


Differential privacy: a privacy cloak for preserving utility in heterogeneous datasets

Saurabh Gupta¹ · Arun Balaji Buduru¹  · Ponnurangam Kumaraguru²

Received: 19 January 2022 / Accepted: 5 February 2022 / Published online: 1 March 2022
© CSI Publications 2022

Abstract Data has become an integral part of day-to-day human life. Users leave behind a trail of digital footprint that includes their personal and non-personal information. A normal user puts 1.7 megabytes of data every second into the hand of service providers and trusts them to keep it safe. However, researchers have found out that in the name of improving the quality of service, the service providers, knowing or accidentally, put users' personal information at risk of getting into the hands of an adversary. The service providers usually apply masking or anonymization before releasing the users' data. Anonymization techniques do not guarantee privacy preservation and are proven to be prone to cross-linking attacks. In the past, researchers were able to successfully cross-link multiple datasets to leak the sensitive information of various users. Cross-linking attacks are always possible on anonymized datasets, and therefore, service providers must use a technique that guarantees privacy preservation. Differential privacy is superior for publishing sensitive information while protecting privacy. It provides mathematical guarantees and prevents background knowledge attacks such that information remains private regardless of whatever information an adversary might have. This paper discusses how

differential privacy can help achieve privacy guarantees for the release of sensitive heterogeneous datasets while preserving its utility.

Keywords Differential privacy · Heterogeneous data · Preserving utility

1 Introduction

Recent advancements in computing have enhanced the way users and organizations interact with data. From personalized recommendations, e-commerce to industrial and scientific research, data is needed everywhere, and that too in vast abundance. There are 4.6 billion internet users worldwide¹ (approximately 60% of the world population), each generating 1.7 megabytes of data every second, according to a report². The generated data, in turn, is used with algorithms in machine learning and deep learning to create solutions for self-driving cars, recommendation engines, automated game playing, and so on.

The generation of data is diversified among several domains, therefore, making it heterogeneous. In an Internet of Things (IoT) scenario, data corresponds to values read by installed sensors across the ecosystem. At the same time, a self-driving car captures videos and images along with sensor values. Similarly, a smart meter generates time-series data of power consumption at the home it is installed. The method of parsing a video is different from that of an image. Similarly, naked sensor values are handled by entirely different tools and techniques. Due to its

✉ Arun Balaji Buduru
arunb@iiitd.ac.in

Saurabh Gupta
saurabhg@iiitd.ac.in

Ponnurangam Kumaraguru
pk.guru@iiit.ac.in

¹ Indraprastha Institute of Information Technology - Delhi, Delhi, India

² International Institute of Information Technology - Hyderabad, Hyderabad, India

¹ <https://www.statista.com/statistics/617136/digital-population-worldwide>

² <https://www.domo.com/solution/data-never-sleeps-6>

heterogeneous nature, coming up with a single solution to protect data privacy becomes a daunting task.

To understand privacy, we first need to understand the categorization of data in the context of privacy. All kinds of heterogeneous data, be it images, videos, text, or any other format, can be categorized as PII (Personally Identifiable Information), pseudo sensitive data, and non-sensitive data. As the name suggests, PIIs are highly sensitive and must be protected from any leak. Pseudo-sensitive data is one where we can choose to protect it based on its application domain. For example, while performing anonymization on a dataset that is to be shared with an employer, the gender attribute might not be considered a PII because just knowing the gender one cannot identify the person. But if it is known that the same employer prefers male candidates over female candidates, gender can be considered as a sensitive attribute because revealing it will cause harm to the candidate.

The vast amount of data about a user has created their digital footprint. The adversary (or a service provider) can use the digital footprint to tell a lot about its user. Therefore, it is crucial to protect it using privacy-enhancing technologies. The digital footprint consists of the user's private information and public information. There are clear distinctions between what is needed to be protected and what can remain public. For example, we might easily give people our Facebook, Twitter, or LinkedIn username with little or no hesitation. Still, we will think twice before offering someone our social security number, credit card details, and address. In an ideal scenario, sensitive data like social security or credit card details must be kept separate from the non-sensitive data.

In the preliminary work towards privacy protection, data augmentation, shuffling, and masking methods were used to obfuscate and hide sensitive attributes [1–3]. In these methods, data is either shuffled to become random or hidden via masking two or more characters present in the data - for example, an email, abc@example.com, when masked might look like a**@example.com. Shuffling might replace the email with another email present in the dataset, xyz@example.com. The techniques are trivial, and data is still exposed, giving minimal privacy protection. In one of our ongoing works, we found a dataset where first four letters of email addresses are masked, and other information like first name, last name were given³. In 90% of the cases, the first four letters are found to be nothing but the first four letters of the first name of the individual. Hence, masking techniques are vulnerable to re-identification if additional information is known.

To curb this visibility, several anonymization techniques like k-anonymity [4], l-diversity [5], t-closeness [6] were

invented. k-anonymity uses suppression and generalization to divide the dataset into k groups. k-anonymity is prone to homogeneity attacks as it prevents identity disclosure but still vulnerable to attribute leakage. l-diversity uses generalization and masking within each of the k-groups created using k-anonymity to diversify sensitive attributes and overcome homogeneity attacks. l-diversity is not effective for single sensitive attributes and is difficult to achieve. t-closeness extends l-diversity by ensuring that the distribution of a sensitive attribute in any k-group is close to the distribution of a sensitive attribute in the overall distribution. While anonymization proved to be effective, researchers have shown it to be vulnerable to de-anonymization attacks [7–10]. Netflix released an anonymized dataset to improve their search recommendations. The health records are anonymized and made public in the US. The American Online Query Logs (AOL) are anonymized and released for research purposes, but researchers successfully deanonymized them and were able to find exact users [11, 12].

Techniques like anonymization are vulnerable and harder to apply where a complete dataset is unknown (stream data) or has images and videos. Even when anonymization is successfully applied, there is a risk of de-anonymization. Therefore, with continuous efforts from researchers, differential privacy has grown to be a universal and a go-to way to release information publicly. Unlike anonymization, the idea behind differential privacy is that if the effect of making an arbitrary single substitution in the database is small enough, the query result cannot be used to infer much about any single individual, and therefore provides privacy [13].

