyboard usage habits, and much more. A lot of this data is often incredibly private and having full access to it along with the identities of users can be incredibly harmful. So Apple included a way for this data to have noise added to it on the devices themselves, thus preventing availability of potentially private data on the database. Studies[11] however showed that despite this implementation of differential privacy, Apple still has to use stronger bounds to ensure guaranteed privacy of customer data.

3) Facebook and Cambridge Analytica: Recent news has made light of a massive data breach - over 80 million Facebook users’ data was accessed by analytics firm Cambridge Analytica [12], which then used that data for targeted advertising and messaging, among other tasks, in order to sway political beliefs and cause political dissent. A lot of this was done as data was collected on users and their friends through Facebook data collection, made possible by a survey that was available on Mechanical Turk. This was initially for an academic purpose, but the data was then sold to Cambridge Analytica, unfortunately without any provision for privacy protection. this is perhaps one of the biggest cases that highlights the need for differential privacy in today’s data storage and data analytics, and it will probably be a landmark event leading to potentially major changes in legal policy regarding data privacy on the cloud.

B. Contributions

This survey aims to explore the breadth of work in privacy for datasets, and more specifically the work corresponding to differential privacy. We discuss differential privacy, its formal definition, and the various application scenarios in detail. We follow this up with a discussion of various methods that have been proposed for privacy in datasets, both interactive and non interactive, and how these methods perform and stack up against each other. We also try to explore strategies for differentially private data mining and machine learning, followed by a brief discussion of challenges and the scope for future exploration in differential privacy.

II. DIFFERENTIAL PRIVACY

One of the problems with early attempts at privacy was determining what privacy specifically means. There were a number of early techniques that considered different threat models and what kind of information an adversary would possess. However, eventually most work agreed on a general definition of privacy: differential privacy.

In differential privacy, there is a trusted curator who holds the data of the individuals in a database composed of rows and columns. In the non-interactive or offline model, the curator makes publicly available a synthetic database that presents similar relationships and frequencies as the original data, but is unable to be used to acquire individual data and can’t be attacked to link to real users. This can be freely mined by any number of analysts without harming the privacy of the original users. The advantages of this model are that the data can be made publicly available and can be used for unlimited research. But, it might be difficult to use for certain types of analysis and may contain more noise than a more specific query. The other model is interactive or online querying. In this model, the curator accepts queries and runs them in a differentially private way on the dataset. This can involve altering the queries or adding noise partway through the computation. This model can be more effective for certain types of queries, but each query involves a privacy cost.

The main idea behind differential privacy is that it bounds the amount of information that can be gained about a single row (or user) in the dataset compared to if that user was not in that dataset. This is necessary because it is not possible to prevent a study from revealing information about a user. Consider a study about the relationship between smoking and lung cancer. If you are a smoker and the study shows that you therefore have a greater chance of getting lung cancer, you have received harm. However, the harm that you receive is not because you were a member of the study. You would be harmed even if you were not part of the study as long as others (such as insurance companies) learn this information. So, the harm you receive from specifically being part of the study must be separated from that of the general knowledge acquired through the study. [13]

To formalize this, let us define a database $x$ as a histogram representing the count of each element $x_i$ where the type $i \in \chi$ of the universe of all types $\chi$. Then, we can consider two adjacent
databases based on our definition of databases. The $\ell_1$ norm of a database $x$ is denoted $||x||_1$ and is defined to be:

$$||x||_1 = \sum_{i=1}^{\lvert X \rvert} |x_i|$$

Therefore, the distance between two databases $x$ and $y$ is $||x - y||_1$, and they are adjacent when the distance is equal to 1 [13].

A randomized algorithm $M$ is $(\epsilon, \delta)$-differentially private if for all databases $x$ and $y$ with distance at most 1 and $S \subseteq \text{Range}(M)$,

$$\Pr[M(x) \in S] \leq \exp(\epsilon) \Pr[M(y) \in S] + \delta$$

[13], [14]

One important guarantee is that differentially private functions are safe when considering post-processing. Therefore, a data analyst without additional knowledge of the private database cannot compute a function of the output of $M$ and make it less differentially private. Formally, for all $(\epsilon, \delta)$-differentially private functions $M$ and arbitrary randomized mapping $f$, $f \circ M$ is $(\epsilon, \delta)$-differentially private.

