

Tabular Data Synthesis with Differential Privacy: A Survey

MENGMENG YANG, CSIRO, Data61, Australia

CHI-HUNG CHI, KWOK-YAN LAM, Nanyang Technological University, Singapore

JIE FENG*, Xidian University, China

TAOLIN GUO, Chongqing Normal University, China

WEI NI, CSIRO, Data61, Australia

Data sharing is a prerequisite for collaborative innovation, enabling organizations to leverage diverse datasets for deeper insights. In real-world applications like FinTech and Smart Manufacturing, transactional data, often in tabular form, are generated and analyzed for insight generation. However, such datasets typically contain sensitive personal/business information, raising privacy concerns and regulatory risks. Data synthesis tackles this by generating artificial datasets that preserve the statistical characteristics of real data, removing direct links to individuals. However, attackers can still infer sensitive information using background knowledge. Differential privacy offers a solution by providing provable and quantifiable privacy protection. Consequently, differentially private data synthesis has emerged as a promising approach to privacy-aware data sharing. This paper provides a comprehensive overview of existing differentially private tabular data synthesis methods, highlighting the unique challenges of each generation model for generating tabular data under differential privacy constraints. We classify the methods into statistical and deep learning-based approaches based on their generation models, discussing them in both centralized and distributed environments. We evaluate and compare those methods within each category, highlighting their strengths and weaknesses in terms of utility, privacy, and computational complexity. Additionally, we present and discuss various evaluation methods for assessing the quality of the synthesized data, identify research gaps in the field and directions for future research.

CCS Concepts: • **Do Not Use This Code** → **Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

Additional Key Words and Phrases: Tabular Data synthesis, Differential privacy, Statistical models, Deep learning generation models

ACM Reference Format:

Mengmeng Yang, Chi-Hung Chi, Kwok-Yan Lam, Jie Feng, Taolin Guo, and Wei Ni. 2018. Tabular Data Synthesis with Differential Privacy: A Survey. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 35 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

Data sharing is essential as it drives innovative collaboration and enables informed decision-making across various domains. Numerous public data-sharing platforms, including Kaggle [4], Data.gov [3], and the UCI repository [45],

*Participated in the work during the visit to NTU.

Authors' Contact Information: Mengmeng Yang, CSIRO, Data61, Melbourne, Australia, mengmeng.yang@data61.csiro.au; Chi-Hung Chi, Kwok-Yan Lam, Nanyang Technological University, Singapore, chihung.chi@ntu.edu.sg; Jie Feng, Xidian University, Xi'an, China, fengjie@xidian.edu.cn; Taolin Guo, Chongqing Normal University, Chongqing, China, tguo@cqnu.edu.cn; Wei Ni, CSIRO, Data61, Sydney, Australia, wei.ni@data61.csiro.au.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

Manuscript submitted to ACM

offer access to extensive datasets, with the primary goal of facilitating knowledge discovery and advancement. In most applications, such as FinTech and Smart Manufacturing, these datasets are represented in tabular form, given their structured nature and widespread applicability across different fields. However, it is important to note that these datasets often contain sensitive personal/business data, which can raise significant privacy concerns. In addition, due to evolving privacy regulations, exemplified by recent legislation like the AI Act [2], there is a heightened need for innovative methods for data sharing that protect individual privacy while enabling meaningful data analysis.

Data synthesis has been attracting growing attention due to its unique ability to generate synthetic data based on statistical information without being linked to specific individuals or identities. However, it is important to note that while synthetic data offers privacy protection, several studies [43, 58] have shown that the attacker can still potentially infer sensitive information about users. For example, Jordon et al. [76] show that “*Synthetic data can leak information about the data it was derived from and is vulnerable to privacy attacks.*” Moreover, Stadler et al. [115] have demonstrated that generative models trained without privacy safeguards offer limited defence against inference attacks when compared to the alternative of directly sharing the original data.

A cutting-edge solution involves integrating provable privacy measures, such as differential privacy (DP), into the synthetic data generation process. Differential privacy aims to ensure that the information derived from the released synthetic data remains nearly identical, whether or not specific individuals were part of the original datasets, thus effectively preventing the inference of personal information. Importantly, it does not rely on assumptions about the capabilities of potential attackers, providing robust privacy protection even in the presence of adversaries with significant background knowledge and resources [144]. The United States National Institute of Standards and Technology (NIST) has been instrumental in championing data sharing and privacy protection. In 2018, NIST organized the “Differential Privacy Synthetic Data Challenge” [1], a competition dedicated to advancing the field of differential privacy for generating synthetic data that retains the statistical characteristics of actual data while safeguarding individual privacy. The challenge highlighted the growing importance of balancing the need to share valuable insights with the necessity of protecting personal information, making differentially private data synthesis a promising research focus.

In this paper, we present a comprehensive review of existing differential private tabular data synthesis methods. The generation of differentially private synthetic tabular data primarily falls into two key categories: statistical methods and deep learning-based methods. We delve into both approaches, analyzing their strengths and limitations under both centralized and distributed settings. Furthermore, we offer insights into the unique challenges and considerations that arise in each context.