Differential privacy provides a way to share information publicly (or release datasets) without compromising the privacy of individual samples present in the dataset [14]. The former is achieved by describing the patterns experienced by the groups within a dataset. Another way differential privacy is used is when instead of revealing the full dataset, only aggregate information is released, limiting the disclosure of private information of records. For example, organizations publish statistical aggregates like percentages and mean values to ensure the confidentiality of survey responses. We will explain differential privacy in Sect. 3 in more details. With its increasing popularity, differential privacy is being used in multiple domains and modified to be applied to heterogeneous data. In 2018, Facebook used differential privacy to release a dataset for researchers to study the role of social media in elections and democracy⁴. We are going to explore the application of

³ <https://engineeringstudentsdata.com>

⁴ <https://research.facebook.com/blog/2020/02/new-privacy-protected-facebook-data-for-independent-research-on-social-medias-impact-on-democracy>

differential privacy for private data release with the following types of data: (i) images [15, 16]; (ii) high dimensions [17, 18]; (iii) time series [19]; (iv) personalized recommendations [20]; (v) streams [21, 22]; (vi) graphs [23]; (vii) statistical computations [24]; (viii) internet of things [25]. Each of the cases mentioned above is discussed in more detail in Sect. 6.

In the remainder of the paper, we first talk about the risks involved in releasing data in public via releasing a sanitized dataset or providing a query interface that fetches anonymized data in Sect. 2. Then we give a brief introduction to differential privacy in Sect. 3. We also talk about how it is infused in machine learning, and deep learning as a lot of high utility and heterogeneous computations involve them to some extent in Sect. 4. In Sect. 5, we discuss the release strategies widely used with differential privacy. Further, in Sect. 6, we discuss how differential privacy is used with several heterogeneous data, including graphs, sequential and time-series, streams, and so on. We then mention its limitations in Sect. 7 followed by real world use cases in Sect. 8. Finally, in Sect. 9, we conclude with a few ideas where the future of differential privacy is headed.

2 Risks involved in releasing data

2.1 Re-identification of anonymized records

De-anonymization of anonymized records can cause direct harm to individuals, and therefore, organizations are reluctant to make a dataset public. If it happens, a breach of privacy causes the organization a loss of trust and their reputation in keeping the data safe. Consider an example where hospitals release anonymized data to know statistics about patients admitted daily. Now, an attacker can use attributes like timestamps or a unique disease and match them with other public health records to reveal an individual's information. The State of Washington sells patient-level health data for \$50. A research study shows that the health data can be purchased and cross-linked to reveal anonymized information [26]. The authors also showed that 87 percent of all Americans could be uniquely identified using only three bits of information: ZIP code, birthdate, and sex.⁵ We did a similar research where we took election data from Twitter and successfully cross-linked it with publicly available electoral rolls to identify a Twitter user's address [27].

⁵ <https://arstechnica.com/tech-policy/2009/09/your-secrets-live-online-in-databases-of-ruin>

2.2 Queries over large sets are not protective

One might think that having a large enough dataset will provide better privacy, the truth is, the dataset becomes prone to differencing attacks [28]. Suppose in a survey, it is known that entity A is a smoker in database D. Answering the two queries in D: “How many people in the dataset are smokers?” and “How many people, not named A, in the database are smokers” yields whether A is a smoker or not. The following demonstrates a differencing attack: The summation function in SQL is not differently private. Say we want to find the specific value of a person in that database. What we need to do is to first find the sum of values for all individuals in the database, then find the sum of the database after removing this individual.

```
SELECT count(*) from table; SELECT count(*) from
table WHERE name!="john";
```

And finally the difference between the two sums, would result in the value of this individual. If we have the value of two queries like the ones above, we could easily compromise the data value for “john”.

2.3 Query auditing is problematic

There are two problems in the case of query audits to check whether a sequence of queries and their responses would compromise privacy. First, the algorithm's refusal to respond to a query can be disclosive in itself. Second, auditing historical queries can be computationally expensive and infeasible. If the query language is sufficiently rich, an algorithmic procedure that analyzes if a pair of queries constitutes a differencing attack may not exist. Therefore, manual inspection will be required, which might not be a feasible solution.

2.4 Summary statistics are not safe

Summary statistics are prone to a variety of reconstruction attacks against a dataset where each individual has a “secret bit” of information to be protected. Consider a utility-based query asking, “how many people having property P have a secret bit as 1?”. The goal of an attacker is to successfully guess the secret bit value of as many individuals as possible. The reconstruction can look like multiple queries adjusting a few conditions and boiling the responses down to a few.

3 Differential privacy

“Differential privacy” (DP) refers to an assurance made by a data collector (can be an organization or an individual user) to the data owners (entities whose data is being

collected) that their data is safe and protected. They will not be affected by allowing their data to be used in a study no matter what other information or data sources are available. DP is a protection guarantee from privacy breaches due to cross-linking of multiple data sources, which is better than anonymization techniques. Data cannot be fully anonymized and remain functional. The richer the data, the more exciting and valuable it is. This has led to notions of “anonymization” and “removal of personally identifiable information”. The hope is that the data owner can suppress portions of the data records and the remainder published and used for analysis. The Fundamental Law of Information Recovery states that “overly accurate answers to too many questions will destroy privacy in a spectacular way”. In ideal cases, differential privacy-based database mechanisms can protect privacy while making confidential data widely available for accurate data analysis without requiring data protection plans and usage agreements.

The paradox of discovering useful information about a population without compromising even a single bit of information about an individual belonging to that population is addressed using differential privacy. For example, suppose a hospital surveys individuals to know whether they smoke for research purposes. These individuals own health insurance, which might be affected if information about an individual’s smoking habits reaches the insurance provider. Differential privacy guarantees that the hospital will be able to conduct its experiments, in a way such that, even if the insurance provider has access to the data involved, they will not be able to link it to an individual with complete certainty. It ensures that the same conclusions, for example, smoking causes cancer, will be reached, independent of whether any individual opts into or opts out of the data set. Specifically, it ensures that any sequence of outputs (responses to queries) is “essentially” equally likely to occur, independent of the presence or absence of any individual.

DP introduces privacy in a dataset by introducing randomness to individual data samples. An example of privacy using randomized response, a technique developed in the social sciences to collect statistical information about embarrassing or illegal behavior, captured by having a property P . For this example, let’s assume the property P is “smoking”. Participants are asked to answer whether or not they smoke by flipping a coin. If it is a head, answer truthfully. If it is a tail, flip another coin and respond “Yes” if heads and “No” if tails. The same is demonstrated in Fig. 1.