**Proof.** For $f : R \rightarrow R'$, adjacent $x$ and $y$, a fixed event $S \subseteq R'$, and $T = \{r \in R \mid f(r) \in S\}$:

$$\Pr[f(M(x)) \in S] = \Pr[M(x) \in T] \leq \exp(\epsilon) \Pr[M(y) \in T] + \delta \leq \exp(\epsilon) \Pr[f(M(y)) \in S] + \delta$$

which was what we wanted [13].

It is also important to consider the privacy of groups. In differential privacy, the privacy of the group drops linearly with the number of members. This is reasonable when you consider than knowledge about the group must be obtained in order for data mining to succeed. Any $(\epsilon, 0)$-differentially private mechanism $M$ is $(k\epsilon, 0)$-differentially private for groups of size $k$. So, for all $||x - y||_1 \leq k$ and all $S \subseteq \text{Range}(M)$,

$$\Pr[M(x) \in S] \leq \exp(k\epsilon) \Pr[M(y) \in S]$$

[13]

For interactive queries, each individual query involves an additional privacy cost. Because of this cost, users can not be allowed to run as many queries as they would like. If they could, it is possible to obtain the entire database in a linear number of queries [15]. Therefore, the curator must then be careful with those who are allowed to query and how many queries they can run. Before differential privacy, these methods would be more ad hoc and fail to accurately determine the privacy cost of an individual query. With differential privacy, the cost is computed against the privacy budget. For a sequence of queries with privacy bounds $\epsilon_1, \cdots, \epsilon_k$, the total privacy budget used is therefore $\sum_{i=1}^{k} \epsilon_i$. Therefore, a total limit $P_B$ can be set for the total amount of privacy cost that a user can accumulate.

However, this bound is not always tight. In [16], they found a new bound of $(\tilde{\epsilon}, k\delta + \delta)$-differentially private where

$$\tilde{\epsilon} = O(k\epsilon^2 + \epsilon \sqrt{k \log(1/\delta)}), \quad \delta = \Theta(k\delta).$$

Furthermore, they demonstrate that their bound is in fact tight for a general composition of $k$ differentially private queries.

### A. Real Usages of Differential Privacy

Many companies have used differential privacy. Much of the early research in the field took place at Microsoft. But, one of the companies who put the most effort into popularizing the technique was Apple. They try to use local differential privacy, so that the data is private on a user’s device (iPhone or Macbook) before it is ever sent to their servers for analysis. The users are even able to change the $\epsilon$ parameter for their privacy. Apple then uses the data collected to, among other things, track emoji popularity, identify high energy and memory usage in Safari, and discover new words [17]. Although there have been some questions as to the effectiveness of their privacy [18], it remains that their usage of differential privacy presents a clear attempt at privacy-protection far beyond the aggressive disregard of personal privacy that Google and Facebook are famous for.

Following them, other companies have also begun putting differential privacy into practice. Uber attempted to improve their internal analytics with differential privacy. To do so, they created a new tool to convert standard SQL queries into differentially private ones [19] and have put it into usage throughout the company. Their tool was called FLEX and was produced in collaboration with The University of California Berkeley [20]. Google has begun putting effort into federated learning, a new mechanism for distributed machine learning to avoid centralizing and acquiring user data in the first place [21]. To combine the model updates, they worked on a method for differentially private updates [22].

### B. Goals of Differentially Private Mechanisms

For differentially private mechanisms, there exists a tradeoff between the accuracy that can be gained from the data, the size of the data set, and the private requirement $\epsilon$. Assuming a fixed size dataset, there is then a pareto frontier, such as in Figure 1, between the accuracy and $\epsilon$. While it should be
obvious that this tradeoff does exist, it is important to realize that not all mechanisms are pareto efficient. A mechanism is pareto efficient when no improvements can be made to accuracy or privacy without worsening the other. For this reason, differentially private mechanisms can be evaluated mainly by the accuracy that can be gained given the same \( \epsilon \) [23].

One important possibility for optimizing that tradeoff (high accuracy per privacy) is using interactive queries instead of non-interactive datasets. However, that introduces a new tradeoff of user friendliness, that makes both options important to consider and develop. For this reason, we divide our analysis of algorithms for differential privacy into an interactive section and a non-interactive section.