Table 1. Comparison with existing surveys

Paper	Year	Consideration of tabular data synthesis with DP	Centralized data synthesis		Distributed data synthesis	Discussion on Evaluation	
			S-M	DL-M		Fidelity/Utility	Privacy
[30]	2020	✓	✓				
[29]	2021			✓		✓	
[53]	2022			✓		✓	
[59]	2022	✓	✓	✓		✓	
[136]	2022		✓	✓		✓	✓
[91]	2023			✓		✓	
[67]	2023	✓		✓		✓	
[70]	2024	✓	✓	✓			
[22]	2024		✓	✓			
Ours	-	✓	✓	✓	✓	✓	✓

Differences between this survey and others. Currently, several synthesis surveys have been published, as shown in Table 1. Bourrou et al. [29] reviewed several popular GAN-based models for tabular intrusion detection system data synthesis and experimentally evaluated their performance. However, this review exclusively focused on GAN-based models, without considering other methods. Figueira and Vaz et al. [53] slightly extended the scope but still concentrated on GAN-based models for data synthesis. Xing et al. [136] provided a broader review, covering data synthesis methods for non-imaging medical datasets, including both tabular and sequential data. Their discussion included statistical and deep learning-based methods, as well as evaluation metrics. Lu et al. [91] explored machine learning-based approaches for data synthesis and applications of synthetic data generation, addressing privacy and fairness concerns related to synthetic data.

However, all the aforementioned papers discuss pure synthetic data generation methods, and none of them consider the methods with differential privacy protection. Bowen and Liu [30] conducted an experimental study on differentially private data synthesis methods, focusing solely on statistical approaches. Hassan et al. [67] explored the intersection of synthetic data and differential privacy, with a primary focus on deep generative models. Bauer et al. [22] conducted a comprehensive study of various model types suitable for synthetic data generation, including methods that incorporate differential privacy. Ghatak and Sakurai [59] considered the methods with differential privacy protection, but exclusively discussed data synthesis methods that emerged victorious in the NIST 2018 challenge. Hu et al. [70] provided a review of differentially private data synthesis, including tabular data. However, their approach was more of a simple summary of existing methods rather than an in-depth analysis and discussion.

Furthermore, all existing surveys focus on centralized data synthesis methods and do not address distributed data synthesis. In our paper, we target tabular data synthesis methods with differential privacy protection under both centralized and distributed settings. Additionally, we summarize and discuss various evaluation methods for the generated synthetic data, focusing on fidelity, utility, and privacy.

Contributions of this survey. This survey provides a comprehensive review of differential private data synthesis methods, focusing on tabular data. We consider two application scenarios: centralized data synthesis, where the data curator holds all users' datasets and aims to generate synthetic datasets for data analytics or sharing purposes, and distributed data synthesis, where data owners retain their data locally and collaborate with other parties for joint data synthesis. Our contributions are summarized as follows:

- We provide a thorough and comprehensive overview of existing methods for differentially private tabular data synthesis, along with the evaluation techniques used to assess their performance and effectiveness.
- Based on the synthetic data generation models, we categorize the primary approaches for data synthesis into two key research directions: statistical-based methods and deep learning-based methods, both applicable under two main scenarios: centralized and distributed data synthesis.
- We provide an in-depth review and analysis of existing methods for generating synthetic data, highlighting strengths and weaknesses in capturing attribute dependencies, modeling the distribution of attributes, computational complexity, and the noise scales introduced during the model learning process, etc.
- By analyzing the state-of-the-art in the field, we discuss the research gaps and identify several promising future research directions to address the emerging challenges and advance the domain of private tabular data synthesis.

In this survey, we present the material in a tutorial manner, providing a clear introduction, comprehensive discussion, and valuable insights into the topics and methods. We aim to make the content accessible and informative for readers who are new to the subject as well as those looking to deepen their understanding.

Roadmap. The rest of the paper is organized as follows: Section 2 provides background knowledge on tabular data synthesis and differential privacy. Section 3 and Section 4 discuss centralized data synthesis methods with differential privacy protection and distributed data synthesis methods with differential privacy protection, respectively. Section 5 introduces the synthetic data evaluation metrics. The research gaps and promising research directions are identified in Section 6, and the survey is concluded in Section 7.

2 BACKGROUND KNOWLEDGE

2.1 Tabular data synthesis

2.1.1 Concepts. This Section briefly presents the concept of tabular data synthesis.

Tabular data. Tabular data is data organized in a structured format with rows and columns, similar to a table. Each row represents a specific record or instance, such as a customer, transaction, or observation, and each column corresponds to a variable or attribute, such as name, age, or product price.

Data synthesis. Data synthesis is the process of creating artificial datasets that replicate the structure, statistical properties, and relationships of real-world tabular data. The primary goal is to generate synthetic data that preserves the essential characteristics of the original data while ensuring privacy and enabling safe data sharing. This process is particularly valuable for scenarios where privacy concerns or data scarcity limit the use of real datasets.

Centralized data synthesis. Centralized data synthesis refers to the process of generating synthetic datasets where all the original data is collected, stored, and processed in a single, centralized location. A data curator or central authority typically holds the entire dataset and applies data synthesis techniques to produce a synthetic version.

Distributed data synthesis. Distributed data synthesis involves generating synthetic datasets from data that is stored and processed across multiple locations or nodes. Each data owner retains control over their local data and collaborates with other parties to jointly create synthetic data without centralizing the original datasets.

2.1.2 Challenges. Tabular data synthesis poses several challenges, particularly due to the complexity and diversity of tabular datasets compared to other data types like images or text. Some of the key challenges include:

Data heterogeneity. Tabular data often includes a mix of different data types (e.g., categorical, numerical and ordinal). Modelling the relationships between these different types of features is challenging.

Data distribution complexity. First, many features in tabular data may not follow a simple distribution, and some may have multiple peaks (multi-modality), making it hard to model these distributions accurately. Additionally, real-world tabular data is often imbalanced, with skewed distributions in certain classes, it is challenging to represent the imbalance.

Feature dependencies. The relationships between features can be highly complex (e.g., non-linear correlations). Capturing these relationships accurately in the synthetic data is challenging.

Generating synthetic tabular data with differential privacy introduces additional challenges beyond those involved in general tabular data synthesis. These include the high sensitivity and dimensionality of the tabular data, which require significant noise to ensure privacy, and the cumulative noise can substantially degrade the overall quality of the synthetic data. Therefore, It is challenging to balance the utility and privacy of the synthetic data.