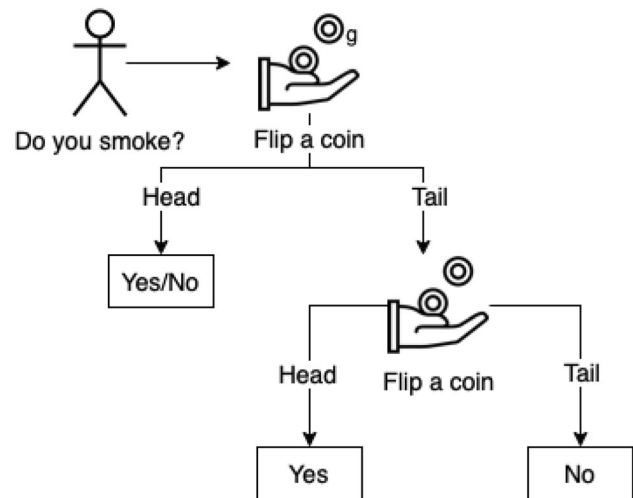


Fig. 1 Plausible Deniability. An example of privacy using randomized response. Participants are asked to answer whether or not they smoke by flipping a coin. If it is a head, answer truthfully. If it is a tail, flip another coin and respond “Yes” if heads and “No” if tails

Any outcome in the experiment shown in Fig. 1 has plausible deniability. If the insurance agency tries to cancel your insurance because they think you are a smoker, you can argue that the probability of me smoking is at least $1/4$. The insurance agency can never say with a 100% surety that you smoke. Therefore, privacy comes from the plausible deniability of any outcome. However, the noise in data comes from the procedure through which we introduce randomness.

Randomization is necessary for any non-trivial privacy guarantee to hold. With randomness, the guarantee will hold regardless of how much auxiliary information is at your disposal. In the above example, the expected number of “Yes” answers is $1/4$ times the number of participants who do not smoke plus $3/4$ times the number of people who smoke. If “ s ” fraction of users are actual smokers, then the expected number of smokers is $(1/4)(1 - p) + (3/4)p$ and turns out to be $1/4 + p/2$. Even if I remove a sample from the experiment, the probability of the expected outcome remains the same due to its dependence on the randomness of the noise, which is tossing two coins in this case. Based on this, differential privacy is defined as:

Definition 1 (Differential Privacy, DP) A randomized algorithm A_P is (ϵ, δ) -differentially private if for any two databases D and D' differing in a single point and for any subset of outputs S [29]:

$$P(A_P(D) \in S) \leq e^\epsilon \cdot P(A_P(D') \in S) + \delta$$

where $A_P(D)$ and $A_P(D')$ are the outputs of the algorithm for input databases D and D' , respectively, and P is the randomness of the noise in the algorithm.

The typical values δ are less than the inverse of any polynomial in the size of the database. The delta values are of the order $1/n$, n being the size of the database, are dangerous as they are prone to “preserving privacy” by publishing a small number of database records. The values of ϵ can be adjusted to increase/decrease the amount of privacy depending on the utility function understudy. ϵ is used to balance the privacy and accuracy level. If ϵ is small, then more privacy is preserved but data accuracy gets worse. If ϵ is large, privacy will be worse but data accuracy will be preserved. Note that ϵ goes from 0 to infinity. The typical ϵ values used in practice range from 10^{-2} to 10^4 . Choosing an optimal ϵ value depends on the “privacy budget”, which is the amount of permissible noise.

Apple uses differential privacy in order to maintain their users’ privacy. They run operations of the app data they receive from the devices and use a privacy budget with epsilon as 4 for lookup hints, same for emoji, and an epsilon value of 8 for quick type feature [30].

4 Differential privacy for machine and deep learning

One of the most valuable tasks in data analysis is machine learning or deep learning. It is a problem of automatically finding a simple rule to predict specific unknown characteristics of never-before-seen data accurately. In the works that study differential privacy in deep learning, [31] change the model’s training algorithm to make it private by clipping and adding noise to the gradients. The authors also propose a privacy accounting technique and introduce a moments accountant that computes the privacy costs. Algorithm 1 shows an infusion of differential privacy into the widely used stochastic gradient descent algorithm in deep learning.

Algorithm 1 Differentially Private SGD: an infusion of differential privacy into the widely used stochastic gradient descent algorithm in deep learning.

Require: Samples x_1, x_2, \dots, x_N , loss function $L(\Theta) = (1/N)\sum_i L(\Theta, x_i)$. Parameters: learning rate η_t , noise scale σ , group size L , gradient norm bound C .

Ensure: Initialize Θ_0 randomly.

```

1: for  $t \in [T]$  do
2:   Take a random sample  $L_t$  with sampling probability  $L/N$ .
3:   Compute Gradient
4:   For each  $i \in L_t$  compute  $g_t(x_i) \leftarrow \nabla_{\Theta_t} L(\Theta_t, x_i)$ 
5:   Clip Gradient
6:    $g'_t \leftarrow g_t(x_i) / \max(1, \frac{g_t(x_i)_2}{C})$ 
7:   Add Noise
8:    $g'_t \leftarrow \frac{1}{L} \sum_i (g'_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 I))$ 
9:   Descent
10:   $\Theta_{t+1} \leftarrow \Theta_t - \eta_t g'_t$ 
11: end for
12: Output  $\Theta_t$  and compute the overall privacy cost  $(\epsilon, \delta)$ 

```

Further, starting from that, authors use differential privacy with a parallel and asynchronous training procedure for a multi-party privacy-preserving neural network [32]. It involves transmitting local parameters between server and local task, which has a high risk of information leakage through man-in-the-middle attacks. [33] models a private convolutional deep belief network by adding noise on its objective functions and an extra softmax layer. In [15], authors use generative adversarial networks with differential privacy, and leverage the moments accountant and the private training procedure from [31] to train a differentially private generator. Authors add noise to the training procedure and avoid a distributed framework to prevent information leaks. Advantages of DPGAN’s techniques over other methods made them a salient choice for privacy preservation in another framework proposed to generate private dataset [15].