### C. Interactive vs Non-Interactive Differential Privacy

This has been discussed and mentioned before, however we will attempt to reiterate and fully clarify the basic differences between the two types of differential privacy. Interactive Privacy is offered when data is made available through queries to a database. This means that the database itself doesn’t have differentially private data, but rather, every data request will return a response that ensures differential privacy. Interactive privacy in many ways is similar to some of the older privacy techniques, such as k-anonymity. Non-Interactive privacy is typically used for a release of entire datasets, where differential privacy has to be ensured across the entire dataset and every single existing sample, as opposed to just for the queried samples. We cover these two, and some example technologies and techniques that implement them, in later sections of this survey.

### III. Standard Techniques

#### A. Laplace Mechanism

One of the more common techniques integrated into other algorithms is the Laplace noise. It adds noise for a numeric query function \( f : \mathbb{N}[|x|] \to \mathbb{R}^k \). It depends on the \( \ell_1 \) sensitivity, typically just referred to as the sensitivity in most differential privacy mechanisms:

\[
\Delta f = \max_{x,y \in \mathbb{N}[|x|], \|x - y\|_1 = 1} \| f(x) - f(y) \|_1
\]

The importance of the \( \ell_1 \) sensitivity is for the addition of noise that is comparable in magnitude to the magnitude of the numbers found. The Laplace mechanism \( M_L \) applied to a function \( f \) is therefore defined as:

\[
M_L(x, f(\cdot), \epsilon) = f(x) + (Y_1, \ldots, Y_k)
\]

where \( Y_i \) are random variables drawn from \( \text{Lap}(\Delta f/\epsilon) \). The Laplace mechanism is used for most counting queries, which are common considering the frequency of the histogram model in many more sophisticated differentially private mechanisms [24], [13].

#### B. Exponential Mechanism

The exponential mechanism is common used to choose between different options that each have a different utility. Let each option \( r \) have a utility \( u(x, r) \) with the total \( \ell_1 \) range of utilities as \( \Delta u \). Each option is chosen with probability \( \exp\left( \frac{u(x, r)}{\Delta u} \right) \). This mechanism is important for making decisions when range of outputs is bounded and the utilities could have varying importance. The most standard example where the exponential mechanism might be used is for auctions where adding noise to the result will produce an error that will remove all value [25], [13].

Although these two mechanisms are simple, it is possible to compose differentially private mechanisms. And, these mechanisms can be used as a building block in more sophisticated mechanisms such as the histogram abstraction and data cubes.

### IV. Interactive Queries

In this section, we aim to discuss some common techniques and popular works focused on interactive differential privacy, i.e., responding to user made queries with differentially private versions of the data. We also look at a few types of attacks that interactive queries are in particular susceptible to.

#### A. Privelet+

Privelet [26] is a mechanism for performing range queries that provides a superior accuracy over standard Laplace noise. It works by utilizing the wavelet transformation that can convert the dataset (treated as a matrix) into a new domain and losslessly convert it back. By adding the Laplace noise in the wavelet domain instead of the standard domain, the total variance added is reduced and its results are more accurate.

One issue addressed is that the standard wavelet transformation only works for ordinal attributes, not nominal ones. This can be rectified by imposing a total ordering on the nominal domain. But, using their Nominal Wavelet Transform [26], the bound on the variance can be further bound.

While the wavelet transform reduces the asymptotic bound of the variance added, the standard Laplace method [24] works better for smaller domains. Therefore, a hybrid system, Privelet+, can be used that gains the benefits from both sides and produces an output with the minimal variance.

#### B. Privacy Integrated Queries (PINQ) Platform

PINQ [27] is an implementation of interactive differential privacy which ensures, at runtime, that queries adhere to a global privacy budget. PINQ provides private access to arbitrarily sensitive data, without requiring privacy expertise of analysts or providers. Third-party client code freely decides how sensitive data sets should be processed and queried. The run-time system ensures that this does not break a specified privacy budget \( \epsilon \). The privacy guarantees of PINQ are the unconditional guarantees of differential privacy.

PINQ builds on a collection of standard differentially private primitive queries, together with simple composition principles mathematical properties enjoyed by the definition of differential privacy. The data miner can use the interface presented by
PINQ to execute over the database, noise versions of aggregate queries such as count, sum and average, and the wrapper uses Laplace noise and the exponential mechanism to enforce differential privacy.