2.1.3 Privacy disclosure in synthetic data. One of the primary purposes of generating synthetic data is to enhance data privacy, facilitating data sharing without compromising users' sensitive information. However, research [11, 127] have shown that sensitive information can still be disclosed if an attacker possesses some background knowledge about the victim. Several types of attacks can be performed on synthetic data, including the following:

Re-identification attack. Re-identification [75] attacks typically exploit auxiliary information or background knowledge that an attacker possesses about the individuals in the dataset. By correlating this additional information with the synthetic data, the attacker can identify specific individuals and extract sensitive information.

Inference attack. An inference attack occurs when an attacker deduces sensitive information about individuals from synthetic data by exploiting statistical properties, patterns, or background knowledge. This type of attack does not necessarily re-identify individuals but can still extract confidential information. It includes membership [153] and attributes [13] inference and correlation exploitation.

2.2 Differential privacy

2.2.1 Definition of differential privacy. Differential privacy, introduced by Dwork et al. [47] in 2006, is a provable privacy concept. It ensures that changing one person's data does not have a big effect on the result, making it hard to tell if that person was included or not, all while still getting useful insights from the data. In differential privacy, ϵ plays a crucial role as a privacy parameter that controls the trade-off between privacy and accuracy. It sets the upper limit on how much the algorithm's output can differ when one individual's data is changed. A smaller ϵ value means stronger privacy protection and also tends to introduce more noise into the data, potentially reducing the accuracy of the results. In contrast, a larger ϵ provides more accurate results but weaker privacy guarantees. (ϵ, δ) -differential privacy is also known as approximate differential privacy, which is an extension of the standard differential privacy framework. It introduces an additional parameter, δ , which allows for a small probability of a stronger privacy loss than ϵ would permit. The pure ϵ -differential privacy is a special case of the approximate differential privacy when $\delta = 0$.

2.2.2 Differential privacy mechanisms. Differential privacy is achieved by introducing randomness into statistical computations. Commonly used mechanisms include the Laplace and Gaussian mechanisms for numerical data and the Exponential mechanism for categorical data.

Laplace Mechanism. The Laplace Mechanism is one of the most widely used methods in differential privacy. It protects privacy by adding noise to the output of a function, and this noise is drawn from the Laplace distribution. The scale of noise is determined by the sensitivity of the function, which measures how much the function's output could change by altering a single individual's data and the privacy parameter ϵ . The Laplace mechanism preserves ϵ -differential privacy.

Gaussian Mechanism. The Gaussian Mechanism works similarly to the Laplace Mechanism but adds noise drawn from the Gaussian distribution instead of the Laplace distribution. The noise magnitude is determined by both the sensitivity of the function and the privacy parameters, ϵ and δ . The Gaussian mechanism satisfies (ϵ, δ) -differential privacy.

Exponential Mechanism. Unlike the Laplace or Gaussian mechanisms, the Exponential Mechanism introduces noise to the selection process by assigning a probability to each possible output based on a utility function that measures how desirable each option is. The probability of selecting an output increases exponentially with its utility value, ensuring that the most useful outputs are more likely to be chosen, but with a privacy-preserving layer of randomness. The exponential mechanism preserves ϵ -differential privacy.

2.2.3 Composition properties. Differential privacy has several composition properties that facilitate the analysis of more complex privacy-preserving algorithms.

Parallel Composition. The Parallel Composition Theorem [48] states that if a method includes m independent randomized functions, each providing its own ϵ -differential privacy guarantee, and if each function operates on a distinct portion of the dataset, then the overall privacy guarantee for the entire method is determined by the function

with the highest ϵ . In other words, the privacy level of the combined method is as strong as the function with the weakest privacy protection (the largest ϵ).

Sequential Composition. The Sequential Composition Theorem [48] states that when multiple differentially private mechanisms are applied sequentially to the same dataset, the overall privacy loss accumulates. Specifically, if m independent functions each provide ϵ -differential privacy guarantees and are applied to the same data, then the combined privacy guarantee is the sum of the individual privacy guarantees. This means that the total privacy loss increases with each additional query or operation on the same dataset, as the more queries are made, the more information about the dataset could potentially be revealed.

Advanced Composition. The Advanced Composition Theorem [48] provides a refined privacy guarantee when applying differentially private mechanisms multiple times. In the context of m -fold adaptive composition, where multiple queries are applied adaptively to the same dataset, the theorem shows that the privacy loss grows slower than the basic composition rule suggests. Specifically, for (ϵ, δ) -differentially private mechanisms, the total privacy guarantee becomes $(\epsilon', m\delta + \delta')$ -differential privacy, where $\epsilon' = \epsilon\sqrt{2m \log(1/\delta')} + m\epsilon(e^\epsilon - 1)$. This results in a tighter bound on the cumulative privacy loss, offering stronger privacy protection compared to the simple summing of the privacy parameters, especially when the number of queries m is large.

Moments Accountant [5]. The Moments Accountant method is a powerful technique used to track and control cumulative privacy loss when applying differential privacy mechanisms multiple times, particularly in scenarios like deep learning. It works by measuring the privacy loss at each step. The approach extends beyond considering just the expectation, using higher-order moments to bound the tail of the privacy loss variable. For each λ -th moment, the Moments Accountant provides a tighter bound on the cumulative privacy loss compared to traditional composition rules.

2.2.4 Relaxations of differential privacy. Strict differential privacy often requires adding significant noise to the data or query results, which can substantially degrade the utility of the data. Relaxations, such as (ϵ, δ) -differential privacy, allow for a controlled trade-off between privacy and utility. In addition to (ϵ, δ) -differential privacy, several relaxations of differential privacy have been proposed, offering greater flexibility in handling data complexities and providing tighter bounds on privacy loss. In this section, we present two commonly used definitions in the literature, Rényi differential privacy and zero-concentrated differential privacy.