5 Strategies for data release

5.1 Interactive model

An interactive model provides an interface to the data requesters to make queries and fetch data. Differential privacy is applied to the data query algorithms. When the request from the data requester is received, it brings the raw data from the database, sanitizes it, and returns it to the data requester. The query number is restricted by this model's privacy budget ϵ . Here, more queries mean a smaller ϵ value, which means that more noise is added to the query result. Therefore, an optimization is required between the maximum number of queries under a limited budget ϵ . Implementing interactive models is computationally expensive and requires regular maintenance and auditing.

Figure 2 describes the interactive model for data release. The data requester makes a request, the data owner applies an algorithm that takes the request as input, fetches responses, and adds noise to the results. The results are then returned to the requester. Such models imply a differential privacy approach where a privacy mechanism is applied at an individual response level.

5.2 Non-interactive model

In non-interactive mode, the individuals have to trust the data curator, and they transfer data ownership to the collector once the information is filled. The data curator then applies differential privacy mechanisms to add noise to the complete dataset. The new sanitized dataset is then entirely released to the data requester. The key to this model is to design the privacy mechanism so that the utility function understudy and the privacy budget ϵ are optimized. Implementing non-interactive models is a one-time task that requires continuous auditing.

Figure 3 describes the non-interactive model for data release. The data owners use differential privacy based mechanisms to offer data requester an already sanitized dataset. The dataset is built to provide noisy results for the requester's queries.

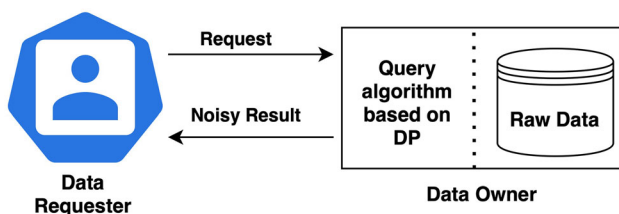


Fig. 2 Interactive model for data release. Provides an interface to the data requesters to make queries and fetch data

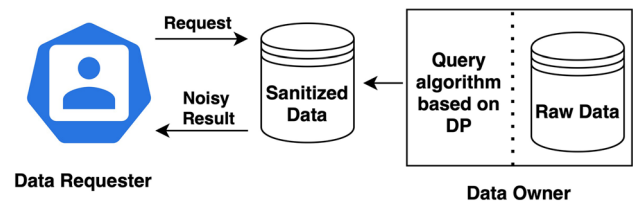


Fig. 3 Non-interactive model for data release. The individuals have to trust the data curator, and they transfer data ownership to the collector once the information is filled, sanitized and passed to a data requester

6 Heterogeneous datasets

6.1 Generating private synthetic data

Generating datasets using affine transformations, augmentation, or deep learning techniques like generative adversarial networks (GANs) is commonplace. The problem with such generation is that the generated dataset looks very similar to the original samples. Imagine a scenario where we want to generate synthetic data that is private at the same time. No individual sample from original samples can be reconstructed using the samples from generated synthetic data. DP-GANs are used with a private training procedure to generate images that are private [15, 16]. The authors also show that the generated data has high utility on a binary classification task.

In [16], the authors have proposed an Information maximizing Differentially Private Generative Adversarial Network (imdpGAN), shown in Fig. 4 taken from [16] (Figure 1). It is a unified framework to simultaneously preserve privacy and learn latent representations. As privacy concerns are rising up there are multiple use cases of our framework. For example, popular face recognition systems (FRS) claim that they store only a representation of users' faces and not the actual image⁶. However, while operating they require a complete face image as input to authenticate an user. Researchers have shown that the facial recognition systems are vulnerable to template reconstruction attack, which might cause harm to privacy of the users [34, 35]. The proposed framework, imdpGAN, can be used to create anonymized face images that are closer to the real face representations by learning meaningful latent codes while generating private faces to preserve user's privacy. The private face representations differ from the ones formed with original faces and hence are shielded from template reconstruction attack.

⁶ Apple tweeted, "Face ID only stores a mathematical representation of your face on iPhone, not a photo." <https://twitter.com/apple/status/1215224753449066497>

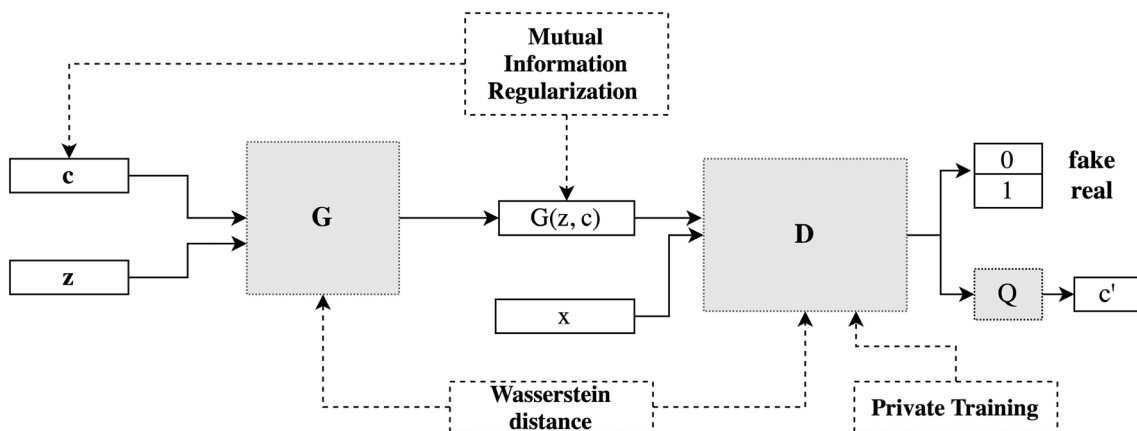


Fig. 4 imdpGAN Architecture: addition of the private training procedure, the mutual information regularization and the Wasserstein distance

6.2 High dimensional data

Due to increased perturbation errors and computational complexity, the trivial data release solutions were ineffective. Most of them requires injecting a prohibitive amount of noise, which renders the published data to be nearly useless [36–39]. For example, a database has 10M tuples, 20 attributes (dimensions), and 10 values per attribute. The full tuple distribution has $10^{20} = 10T$ cells, and most of them have non-zero counts after noise injection. Thus, the average information in each cell can be calculated as $\frac{10M}{10T} = 10^{-6}$. If the average noise is $1/\epsilon = 10$ (for $\epsilon = 0.1$). Obviously, the signal-to-noise ratio is extremely low. To curb this problem, the authors proposed DPPro that used random projections for differentially private data release for high-dimensional data [17, 18]. They prove that DPPro can generate a synthetic dataset with a similar euclidean score (or l_2 norm) between high dimensional vectors while achieving (ϵ, δ) -differential privacy. The utility guarantees depend on the projection dimensions and variance of the added noise. DPPro is proven to be the state-of-the-art model for high-dimensional datasets.