Some central ideas that are critical to the creation, understanding and use of PINQ are as follows:

- Multiple queries have an additive effect on the global differential privacy budget.
- PINQ keeps a track of the sensitivity of functions to estimate the effect on the value of the data by any change in the input.
- A data miner wishing to develop a data mining algorithm using the privacy preserving interface should plan ahead the number of queries to be executed and the value of \( \epsilon \) to request for each.

Together, these components allow the system to track how much to deduct from the global privacy budget on each invocation of a primitive query. The miner has to be careful with the privacy costs of the queries made in order to avoid being blocked by the database upon premature exhaustion of the global budget. PINQ however, introduces new operators that do not exist in standard SQL, so the approach is not compatible with standard databases.

C. Airavat

Airavat [28] integrates decentralized information flow control (DIFC) and differential privacy, to provide rigorous privacy and security control in the computation for the individual data in the MapReduce framework. Data providers control the security policy for their sensitive data, including a mathematical bound on potential privacy violations, while Airavat enables the execution of trusted and untrusted MapReduce computations on sensitive data, within the bounds provided by the security policy. Users without security expertise can perform computations on the data, but Airavat confines these computations, preventing information leakage beyond the data providers policy.

Just like PINQ, Airavat also relies on the concepts of a global privacy budget and function sensitivity. Airavat allows trusted and untrusted mappers, but only trusted reducers. Airavat has some inclusions to prevent encoding of sensitive information within the keys outputted by untrusted mappers, thus restricting the abilities of the untrusted mappers a fair bit. Also, reducers sort the keys, so key order can’t be used by untrusted mappers as a way to leak information. There are multiple other types of attacks that Airavat provides guarantees regarding, when using untrusted mappers.

When evaluated, Airavat does show a significant processing time overhead of nearly 32%. This is definitely not optimal, however considering the security and privacy guarantees that Airavat is able to provide, it is a reasonable cost. It is adaptable and can be used on a variety of different data formats, as was discussed in the evaluation in its paper. Airavat does have limitations presented when considering its vulnerability to timing attacks in particular, however, this is a relatively simple thing to work around and has also been discussed by the paper.

D. Weighted PINQ (wPINQ)

wPINQ [29] is built to extend the work done on PINQ. It does so with support for general equijoins in databases and works by assigning a weight to each row in the database, then scaling down the weights of rows in a join to ensure an overall sensitivity of 1. wPINQ adds noise from a Laplace distribution to results of counting queries, making it suitable for all three types of join functions in databases. However wPINQ is not compatible with all databases. Due to its use of a custom runtime, applying wPINQ in an existing database would require modifying the database to propagate weights during execution.

E. Elastic Sensitivity - Differential Privacy for SQL Queries

Existing differential privacy mechanisms do not support the wide variety of features and databases used in real-world SQL-based analytics systems. FLEX [30] is a system built at Berkeley to enforce differential privacy for SQL queries using elastic sensitivity. [30] discusses how FLEX is compatible with all existing databases, manages to enforce differential privacy requirements for all SQL queries, and has a negligible performance overhead.

FLEX relies on the concept of elastic sensitivity. Elastic sensitivity is a novel approach for calculating an upper bound on a query’s local sensitivity. Global sensitivity, as discussed earlier with PINQ, does not have adequate generalized support for joins in queries. Elastic sensitivity leverages local sensitivity for queries with general equijoins. Its approach models the impact of each join that is represented in the query, using precomputed metrics about the frequency of join keys in the true database. This allows the method to compute
approximate local sensitivity without additional interactions with the database.

Elastic sensitivity supports a number of different aggregation functions such as sum, average, max and min. Also, calculations for elastic sensitivity can optimize for non-sensitive information in the database, helping create tighter bounds for the approximation of local sensitivity. Due to its low computational cost, its adaptability to almost all existing database formats, the existing implementation in the form of FLEX, and the general privacy guarantees provided by it, Elastic sensitivity can be seen to be a very effective method for ensuring differential privacy.

F. Machine Learning

When developing machine learning models based on data, it has been shown that it is possible to use hill-climbing on the output probabilities of a model to reveal specific inputs to the data [31]. Considering that these examples are more likely the interesting ones who might need privacy the most, this could be a large danger to including sensitive data in any model.

To prevent this, differential privacy can be interactively added to the machine learning training. Some techniques will train on a clean data set and then use the Laplace or Exponential mechanisms to add noise to the final data. For iterative algorithms such as Stochastic Gradient Descent based Neural Networks, noise can be added on each iteration. This noise can be put into the training updates. For objective perturbation, the noise is added to the target function.