Definition 1 ((α, ϵ) -RDP [99]). A randomized mechanism $\mathcal{M} : \mathbb{D} \rightarrow \mathbb{R}$ is said to have ϵ -Rényi differential privacy of order α , or (α, ϵ) -RDP for short, if for any adjacent $D, D' \in \mathbb{D}$ it holds that

$$D_\alpha(\mathcal{M}(D) || \mathcal{M}(D')) \leq \epsilon,$$

where $D_\alpha(\mathcal{M}(D) || \mathcal{M}(D'))$ is the Rényi Divergence of order $\alpha > 1$, specifically,

$$D_\alpha(\mathcal{M}(D) || \mathcal{M}(D')) = \frac{1}{\alpha - 1} \log E_{x \sim \mathcal{M}(D)} \left[\left(\frac{\Pr[\mathcal{M}(D) = x]}{\Pr[\mathcal{M}(D') = x]} \right)^{\alpha - 1} \right] \quad (1)$$

RDP provides a flexible and refined method for measuring privacy loss using Rényi divergence. This approach offers a more accurate measure of cumulative privacy loss and tighter bounds when multiple queries are composed. By parameterizing privacy loss with Rényi divergence, RDP enables a smoother and more controlled tradeoff between privacy and utility, making it easier to achieve an optimal balance [78]. It simplifies the accumulation of privacy loss, reducing the complexity of maintaining privacy guarantees throughout multiple stages of data analysis.

Definition 2 (Zero-Concentrated Differential Privacy (zCDP) [31]). A randomized mechanism $\mathcal{M} : \mathcal{X}^n \rightarrow \mathcal{Y}$ is ρ -zero-concentrated differentially private if, for all $x, x' \in \mathcal{X}^n$ differing on a single entry and all $\alpha \in (1, \infty)$,

$$D_\alpha(M(x)||M(x')) \leq \rho\alpha$$

where $D_\alpha(M(x)||M(x'))$ is the α -Rényi divergence between the distribution of $M(x)$ and $M(x')$.

zCDP uses concentration inequalities to provide a more refined measure of privacy loss. This allows for tighter control over the distribution of privacy loss, leading to more efficient privacy guarantees. One of the major advantages of zCDP is its simpler and more efficient composition properties. zCDP allows for the straightforward addition of privacy loss terms when composing multiple queries.

3 CENTRALIZED DATA SYNTHESIS WITH DP

Centralized data synthesis involves generating synthetic data from a centralized server or database, where all original data is collected and stored in a single location. In this setup, the data curator is trusted and aims to release differentially private synthetic data that prevents sensitive information from being disclosed to third parties.

3.1 Statistical method

Statistical methods emphasize maintaining the data distributions by modelling the joint distribution of attributes and subsequently generating samples from this model. It ensures that the synthetic data maintains identical statistical characteristics as those observed in the original dataset. We categorize the methods into several groups based on the statistical models used to generate the synthetic data. Table. 2 summarizes the proposed methods.

3.1.1 Copula. A copula is a statistical concept used to model and analyze the dependence structure between random variables. Copulas works by providing a mathematical framework to separate the modelling of the individual marginal distributions of random variables from the modelling of their joint dependence structure. Sklar's Theorem [114], formulated by Henry Sklar in 1959, is a fundamental theorem in copula theory. It states that any joint cumulative distribution function (CDF) of multiple random variables can be expressed in terms of their individual marginal CDFs and a copula function. Let $F(x_1), F(x_2), \dots, F(x_n)$ be the marginal CDFs of n random variables x_1, x_2, \dots, x_n , and $C(u_1, u_2, \dots, u_n)$ be a copula function of n variables. Then, the joint CDF $F(x_1, x_2, \dots, x_n)$ of these random variables can be represented as:

$$F(x_1, x_2, \dots, x_n) = C(F(x_1), F(x_2), \dots, F(x_n)) \quad (2)$$

In this equation, $C(u_1, u_2, \dots, u_n)$ represents the joint distribution of the random variables with uniform marginal distributions. The general process of using copulas is to first transform your data into uniform marginal distributions, often through the use of CDFs of the individual variables. These uniform variables are then subjected to a copula function that captures how the variables' joint probabilities are related. Choosing a suitable copula function is a critical step. Copulas come in various families, including Gaussian, Clayton, Gumbel, and Frank, each tailored to different types of dependence structures.

Selecting the appropriate copula family can be challenging. Li et al. [82] and Asghar et al. [17] employed a Gaussian copula to model the joint distribution of the data, drawing inspiration from the common observation [103] that many real-world high-dimensional datasets often exhibit Gaussian dependence structures. The main advantage of using a Gaussian copula lies in its efficiency, with a run-time that scales quadratically with the number of attributes. However, it assumes a linear relationship between variables [133]. This can be problematic when modelling financial assets or other

variables that do not have a truly Gaussian distribution or exhibit non-linear dependencies. To capture more complex dependence structures. Gambs et al. [56] adopted vine copulas, which is a specific type of copula that decompose the multivariate functions of the popular into multiple ‘couples’ of the copulas. The computational burden increases significantly in parallel.

Estimating copula functions under differential privacy is a straightforward process that involves computing the DP marginals and the DP correlation matrix. Various well-established techniques [7, 134, 139] are available for estimating DP marginal histograms, typically achieved by introducing Laplace or Gaussian noise to the histogram statistics. Since the introduction of Differential Privacy noise can be applied to individual attributes, confining its magnitude to the dimension of every single attribute, which often proves considerably smaller than when addressing joint distributions. However, when dealing with the DP correlation matrix, a more tailored approach is necessary due to the high global sensitivity exhibited by most correlation matrices, leading to excessive noise.