Figure 5 shows the architecture of DPPro. As we can see, the sensitive data is identified and converted into high-dimensional vectors. A low dimensional distribution is created using random projections and optimizing the l_2 -norm followed by the addition of noise for privacy.

6.3 Time series and sequential data

With time-series data, there is always a correlation between timestamps, and therefore, it is hard to preserve the privacy of such datasets. Even when noise is added trivially, the perturbation errors become large enough to compromise the dataset’s utility. The authors propose estimation

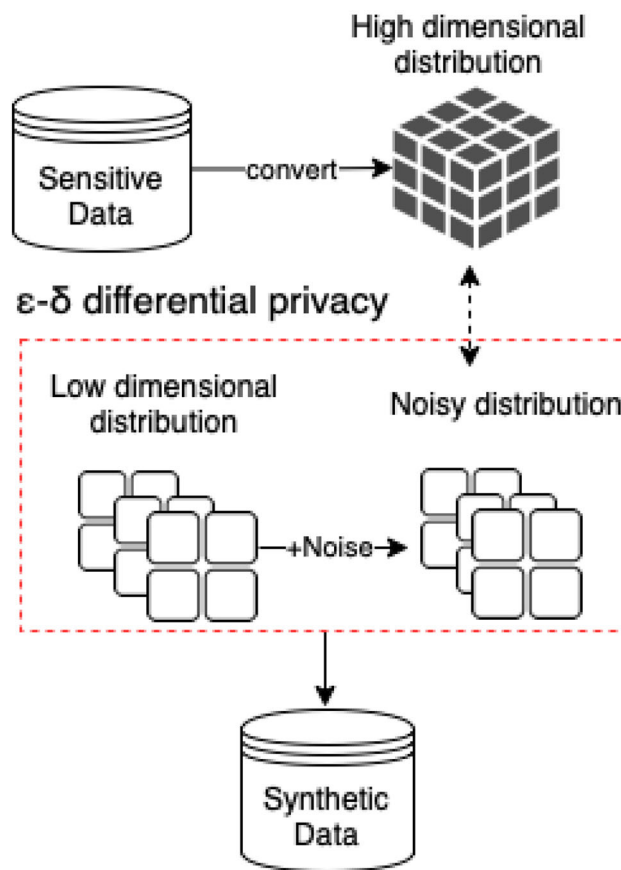


Fig. 5 DPPro: a framework for high dimensional data release. Sensitive data is identified and converted into high-dimensional vectors. A low dimensional distribution is created using random projections and optimizing the l_2 -norm followed by the addition of noise for privacy

algorithms designed to utilize domain knowledge to mitigate the effect of perturbation error [19].

Figure 6 shows a Laplace perturbation with a known mean, and variance is used to transform query responses. Since the noise distribution is known, the query responses

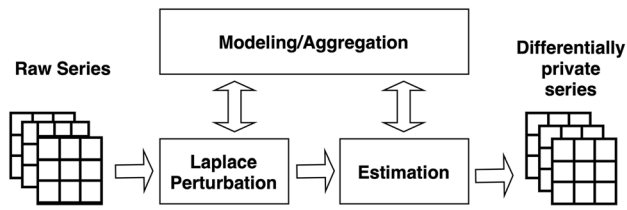


Fig. 6 Framework for time series data. A Laplace perturbation with a known mean, and variance is used to transform query responses

and the noise are optimized so that the generated results do not lose their utility.

6.4 Personalized recommendations

In [20], authors propose a way to make personal recommendations in a private manner. Unlike federated learning [40], the users’ data is first obfuscated and then collected from their devices. The obfuscation serves two purposes: i) gives users more control over their data, and ii) service providers less responsibility for privacy protections. The authors use differential privacy for this obfuscation as it is lightweight and have strong provable privacy guarantees.

Figure 7 shows the architecture for privacy-preserving personalized recommendations. The privacy mechanism (obfuscation of data) is introduced while the collection step. The rest recommendation model remains the same.

6.5 Stream analytics

Many online services collect users’ data continuously for real-time analytics. Most often, the data arrives as streams requiring distributed architectures due to their high volumes. Current architectures are centrally controlled and therefore work on the trust between users’ and the analysts. If sensitive information is removed or noise is added

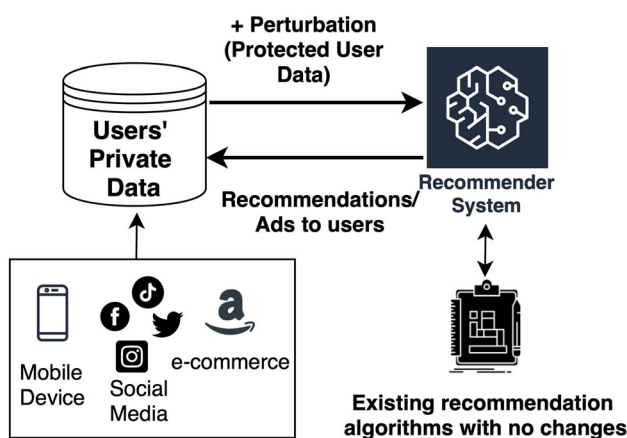


Fig. 7 Privacy-Preserving Personalized Recommendation. A perturbation is added on collected users’ private data and processed into a recommendation pipeline

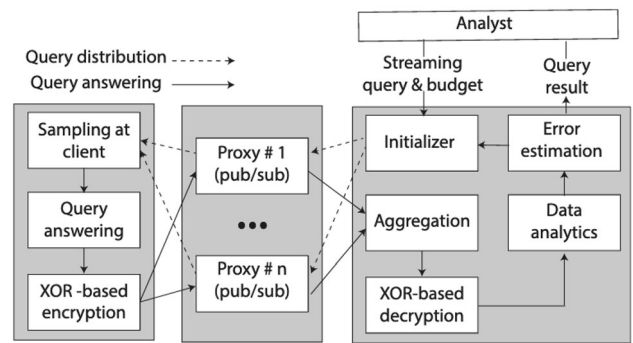


Fig. 8 PrivApprox framework taken from [21](Figure 3). The encrypted responses using XOR based encryption are published on multiple proxies, which are then picked by analysts acting as aggregators to output the final query result

arbitrarily, utility gets effected. The authors proposed PrivApprox for high utility data analytics that provides privacy, utility and low latency for stream data [21, 22].