Another example, Semi-Supervised Knowledge Transfer [32] is a technique that adds a layer of indirection has to be added to separate the final model from the dataset. The dataset is therefore partitioned into n partitions which are each used to train a teacher model. Incomplete public data is then labeled using a differentially private aggregation that adds Laplace noise to the teacher outputs. The student can then be trained with a GAN that transforms vectors with a Gaussian distribution and forces the student to emulate the teacher. The student model therefore does not have direct access to any individual training example and therefore cannot reveal it.

1) State Attack: The simplest problem is a state attack. This is where there exists a leak between sections that should be isolated. For example, Airavat runs each map in a modified JVM, but does not protect against static variables. These can be used to pass data and magnify the effects of particular instances to check for existence. Another similar idea is throwing an error to pass additional information. To counter this, a default value should exist for each query.

2) Privacy Budget Attack: Another type of attack exists on the computation of the privacy budget. If the budget is computed dynamically, this represents an additional bit of privacy cost that could be leveraged. The easiest solution to this is to have a static privacy budget computation so that the querier knows their query cost beforehand without revealing any information about the dataset.

3) Time Attack: The final, and trickiest, type of side-channel attacks is with time. For example, a query could run for 5 hours if it finds that a user has watched a particular video, otherwise return instantly. By using the time needed for a result to be returned, it gains additional information that was not considered in the privacy budget and could surpass the differential privacy. This can not be easily fixed by adding a timeout because returning instantly could be used as a way to pass information instead. The best option would then to run the query on each row in a fixed amount of time and return a default value in the case of timeouts. However, it may not always be possible to determine a time bound and this method additionally reveals the number of rows in the dataset. Another option is to add a delay after each query with an additional amount of noise. This does increase both the $\epsilon$ and $\delta$ of the privacy cost, however. [33]

V. NON-INTERACTIVE

The goal of Non-Interactive differential privacy methods is to produce a synthetic dataset that preserves the frequencies and relationships found in the original dataset. However, it should retain the differential privacy guarantees for individuals. Following this, it can be publicly released and used for various data analytics tasks by anyone who wants to use it.

There are several categories of techniques commonly used:

- Histogram - Produce a histogram that represents the counts of combinations of data attributes
- Sampling and Filtering - Avoid publishing huge contingency tables by filtering out small counts and rare items
- Partitioning - Release data optimized for range queries by partitioning the data based on uniform ranges of attributes through a KD-Tree so that the approximate count in each range can be found
- Dimensionality Reduction - Reduce the number of attributes using standard techniques so that another technique can be easily divide the data
- True Synthetic Databases - Produce a new dataset that follows the same distributions as the original dataset

Most of the following differential privacy mechanism uses techniques that fall into one or several of these categories [34].
A. DiffGen

Generalization is a privacy technique that replaces the range of an attribute with a less precise range. Numerical attributes can be replaced with ranges through a binning process. Categorical attributes can be replaced by grouping the various attributes together.

In DiffGen, begin by constructing a taxonomy tree (see Figure 5 for each attribute where the root of the tree is fully generalized, and going further down has greater specialization. The leaves of the tree represent the original fully-specialized version.

![Fig. 5: Taxonomy Tree of Attributes from [14]](image)

The mechanism will then iteratively specialize while it remains under the privacy budget. During each iteration, probabilistically choose an attribute to specialize by a level. The attribute should be chosen based on a score including the information gain and the summation of the highest frequencies related to the sensitivity. These can be combined into a contingency table such as Figure 6. Finally, the counts for each set of generalized attributes should noisily counted with the standard Laplace noise mechanism.

One important parameter is the number of specializations. Their system was tested by performing classification on the resulting synthesized dataset. They found that there is a curve with an optimal number of specializations. Too few will reduce the data precision by an excessive amount while too many specializations actually increases the impact of the Laplace noise added to the final counts. [14]

B. MultiDimensional Partitioning

There are two algorithms as part of the MultiDimensional partitioning mechanism, with the second dependent on the first. The first is known as the Cell-based algorithm. In this method, the data is considered as points on the interior of a data cube. The only feature that is released are the counts of different regions of that multidimensional data cube. And, to add privacy, different partitions of the data cube are joined and then altered by Laplace noise. There are two sources of error in this scheme: approximation error due to the noise added and perturbation error due to potentially unequal distribution of points within a single partition. Finding the right balance between the tradeoffs of these two errors is therefore the importance of the algorithm. In the cell-based algorithm, the data should be partitioned on all domains and then the noisy count of each partition is released based on the privacy parameter.