Lessons learned and discussion. Copula-based modelling is primarily designed for continuous variables, but existing works [17, 56, 82] attempt to extend its applicability to discrete data. However, it is important to note that this extension typically applies to ordinal data with a sufficiently large domain. Even in these cases, the discrete data may need to be approximated or transformed into continuous data for accurate modelling.

3.1.2 Probabilistic graphical models. Probabilistic graphical models (PGMs) are a class of statistical models that use graphical representations to express and manipulate the joint probability distributions over a set of random variables. There are two main types of PGMs: Bayesian networks and Markov networks (also known as Markov random fields). Besides, Junction Trees play a crucial role in probabilistic inference within PGMs, primarily due to their efficiency in computing marginal probabilities, making them indispensable for handling complex probability distributions. Therefore, we categorize the methods into three main groups: methods based on Bayesian networks, approaches centered around Markov networks, and techniques leveraging Junction Trees.

Bayesian Network. A Bayesian network utilizes a directed acyclic graph-based structure, where nodes serve as representations for random variables or events, and the directed edges convey probabilistic relationships among these variables [117]. The joint distribution within the dataset can be estimated through the conditional distributions of attribute-parent pairs, as demonstrated below.

$$Pr[X_1, X_2, \dots, X_d] = Pr[X_1]Pr[X_2|X_1] \dots Pr[X_d|X_1, \dots, X_{d-1}], \quad (3)$$

where X_i represents the attribute of the dataset and d is the number of attributes.

The procedure for constructing a Bayesian Network and estimating the joint distribution involves several key steps. Initially, the selection of attribute-parent pairs is essential. Once these pairs are identified, the subsequent step is to create a collection of conditional distributions for these selected attribute-parent pairs. To enhance privacy and confidentiality, it is crucial to incorporate differential privacy noise into all these computational steps. The synthetic data can then be sampled from the approximated distribution.

Attribute-parent pair selection. The choice of attribute-parent pair plays a critical role in shaping the estimated joint distribution, aiming to approximate the dataset distribution closely. PrivBayes [147] is the representative work. It employs the KL-divergence to assess the distance between two distributions. The smallest distance is achieved by maximizing the mutual information between the attribute denoted as X_i and its corresponding parent set Φ . To accomplish this, they employ a greedy approach to select attribute-parent pairs with maximal mutual information,

Table 2. Statistical methods

Method	Reference	Marginal selection		Noise addition		Synthetic data generation		Synthetic data evaluation	
		Selection method	Dependence evaluation	Privacy mechanism	Privacy accountant	Tool	method	Utility evaluation	Privacy evaluation
Copula	DPCopula [82]	1-m	GCF	L	CP	Copula	Sampling	RQ	N
	COPULASHIRLEY [56]	1-m	GCF	L	CP	Copula	Sampling	KSD,MCD Classification Regression	Y
	DPSynthesizer [83]	1-m	GCF	L	CP	Copula	Sampling	RQ	N
PGM	PrivBayes [147]	BN + IG	MI	L	CP	BN	Sampling	k-m DWpre	N
	PrivBayes improved [21]	BN + IG	ID	AG	-	BN	Sampling	Clustering Classification Regression	N
	FAPrivBayes [92]	BN + adaptive	MI	L	CP	JT	Sampling	k-m Classification	N
	PrivMRF [32]	MRF + IG	ID	AG	-	MRF + JT	Sampling	k-m Classification	N
	Private-PGM [98]	-	-	L	CP	MRF	Sampling	k-m	N
	MST [96]	MST+IG	MI	G	RDP	Private-PGM	Sampling	k-m, RQ	N
	JTree [37]	SVT + Opt	MI	L	CP	JT	Sampling	k-m Classification	N
Query	MWEM [65]	QS	-	E+L	CP	JD	Sampling	k-m, RQ	N
	DualQuery [55]	QS	-	E+L	CP	JD	Sampling	k-m	N
	DQRS [129]	QS	-	E+L	CP	JD	Sampling	k-m	N
	FEM/sepFEM [129]	QS	-	E+L	CP	JD	Sampling	k-m	N
	AIM [97]	QS + adaptive	-	E+G	zCDP	Private-PGM	Sampling	k-m	N
	PEP [87]	QS	-	E+G	zCDP	JD	Sampling	k-m	N
	RAP [19]	QS	-	G	zCDP	-	Opt + RR	k-m	N
	RAP++ [128]	QS	-	G	zCDP	-	Opt	k-m, LQ Regression	N
	PRIVATE-GSD [87]	QS	-	E+G	zCDP	-	Opt	k-m, LQ Regression	N
	DPPro [138]	-	-	G	CP	-	Projection	LQ Classification	N
Others	PrivSyn [152]	Opt + Greedy	ID	G	zCDP	-	GUM	k-m,RQ Classification	N

1-m: 1-way marginal; BN: Bayesian Network; JD: Joint distribution; CP: composition property; L: Laplace mechanism; G: Gaussian mechanism; AG: Analytic Gaussian Mechanism [20]; E: Exponential mechanism; IG: Information gain; QS: Query set; ME: Maximum entropy; DWpre: Dimension-wise prediction; k-m: k-way marginal; Opt: Optimization; zCDP: Zero-Concentrated differential privacy; GUM: Gradually update method; RQ: Range query; QS: Query set; GCF: Gaussian Copula Function; KSD: Kolmogorov-Smirnov distance; MCD: Mean Correlation Delta; SVT: Sparse Vector Technology; RR: Randomized Rounding; LQ: Linear query Identity attack