As shown in Fig. 8, the queries are samples and the responses are encrypted and decrypted using a XOR-based encryption technique. The encrypted response are published on multiple proxies, which are then picked by analysts acting as aggregators to output the final query result.

6.6 Graph mining

Graph embedding maps graphs with nodes and vertices into low dimensional vectors, known as embedding matrix - while preserving the graph structure to reduce the high computation and space complexity. Personal features mapped onto the graph’s embedding matrix can identify an individual, which puts them at risk of privacy leakage. The authors proposed a private perturbed gradient descent (PPGD) for embedding matrix sharing [23].

Figure 9 shows the framework with PPGD as a step to convert data from high dimensional node similarity matrix to a low dimensional graph embedding matrix. Here, a Lipschitz condition based private mechanism is used with matrix factorization for the transformation.

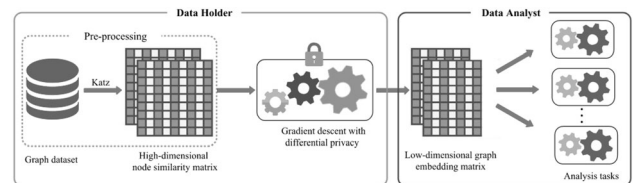


Fig. 9 Private embedding matrix sharing framework to convert data from high dimensional node similarity matrix to a low dimensional private graph embedding matrix [23](Figure 2)

6.7 Statistical computations

While working with sensitive data, we want statistical inferences to take into account the privacy of individual samples and eliminate possibilities of input reconstruction. However, enforcing strict privacy guarantees distort data, and its analysis, thus limiting their analytical ability. To address this issue, the authors propose “integral privacy”, a re-sampling based privacy model. It can be used to compute descriptive analysis without compromising its utility [24].

The objective here is to choose statistical or machine learning model that can be formulated by multiple functions. They will be different combinations of input samples with zero shared records among them. This is will make it hard for the adversary to link an output with an input even if they have access to crucial information. Integral privacy achieves this through re-sampling and discretization of outputs. Suppose the statistical query is mean, integral privacy will select the most recurrent result, which can be generated by unique input samples with no intersection among them.

6.8 Internet of things

More often than not, internet of things data is high dimensional, and most differential privacy based mechanisms have poor utility are not effective. They are computationally expensive because the added noise is proportional to the size of the data domain, and therefore, goes exponential to the dimensionality. To solve this problem, the authors proposed a compressed sensing mechanism (CSM) that uses the compresses sensing framework to provie accurate results to linear queries. [25]

As shown in Fig. 10 a sparse representation is formed from the data followed by compression to reduce the dimensionality of original dataset. The noise is injected on the compressed representation, and then the original representation is reconstructed for query outputs/synthetic dataset. The norm between original and reconstructed data is calcuclated and used as optimization metric.

7 Limitations of differential privacy

7.1 Sharing and collaboration

Differential privacy algorithms work because they add noise, which is a nice way of saying “error”. For some algorithms like computing the mean, the errors can cancel each other out and still lead to accurate results. More complex algorithms are not so lucky. Also, when the data

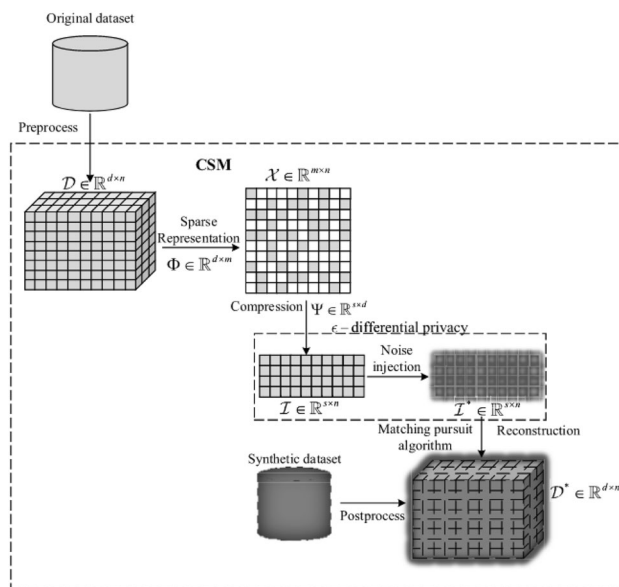


Fig. 10 Compressed Sensing Mechanism (CSM) Framework taken from [25] (Figure 1). A sparse representation is formed from the data followed by compression to reduce the dimensionality of original dataset

sets are small, the effects of the fuzzing can be much more dramatic leading to the potential for big distortions.

7.2 Controlling trade-offs between privacy and accuracy

Epsilon is still just a number that is to be chosen based on experimentation. There is no easy guide and the best practices haven’t evolved yet for choosing the optimal ϵ . Setting the value can be complex, especially when the data sets are less predictable. Algorithms try to suppress the sensitivity of the data defined by how close the data values may be to each other. The ideal noise will blur the distinction among people making it impossible for an attacker to identify one. Sometimes the data cooperates and sometimes it can be hard to find a single good value of epsilon.

7.3 Machine/Deep learning

Noise can have unknown effects. Machine learning algorithms can be black box making the decisions without any explanation. The mystery is compounded when the algorithms are fed fuzzed data because it is often impossible to know just how the changes in the data affected the outcome. Some simple CSM algorithms like finding the mean are easy to control and understand, but not complex and large ML/DL models.

8 Use cases

8.1 Statistical computations

Google has a differential privacy project⁷ to generate ϵ - and (ϵ, δ) -differentially private statistics over datasets. OpenMined has released a python based library, known as Pysyft⁸, used for federated learning and training models using private procedures. A similar organization, OpenDP has released a tool to generate synthetic and private dataset⁹. Aforementioned are the best tools available to use differential privacy for statistical computations.