The second algorithm is K-d tree based. Begin by generating a synthetic dataset using the Cell-based algorithm with half of the privacy parameter. Then, the synthetic dataset should be partitioned iteratively using a K-d tree. The k-d tree should attempt to partition the data to optimize for uniformity with such metrics as information entropy and variance. Once it is fully partitioned, the keys found through the K-d tree on the synthetic database should be used to repartition the original database. Then, the noisy counts for each of those partitions should be released as the final resultant synthetic database.

The cell-based algorithm is used to prevent the choice of partitions from revealing too much additional information about the breakdown of the data. So, using it allows for a more optimal usage of the privacy budget [35].

C. Data Cubes: Optimizing Noise Sources and Consistency

When aggregating over differentially private data cubes, the noise increases. So, if noise were added to the base cube, the variance of the noise added would be great for the top level cell (see Figure 7). To combat this, noise is instead added to a small set of cuboids independently and then aggregates are computed off of this set. However, the result would not be consistent and the counts may not add up. While this could be explained and is perfectly reasonable considering the methods, it would undermine faith that many would have upon first seeing the data. Therefore, the noisy DP data would then be altered to enforce consistency so it seems like normal data.

The problem with choosing the set of cuboids to compute with noise is that it requires enumerating all possible set choices, a $O(2^d)$ operation. Therefore, an approximation is necessary. It assumes a method Feasible that finds a subset of cuboids that cover the full set of cuboids with a total variance below a maximum threshold and a maximum cardinality of the subset. This can be computed with a greedy approximation algorithm based on SET COVER. To find the variance, binary search can be performed. If there is room within the variance and it is possible to add the base cuboid, it covers the full set of all cuboids.

After the set of covering cuboids is found, there are aggregated to fully fill out the data cube. Then, the problem of finding a consistent solution is equivalent to minimizing the L2 distance is a linear programming problem. If can be solved through the least-norm problem. Since finding the consistency does not use the original data, it does not further reduce the privacy. One issue with this is that the requirement for adding the consistency means that the full consistency must be stored, which is not reasonable for data cubes of high dimensionality [36].

D. LoPub

One issue with many of the above mechanisms for non-interactive data release is the treatment of high-dimensional data. LoPub attempts to resolve this through the use of dimensionality reduction to avoid correlations between attributes. It begins by outputting locally sanitized and centralizing it in a server. Then, they use a novel EM and Lasso based technique for finding a multi-dimensional joint distribution estimation that works even with the sparsity of high-dimensional data. Then, they apply that mutual information into the form of an undirected graph. The graph is pruned using a special heuristic
Fig. 6: A raw data table and contingency tables both with and without generalization (pre-Laplace noise) [14]

Fig. 7: Lattice of Cuboids and Variance Magnification from [36]

pruning scheme to improve performance, then clusters of low-dimensional data are formed. With these, a new synthetic dataset can be formed that avoids the issues of correlated attributes and remains more secure.

Unlike similar locally differentially private schemes like RAPPOR [37], it is also designed to be more efficient with communications. It requires only linear communication per the number of attributes while RAPPOR is exponential in the number of attributes. This will enable much higher dimensional data which is important for finding essential relationships within the data. Furthermore, the time and space complexity are greatly reduced.

E. GAN

One idea for protecting differential privacy is through the use of Generative Adversarial Networks (GANs) [38]. GANs are a technique from deep learning that have found wide use in their ability to generate items that resemble a data set, even for such sophisticated items as generating images based on sketches. GANs work by creating two neural networks: a generator and a discriminator. The generator aims to take some input (such as random noise) and produce an output that can fool the discriminator. The discriminator attempts to discern whether an input/output pair come from the training examples or the GAN. When they reach an equilibrium, the GAN should produce results that can not be discerned by the discriminator.

In this instance, the GAN can take as input random noise and output artificial elements that follow a similar distribution to those of the original dataset. Additionally, it should not possess any difference in the correlations and relations between attributes that could be inferred from the discriminator, therefore it should retain a reasonable amount of the information. But, because of the high level of obfuscation, it should be capable of maintaining a very high amount of privacy. Due to the reduction in data, it is possible for this to have one of the lowest privacy budgets of all techniques in this paper.