thereby refining the model’s fidelity to the original data distribution. The Exponential mechanism is employed to select the attribute-parent pair to achieve differential privacy in this process. In this context, the Exponential mechanism’s score function is defined as the mutual information. It is worth emphasizing that mutual information exhibits high sensitivity, which results in the introduction of significant noise during the privacy-preserving operation [147]. To address this problem, in their subsequent research [21], they introduce a solution based on utilizing a metric that quantifies the divergence between the joint distribution of attribute pairs, denoted as $P(A_i, A_j)$, and their product distribution, expressed as $P(A_i) \otimes P(A_j)$. It offers a sensitivity value of 2, which is much smaller compared with its range n . In addition, Ma et al. [92] have introduced an approach to enhance the efficiency of network construction. Their method involves reducing the scale of candidate parent node sets. To achieve this, they assess the significance of each node by calculating the sum of its score function in relation to other nodes. Nodes with higher summed values exert greater influence on the network and are added earlier, up to a predefined threshold. This approach does come at the cost of consuming an additional privacy budget to perturb the importance of nodes. However, it proves effective in enhancing computational efficiency, particularly for datasets with a growing number of attributes when employing the exponential mechanism for attribute-parent pair selection.

Conditional distribution generation. The conditional distribution can be derived directly from the joint distribution. Once we have identified the attribute-parent pair, denoted as X_i and Φ_i , we can obtain the joint distribution, represented as $Pr[X_i, \Phi_i]$. A simple approach to introduce differential privacy is to inject Gaussian or Laplace noise into this distribution. However, Eq. 3 highlights the need for a set of high-dimensional marginal distributions. To mitigate dimensionality issues, Zhang et al. [147] introduced the concept of a k -degree Bayesian Network, where each attribute is constrained to have at most k parents. This reduces the scale of noise added to the dataset but may capture fewer correlations between attributes. It is important to acknowledge the practical trade-off involved in making this choice. *Synthetic data generation.* Once the conditional distribution is obtained, we have two options for generating synthetic data. First, we can directly sample synthetic data points from the distribution by computing the probabilities for each element within the variable’s domain. Alternatively, we can opt for a sequential approach where attributes are sampled one by one in accordance with Eq. 3, with each attribute being sampled based on its conditional distribution rather than the joint distribution. The sequential approach allows for a more efficient sampling process. To ensure the generation of tuples with low, yet non-zero frequencies, Bao et al. [21] introduced an alternative sequential approach. In this method, they convert the marginal distributions into histograms by scaling the probabilities based on the number of tuples to be generated. Subsequently, they create tuples one at a time, continuously adding to the histogram until they achieve the desired number of tuples.

Lessons learned and discussion. Bayesian networks have the flexibility to model both discrete and continuous variables, making them suitable for a wide range of data types. However, they heavily rely on assumptions about the conditional dependencies between variables [80]. If these assumptions are incorrect or incomplete, the synthetic data generated by the network may not accurately reflect the true data distribution. In addition, when estimating the joint distribution, there is a constraint placed on the number of marginals used, typically limited to a maximum of d marginals, where d represents the number of attributes. Consequently, capturing all important dependencies among the attributes becomes a challenging task. Furthermore, constructing Bayesian networks can be quite challenging, as it involves navigating a vast search space of potential network structures. Additionally, conducting inference in large Bayesian networks can be computationally expensive, posing challenges in efficiently generating synthetic data, especially for datasets with numerous variables.

Markov network. Markov networks, also named Markov random fields, are one of the most widely used graphical models. In Markov networks, nodes represent random variables, while edges capture dependencies between them. Unlike Bayesian networks, Markov networks use undirected edges, making them well-suited for scenarios where relationships between variables are not easily represented by a causal hierarchy. The key feature of the Markov network is that it uses potential functions, denoted as $\phi_c(X_c) = e^{\theta(X_c)}$, to describe the relationships between nodes (random variables) in the cliques (a subset of nodes fully connected) of the graphical model. θ is the parameter corresponding to X_c . The joint probability distribution over all variables is defined as a product of these potential functions:

$$Pr[X_1, X_2, \dots, X_d] = \frac{1}{Z} \prod_{c \in C} \phi_c(X_c) \quad (4)$$

where C is the set of all cliques and Z is the normalization factor.

Several key steps are involved in constructing a Markov network. Initially, a dependency graph is created to represent the interactions between variables. The noisy marginals are generated by adding Laplace or Gaussian noise to the marginal statistics. Parameters for the potential functions are then determined using statistical inference techniques, such as maximum likelihood estimation. The joint distribution is then estimated as described in Eq. 4. Finally, the synthetic data can be generated using the joint distribution information.

Dependency graph construction. Constructing the dependency graph plays a crucial role in capturing attribute correlations. The general idea of constructing the dependency graph is to insert the edges to the graph when two attributes exhibit a high degree of correlation as measured by certain metrics, such as mutual information. Researchers often focus on modelling low-order correlations to mitigate the challenges posed by high-dimensional data and ensure differential privacy. Chen et al. [37] assessed the mutual information between pairs of attributes and added an edge to the graph if the noisy mutual information is larger than a threshold. Besides, they utilize sampling techniques to amplify the privacy level to reduce the noise added to the mutual information calculation. A potential problem of the algorithm developed in [37] is that the constructed dependency graph may result in quite sizable cliques. When this occurs, the marginal distribution of the clique becomes high-dimensional, demanding a significant amount of noise for privacy protection. Cai et al. [32] proposed an iterative approach that greedily selects the node pairs with larger noisy scores, and insert the edges into the graph, while ensuring that the maximal clique in the triangulated graph remains below a specified threshold, which ensures there is no over-size clique in the tree. McKenna et al. [98] proposed a method, named Private-PGM, that aims to infer a data distribution by formulating an optimization problem that aligns the produced marginals closely with the observed ones. It does not offer a way to determine which marginals should be selected initially. One key consideration when working with Private-PGM is that the quality of the synthetic data it generates is heavily reliant on the accuracy of the provided marginals. To mitigate this issue, in their following work [96], they constructed a maximum spanning tree to select the marginals combined with some selection rules, then employed Private-PGM as a post-processing tool for distribution estimation given noisy marginals.