Example. There are 200 animals in a zoo. Each animal eats carrots everyday, and the number of carrots eaten along with the animal name is used to create a dataset. At the end of each day, the zoo owner asks aggregate question about the number of carrots everyone ate so that they can order carrots for the next day. The animals are scared to share the information, as it can be used against them. The farmer can plan to get rid of the animals who are eating maximum number of carrots to reduce operational costs. To protect their interests, the animals used differential privacy to aggregate their data. The DP implementation protects the animals from getting identified, while accurately giving out the number of carrots needed for the next day for efficient operation of the zoo. The example is borrowed from one of the Google's differential privacy project examples¹⁰. Please follow the footnote link to check out the implementation.

8.2 Machine/Deep learning

Machine/Deep learning is widely used across the globe for different tasks. Any machine learning problem involves collecting a large dataset, followed by training models that can learn patterns within that data. However, these models tend to leak information about the original data in some cases [41], and therefore, training machine learning models is becoming commonplace in sensitive applications. Tensorflow¹¹ and Pytorch¹² are the most widely used machine learning frameworks and both of the them have built-in tools to impose privacy while training.

Example. Consider data from healthcare with images of patients having a benign and a malignant lung cancer. To train a model for a classification task, we will need access

to these images. But, the images under healthcare acts are considered sensitive and cannot be used unless made private. In such cases, we can use machine learning models to generate private images and use them for the classification task. As an example implementation, [16] and [15] used differentially private training procedures to generate synthetic datasets. Similar models can be created using the aforementioned tools. An implementation of such kind is discussed here¹³. Please follow the footnote link to check out the implementation.

9 Conclusion and future works

According to the dictionary, the meaning of the term “statistics” is given as dealing with data that tells the condition of a group or a community. Differential privacy, as we defined it, states: if the presence or absence of an individual sample in a study does not affect the outcome of the study, we can say the outcome is about the group or the community. But if in case, the outcome of the study changes, we can say that the outcome was about the few individuals whose data is included in the study. Therefore, differential privacy has two properties: (i) it is stable to small perturbations in the data, and (ii) it is statistical in that the analysis done tells us about the whole community.

The existence of the first property directly leads to the use of differential privacy into securing machine/deep learning models. Differential privacy protects the models from adversarial attacks that focus on adding small perturbations in the training data [42, 43]. Differential privacy ensures generalization in adaptive data analysis. Adaptive analysis means that the questions asked and hypotheses tested depend on the outcomes of earlier questions. Generalization refers to bringing the outcome of a test on a sample closer the ground truth of the distribution as much as possible. Also, generalization requires the outcome to remain the same no matter how the data is sampled. Answering with differentially private mechanisms ensures privacy and generalizability with high probabilities. Hence, differential private mechanisms of adding controlled noise have promising statistics results and applications.

While using differential privacy and its various methods for data release, the data is stored in one entity. We need to find the answer: if the data is stored at multiple entities, how can it be shared while guaranteeing users' privacy between these entities. Another is the data lineage problem with third-party sharing. When different attributes are used to make the data private and the confidential data is shared with multiple parties, how do we ensure that the cross-

⁷ <https://opensource.google/projects/differential-privacy>

⁸ <https://github.com/OpenMined/PySyft>

⁹ <http://psiprivacy.org/static/about/index.html>

¹⁰ <https://github.com/google/differential-privacy/tree/main/examples/cc>

¹¹ <https://github.com/tensorflow/privacy>

¹² <https://opacus.ai>

¹³ <https://blog.openmined.org/differentially-private-deep-learning-using-opacus-in-20-lines-of-code/>

linking will not happen if both of the datasets reach an adversary? Most of the problems stated above can be resolved if we can infuse differential privacy into a blockchain-based distributed environment instead of a central entity controlling everything [44]. However, we need more work before a successful solution is reached.

Despite its vast applications, differential privacy is not yet applied to multi-modal data. All of the works that we discussed focus on a single mode dataset. Due to the risk of information leakage caused by data correlations, the methods discussed earlier in the paper cannot be used directly with multi-modal data. In an early attempt at multi-modal privacy protection, [45] used an anonymity-based privacy model to protect an integration of transaction and trajectory data. More research is needed in the multi-modal direction to understand its advantages and disadvantages while using differential privacy mechanisms.