Fig. 8: The GAN architecture. The sensitive data X and Generator output feed into the Discriminator and pass through the privacy preserving layer noted with a dashed line. The generator can then output an artificial dataset $\tilde{X}$ [39].

In order to maintain differential privacy with this method, gaussian noise is added to the results of the generator. Before data passes through the discriminator, additional differential privacy noise is further added to prevent the network from too closely mimicking individual dataset elements (see Figure 8). For data that has labels, they can either be handled as standard attributes or through an additional privacy preserving method [39].

Another effort towards this is dp-GAN [40]. They found that the $\epsilon$ privacy cost is linear in the number of iterations to convergence, so they added a number of optimizations to improve the operation. The first is adapting clipping which restricts the sensitivity of each individual variable. This helps reduce the privacy cost used per iteration. Their other improvement was to initialize the GAN execution using a public data set. The stronger starting point allows better usage of the privacy costing iterations as fine tuning instead of providing general direction.

VI. MERITS AND DEMERITS OF DISCUSSED TECHNIQUES

We attempt to give an overview of the various techniques we discussed, along with a comparison of their merits and demerits, in an effort to identify certain methods that will be preferable over others, and to get a brief summary of these strategies. Take a look at Table I for the full overview.
<table>
<thead>
<tr>
<th>Technique</th>
<th>Merits</th>
<th>Demerits</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-anonymity</td>
<td>Intuitive, easy to use, effective when not too many dimensions of sensitive information.</td>
<td>Not differentially private, shown to fail in many scenarios.</td>
</tr>
<tr>
<td>Generalization</td>
<td>Simple to understand, straightforward to use, works well in many simplistic cases.</td>
<td>Not differentially private, shown to fail in many scenarios, not optimized for a larger number of sensitive dimensions.</td>
</tr>
<tr>
<td>Privelet+</td>
<td>Lesser variance added, more accurate results, hybrid model that gets benefit of wavelet transform and Laplace model.</td>
<td>Works only for ordinal attributes, slightly costly.</td>
</tr>
<tr>
<td>PINQ</td>
<td>Private access for arbitrarily private data without need for expertise of analysts or providers. Provides unconditional guarantees of differential privacy. Acts as an interface on top of a database.</td>
<td>Not compatible with standard databases, adds its own operators on top of traditional SQL operators.</td>
</tr>
<tr>
<td>wPINQ</td>
<td>Extends PINQ, adds support for general equijoins in databases. Adds noise from a Laplace distribution, is suitable for all 3 types of joins in databases.</td>
<td>Not compatible with all databases, uses a custom runtime, common databases can’t propagate weights during runtime.</td>
</tr>
<tr>
<td>Airavat</td>
<td>Adaptable, suitable for different data formats, flexibility with trust requirements for machines, easy to use for users without security expertise.</td>
<td>High processing overhead, vulnerable to timing attacks.</td>
</tr>
<tr>
<td>Elastic Sensitivity/ FLEX</td>
<td>Best existing interactive privacy method. Scalable, can be used on top of any existing database technology, uses local sensitivity to have better support for joins in queries. Negligible difference in performance.</td>
<td>None in particular. FLEX is incredibly powerful and capable.</td>
</tr>
<tr>
<td>DiffGen</td>
<td>Early differentially private non-interactive release, simple to formulate and use</td>
<td>High sensitivity to to the number of specializations/generalization categories used, requires manual effort for categorical taxonomy trees, relatively high noise.</td>
</tr>
<tr>
<td>Multidimensional Partitioning</td>
<td>Better privacy budget utilization, partitioning allows approximation of any region, subregion, or section of the input space (better than histogram methods)</td>
<td>Accumulating error for summing up regions and errors when attempting to analyze part of a single region</td>
</tr>
<tr>
<td>GANs</td>
<td>Very high privacy and accuracy, generates synthetic data set that is easiest to train since it is most similar to true dataset, can generate arbitrary amounts of data, space usage constant to number of rows in dataset</td>
<td>May fail to converge, long training time, may require public sample data for initialization, could encode too much data in model parameters</td>
</tr>
<tr>
<td>Data Cubes</td>
<td>Deals with the possibility of having too much noise when aggregating over the data cube, data cube format provides fast analysis as it caches higher dimensions</td>
<td>Feasibility issues with high dimensional data cubes, costly operations for noise addition.</td>
</tr>
<tr>
<td>LoPub</td>
<td>Locally differentially private, intended to improve communication costs as a distributed system, removes correlations with high dimensional data through dimensionality reduction</td>
<td>More sophisticated model to handle the additional difficulties of the local model and dimensionality reduction</td>
</tr>
</tbody>
</table>