Synthetic data generation. Once the noisy marginals are obtained and the parameters of the potential functions are estimated, the joint distribution can be determined using Eq. 4. Following this, synthetic data can be sampled based on this joint distribution. However, as the dataset's dimensionality increases, Markov networks can become complex, resulting in computationally expensive when sampling from high-dimensional joint distributions. Junction tree is commonly applied to ensure efficient and precise inference [32].

Lessons learned and discussion. Compared to Bayesian networks, Markov networks offer greater flexibility in modelling relationships between variables. However, learning the optimal structure of a Markov network from data can be

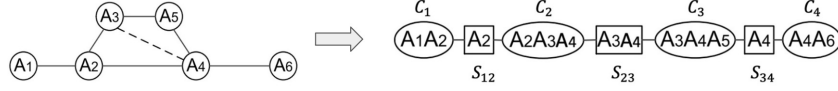


Fig. 1. An example of the creation of the junction tree (The figure is modified from [37])

computationally intensive and challenging, particularly as the number of variables increases. Estimating the parameters that define the potential functions, which describe how variables interact within the network, can also be difficult when data are scarce or the network structure is complex. Additionally, sampling from Markov networks, especially those with complex and densely connected structures, can be challenging and slow. Therefore, the junction tree algorithm is often utilized for probabilistic inference to manage these complexities effectively.

Junction tree. A junction tree is a data structure and associated algorithm used to facilitate efficient probabilistic inference in Markov networks. By constructing a junction tree, one can transform the Markov network into a tree-structured graphical model, which enables the calculation of joint and marginal distributions. Constructing a junction tree from a given dependency graph involves several steps. First, the graph must be triangulated by adding edges to ensure that any new edge between non-adjacent nodes forms a triangle, creating a chordal graph. Next, maximal cliques are identified within this triangulated graph; these are subsets of nodes that are fully connected and cannot be expanded without losing this connectivity. Then, for each pair of adjacent cliques, separator sets are determined. These sets consist of nodes that are common to both cliques and serve to separate them. Finally, the junction tree is constructed by representing each maximal clique as a node and each separator set as an edge connecting these nodes.

Figure 1 illustrates an instance of a junction tree denoted as T constructed from a dependency graph. When provided with an attribute set $A = \{A_1, A_2, \dots, A_d\}$, the estimation of the joint distribution $Pr[A]$ is achieved by leveraging the marginals of a collection of cliques C_i and their corresponding separators $S_{ij} = C_i \cap C_j$. The calculation [37] is shown as follows.

$$Pr[A] = \frac{\prod_{C_i \in T} Pr[C_i]}{\prod_{S_{ij} \in T} Pr[S_{ij}]} \quad (5)$$

Once the junction tree is constructed, the marginal distributions of the cliques are determined. These are then used to estimate the overall joint distribution, enabling the generation of synthetic samples.

Noisy marginal generation. The straightforward approach to get a noisy margin is to compute the marginals of the cliques within the tree and the marginals of the separators. Then, add differential privacy noise to each of these marginals. Since the separators represent the shared attributes between adjacent cliques, we can always derive the marginal distribution of the separators from the cliques. Therefore, the focus should be on obtaining noisy marginals for the cliques, which we can then use to get the separator marginals and estimate the joint distribution as described in Eq. 5. The straightforward approach comes with a potential challenge. Specifically, when dealing with a junction tree containing a substantial number of cliques, a larger privacy budget is needed to obfuscate these marginals. This, in turn, can lead to a reduction in the accuracy of the estimated joint distribution. To mitigate this issue, similar strategies to those applied when deriving the separators have been considered. In particular, Chen et al. [37] proposed to merge the cliques to minimize statistical variance and obtain the noisy merged marginal distribution first. Subsequently, they derive the marginal distribution for each individual clique.

Synthetic dataset generation. The joint distribution of the dataset, as defined by Eq. 5, presents computational challenges when it comes to sampling. To overcome this computational hurdle and efficiently sample data points from this

distribution, it is rewritten as Eq. 6 shown as follows.

$$Pr[A] \approx Pr[C_1] \prod_{i=2}^{|C|} Pr[C_i \setminus S_i | S_i] \quad (6)$$

Following Eq. 6, the synthetic dataset can be generated through a series of efficient local computations, as demonstrated by Cai et al. [32] and previously by Chen et al. [37].

Lessons learned and discussion. The junction tree algorithm is primarily associated with transforming Bayesian and Markov networks for efficient probabilistic inference. It is fundamentally a tool for simplifying and optimizing the inference process in complex graphical models. The advantage of using a junction tree is that it can directly sample from the joint distribution represented by the tree, using local computations within each clique and passing messages between cliques to maintain consistency across the entire network. But steps of triangulating the graph and identifying maximal cliques can be particularly challenging, especially for large or dense networks.

3.1.3 Query-based method. This stream of methods typically involves a series of queries that calculate the proportion of instances meeting certain criteria. It aims to respond to a wide range of statistical queries by creating a synthetic dataset from which the answers are derived.

Hardit et al. [65] proposed a Multiplicative Weights Exponential Mechanism (MWEM) to approximate the distribution across the dataset’s domain. It continually refines this approximation to enhance accuracy concerning both the private dataset and the desired query set by employing a combination of the Multiplicative Weights (MW) method [66] and the Exponential Mechanism. Specifically, it initializes a uniform distribution. Then, it identifies the query where the answer on the real data significantly differs from its corresponding answer on the approximate data, utilizing the Exponential Mechanism. It then increases the weights of the records that contribute positively to the approximation, while simultaneously decreasing the weights of records that have a negative impact.