References

1. Wang K, Yu PS, Chakraborty S (2004) Bottom-up generalization: a data mining solution to privacy protection. In: Fourth IEEE international conference on data mining (ICDM'04). IEEE, pp 249–256
2. Martínez S, Sánchez D, Valls A, Batet M (2012) Privacy protection of textual attributes through a semantic-based masking method. *Inf Fusion* 13(4):304–314
3. Archana R, Hegadi RS, Manjunath T (2018) A study on big data privacy protection models using data masking methods. *Int J Electr Comput Eng* 8(5):3976
4. Sweeney L (2002) K-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, Oct. [Online]. Available: <http://dx.doi.org/10.1142/S0218488502001648>
5. Machanavajjhala A, Gehrke J, Kifer D, Venkatasubramanian M (2006) L-diversity: privacy beyond k-anonymity. In: 22nd international conference on data engineering (ICDE'06), April 2006, pp 24–24
6. Li N, Li T, Venkatasubramanian S (2007) t-closeness: Privacy beyond k-anonymity and l-diversity. In: 2007 IEEE 23rd International Conference on Data Engineering, April 2007, pp 106–115
7. Narayanan A, Shmatikov V (2008) Robust de-anonymization of large sparse datasets. In: Proceedings of the 2008 IEEE Symposium on Security and Privacy, ser. SP '08. Washington, DC, USA: IEEE Computer Society, pp 111–125. [Online]. Available: <https://doi.org/10.1109/SP.2008.33>
8. Warner SL (1965) Randomized response: a survey technique for eliminating evasive answer bias. *J Am Stat Assoc* 60(309):63–69
9. Homer N, Szelling S, Redman M, Duggan D, Tembe W, Muehling J, Pearson JV, Stephan DA, Nelson SF, Craig DW (2008) Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS Genet* 4(8):e1000167
10. Dinur I, Nissim K (2003) Revealing information while preserving privacy. In: Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp 202–210
11. Narayanan A, Shmatikov V, Robust de-anonymization of large sparse datasets. In: (2008) IEEE Symposium on Security and Privacy (sp 2008). IEEE pp 111–125
12. Lane ND, Xie J, Moscibroda T, Zhao F (2012) On the feasibility of user de-anonymization from shared mobile sensor data. In: Proceedings of the Third International Workshop on Sensing Applications on Mobile Phones, ser. PhoneSense '12. New York, NY, USA: Association for Computing Machinery, [Online]. Available: <https://doi.org/10.1145/2389148.2389151>
13. Dwork C, McSherry F, Nissim K, Smith A (2016) Calibrating noise to sensitivity in private data analysis. *J Privacy Confid* 7(3):17–51
14. Xiong P, Zhu T-Q, Wang X-F (2014) A survey on differential privacy and applications. *Jisuanji Xuebao/Chinese J Comput* 37(1):101–122
15. Xie L, Lin K, Wang S, Wang F, Zhou J (2018) Differentially private generative adversarial network. *CoRR*, [arXiv:1802.06739](https://arxiv.org/abs/1802.06739)
16. Gupta S, Buduru AB, Kumaraguru P (2020) imdpGAN: Generating private and specific data with generative adversarial networks. In: 2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA). IEEE, pp 64–72
17. Xu H, Ding X, Jin H, Yu Q (2021) A multi-dimensional index for privacy-preserving queries in cloud computing. *Concurr Comput Pract Exp* 33(8):e5458
18. Xu C, Ren J, Zhang Y, Qin Z, Ren K (2017) Dppro: Differentially private high-dimensional data release via random projection. *IEEE Trans Inf Forensics Secur* 12(12):3081–3093
19. Fan L, Xiong L, Sunderam V (2013) Differentially private multi-dimensional time series release for traffic monitoring. In: IFIP annual conference on data and applications security and privacy. Springer, pp33–48
20. Shen Y, Jin H (2014) Privacy-preserving personalized recommendation: an instance-based approach via differential privacy. In: 2014 IEEE international conference on data mining. IEEE, pp 540–549
21. Beck M, Bhatotia P, Chen R, Fetzer C, Strufe T et al. (2017) Privapprox: privacy-preserving stream analytics. In: 2017 USENIX Annual Technical Conference USENIX ATC 17, pp 659–672
22. Wang J, Liu C, Fu X, Luo X, Li X (2019) A three-phase approach to differentially private crucial patterns mining over data streams. *Comput Secur* 82:30–48
23. Zhang S, Ni W (2019) Graph embedding matrix sharing with differential privacy. *IEEE Access* 7:89 390–89 399
24. Senavirathne N, Torra V (2019) Integral privacy compliant statistics computation. In: Data privacy management, cryptocurrencies and Blockchain Technology. Springer, pp 22–38
25. Zheng Z, Wang T, Wen J, Mumtaz S, Bashir AK, Chauhdary SH (2019) Differentially private high-dimensional data publication in internet of things. *IEEE Internet Things J* 7(4):2640–2650
26. Sweeney L (2013) Matching known patients to health records in Washington state data
27. Gupta S, Anant A, Vyalla SR, Buduru AB, Kumaraguru P (2020) Ivotedto# igotpwned: Studying voter privacy leaks in Indian Lok Sabha elections on Twitter.(2020)
28. Cohen A, Nikolov A, Schutzman Z, Ullman J (2020) The theory of reconstruction attacks. *DifferentialPrivacy.org*, 10, <https://differentialprivacy.org/reconstruction-theory/>
29. Dwork C (2006) Differential privacy. In: *International Colloquium on Automata, Languages, and Programming*. Springer, pp 1–12
30. Inc A (2022) Differential privacy. [Online]. Available: https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf

31. Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, Zhang L (2016) Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ser. CCS '16. New York, NY, USA: ACM, pp 308–318. [Online]. Available: <http://doi.acm.org/10.1145/2976749.2978318>
32. Shokri R, Shmatikov V (2015) Privacy-preserving deep learning. In: Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security, ser. CCS '15. New York, NY, USA, ACM, pp 1310–1321. [Online]. Available: <http://doi.acm.org/10.1145/2810103.2813687>
33. Phan NH, Wu X, Dou D (2017) Preserving differential privacy in convolutional deep belief networks. pp 1681–1704, 06
34. Mai G, Cao K, Yuen PC, Jain AK (2019) On the reconstruction of face images from deep face templates. *IEEE Trans Pattern Anal Mach Intell* 41(5):1188–1202
35. Ramachandra R, Busch C (2017) Presentation attack detection methods for face recognition systems: a comprehensive survey. *ACM Comput. Surv.*, vol 50, no 1, mar [Online]. Available: <https://doi.org/10.1145/3038924>
36. Chen R, Xiao Q, Zhang Y, Xu J (2015) Differentially private high-dimensional data publication via sampling-based inference. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp 129–138
37. Li H, Xiong L, Jiang X (2014) Differentially private synthesis of multi-dimensional data using copula functions. In: Advances in database technology: proceedings. International conference on extending database technology. vol NIH Public Access, p 475
38. Qardaji W, Yang W, 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. ser. SIGMOD '14. New York, NY, USA: Association for Computing Machinery, pp 1435–1446. [Online]. Available: <https://doi.org/10.1145/2588555.2588575>
39. Zhang J, Cormode G, Procopiuc CM, Srivastava D, Xiao X (2017) Privbayes: private data release via bayesian networks. *ACM Trans Database Syst (TODS)* 42(4):1–41
40. Yang Q, Liu Y, Cheng Y, Kang Y, Chen T, Yu H (2019) Federated learning. *Synth Lect Artif Intell Mach Learn* 13(3):1–207
41. Hitaj B, Ateniese G, Perez-Cruz F (2017) Deep models under the gan: information leakage from collaborative deep learning. In: Proceedings of the 2017 ACM sigsac conference on computer and communications security, pp 603–618
42. Lecuyer M, Atlidakis V, Geambasu R, Hsu D, Jana S (2019) Certified robustness to adversarial examples with differential privacy. In: (2019) IEEE symposium on security and privacy (SP). IEEE, pp 656–672
43. Giraldo J, Cardenas A, Kantarcioglu M, Katz J (2020) Adversarial classification under differential privacy. In: Network and Distributed Systems Security (NDSS) Symposium 2020
44. Liu L, Piao C, Jiang X, Zheng L (2018) Research on governmental data sharing based on local differential privacy approach. In: 2018 IEEE 15th international conference on e-business engineering (ICEBE), pp 39–45
45. Sui P, Li X (2017) A privacy-preserving approach for multimodal transaction data integrated analysis. *Neurocomputing*, vol 253, pp 56–64, learning Multimodal Data