**TABLE I: Merits and Demerits of Various Techniques**

VII. DATA MINING WITH DIFFERENTIALLY PRIVATE DATASETS

With differentially private datasets, noise is added to each parameter. Consider as an example a dataset that contains a single boolean attribute. The simplest differential privacy method for this is to return true $\frac{3}{4}$ of the time, return false $\frac{1}{4}$ of the time, and return the actual answer the remaining $\frac{1}{4}$ of the time. When counting, the effective size of the dataset is only half of the actual size and the approximately half of the people who return fixed values have to be discounted in order to find the actual ratio of true to false responses.

In a similar manner, the same kind of noise is applied to multi-attribute relationships like correlations. When mining for this data, the patterns found would be similar to the actual ones, but with reduced magnitude. Therefore, when mining the frequencies and correlation magnitudes should be corrected either beforehand or after the mining. Some problems like classification could still be done on the data like normal but will just have poorer results on the training set then on a non-private test set.

VIII. ADVANCED TOPICS AND FUTURE DIRECTIONS

A. Models Beyond the Trusted Curator

In the standard differential privacy model, it is assumed that there exists a central trusted curator who has all of the raw data. Then, that curator will either facilitate queries in the interactive mode or release a synthetic dataset in the non-
interactive mode. However, what there exist other models for when no such trusted curator exists.

1) **Local Differential Privacy:** One of the first examples is the local model. In this model, the data is not centralized and stays on the device of the user who owns the data. Those users can then be queried and will each add noise on their own. One example of the local model in practice is the work performed by Google on Federated Learning [21]. However, this model has also been taking off greatly and is featured in many new papers under the name of Local Differential Privacy (LDP) [41], [42], [43]. It presents a greater privacy guarantee under OLAP than differential privacy by avoiding the requirement for a trusted curator, a requirement getting increasingly concerning as the trust in large corporations drops.

2) **Pan-private Streaming:** Another model is the pan-private streaming model. In this model, the curator can generally be trusted, but might suffer an intrusion. These intrusions can be legal (subpeona) or illegal (hacking). To combat this, the data must be made differentially private once received so that it can’t be misused later in a way that infringes upon the privacy. Apple uses this model by actually applying differential privacy to the updates before they are even centralized [17].

3) **Continual Observation:** The last additional model to consider is for continual observation. While most of the work is for OLAP data, live OLTP might sometimes be necessary as well. To handle this scenario, noise has to be added to the counts before counting, rather than after. It can also be handled in a similar manner as pan-private because they both involving privatizing the data immediately rather than as a final step [13].

Another missing piece that follows the idea of continual observation is multiple non-interactive data releases. Naively, each one can be considered a separate query that combine in terms of privacy cost. As more are released, the privacy degrades. Some new methods have begun analyzing this issue [43].

### B. High-Dimensional Data

Another problem with differential privacy is in the treatment of high-dimensional data. Normal differential privacy assumes that the attributes are independence. When this is false, the privacy degrades exponentially with the correlation of the attributes eroding the privacy guarantees. Recent work has been trying to fix this by dimensionality reduction [42], Bayesian Networks [44], and dependence perturbation [45].

### IX. Conclusion

We have discussed differential privacy in considerable detail here. We looked at the need for differential privacy, and many actual scenarios that could be much safer with the use of differential privacy. We try to understand why differential privacy is the need of the hour and why it needs more adoption than a few other more intuitive techniques that are often used. We cover a formal analysis of differential privacy and the requirements laid out under the umbrella of differential privacy. We then split differential privacy into two categories: interactive and non-interactive. Following this we try to analyze some popular research and techniques under each of these categories, trying to identify methods that significantly outperform others, and perhaps to take a look at commonalities and differences between different methods. We then try to gain a better understanding of how data mining differs when datasets are made differentially private, as well as take a look at what the future goals and challenges for differential privacy will be, and how we can circumvent those challenges effectively.

This paper has explored a number of topics and presents a vast variety of information, techniques, and possible future work.

### References