MWEM offers simplicity in implementation and usage, demonstrating commendable accuracy in practical applications [65]. However, its efficacy diminishes when handling high-dimensional data due to its requirement to manipulate an object whose size scales linearly with the data universe’s size. Several follow-up research tries to make improvements on top of MWEM. For example, McKenna et al. [98] scaled MWEM by replacing the multiplicative weights update step with a call to Private-PGM, which does not need to materialize the full contingency table. Later, they proposed AIM [97], which refined MWEM-PGM [98] by implementing a superior initialization strategy, allocating a fraction of the privacy budget to measure 1-way marginals. Additionally, they devised new selection criteria, such as imposing more significant penalties on larger marginals, when selecting queries. Furthermore, they introduced adaptive rounds and budget allocation adjustments, further enhancing the statistical accuracy of the method. Liu et al. [87] improved the accuracy of MWEM by adaptively reusing past query measurements and selecting the synthetic data distribution with maximum entropy.

Gaboardi et al. [55] model the synthetic data generation process as a zero-sum game, involving two players: a data player and a query player, which was first proposed by Hsu et al. [68]. The data player’s action set is the entire data universe χ , and their strategy involves approximating the distribution of the true database. Conversely, the query player’s action set is the query class \mathcal{Q} . Gaboardi et al. [55] inverted the roles of the two players in the game, introducing a method named DualQuery. Specifically, it comprises a query player employing the no-regret learning algorithm and a data player that identifies optimal responses through the resolution of an optimization problem using MWEM. The proposed method is effective in managing high-dimensional data, but it doesn’t lead to an improvement in accuracy. Vietri et al.

[129] observed that the MW distribution changes slowly between rounds in no-regret dynamics. Leveraging this, they reuse previously drawn queries with rejection sampling to approximate the current MW distribution, enhancing the accuracy of DualQuery. Furthermore, since the MW method maintains a full distribution over the domain, causing MWEM to run in exponential time, they propose FEM and sepFEM. These methods, while still following no-regret dynamics, replace the MW approach with two variants of the follow-the-perturbed-leader (FTPL) algorithm, which solves a perturbed optimization problem instead of managing an exponentially large distribution.

To create synthetic data that most closely aligns with a noisy set of query measurements obtained from the original data. Several works use gradient-based optimization to solve this problem. Aydoore et al. [19] proposed a relaxed adaptive projection mechanism (RAP), which adaptively uses a continuous relaxation of the Projection Mechanism [104] to generate the synthetic dataset. It iteratively updates an initially synthesized dataset to minimize the l_2 distance between the noisy query results on the real dataset and those on the synthetic dataset. To conserve the privacy budget, rather than answering all the queries upfront, they start by answering a small number of queries. These are then projected onto a vector of answers consistent with a relaxed synthetic dataset. In subsequent rounds, they privately identify additional queries where the synthetic dataset underperforms, respond to these queries, and generate a new dataset through continuous projection. Vietri et al. [128] extended this research by introducing RAP++, which employs random linear projection queries to manage mixed-type data, eliminating the need for discretization of numerical features. Due to the differentiability requirement of the gradient-based optimization, they circumvent discretization by approximating non-differentiable queries with differentiable surrogate functions, which may introduce extra error due to the relaxation. To solve this problem, Liu et al. [87] proposed using a generative algorithm, which does not require differentiability in the optimization objective. In these methods, each iterative step involves updating the entire synthetic dataset to minimize the discrepancy between the real and synthetic data based on the new query responses. This process can be computationally demanding, especially as the size of the dataset and the number of queries increase.

Lessons learned and discussion. This group of methods typically requires a set of queries. Therefore, no need to make the selection of the marginals to estimate the joint distribution, all marginals are determined by the workload. An advantage of this approach is that it does not waste the privacy budget on selection processes, allowing the generated synthetic dataset to achieve more accurate performance to the queries in the set. However, if the set of queries does not comprehensively cover all relevant aspects of the data, the synthetic dataset might not adequately represent the characteristics of the original dataset beyond those queries.

3.1.4 Other methods. In addition to the mainstream statistical methods we have discussed, several other approaches have been proposed in the literature.

Maximum entropy optimization. Maximum entropy is a powerful and widely used principle for estimating joint probability distributions of multiple random variables, which has been widely adopted in the research of k -way marginal estimation [108, 151], where k can be as large as the number of attributes. Privacy is ensured by introducing noise into the selected lower-dimensional marginals. The key idea behind maximum entropy is to find the most unbiased or least informative probability distribution that is consistent with the available information (e.g., the marginals) or constraints (e.g., the consistent or non-negative constraint). It seeks to maximize the entropy of the distribution while satisfying these constraints. Existing studies [40, 108, 121] primarily focus on estimating low-dimensional marginals. Applying these methods to high-dimensional data would be computationally intensive.

Projection-based method. Xu et al. [138] proposed the use of the Johnson-Lindenstrauss transformation to project high-dimensional datasets into a lower-dimensional space. According to Johnson-Lindenstrauss’s theory, this transformation yields a reduced representation that preserves pairwise distances between points. Privacy is enhanced by adding noise to the projected dataset. While this method maintains the Euclidean distances between high-dimensional vectors, the resulting dataset often differs in shape from the original dataset, which may not be desirable.

Gradually update method. Zhang et al. [152] presented an alternative method in their work called the “gradually update method (GUM)” to generate the synthetic dataset utilizing the selected noisy marginal. This method begins with the initialization of a random dataset and then proceeds to iteratively update its records to ensure consistency with the provided marginals. The resulting data records generated using this method tend to generally align more closely to the noisy marginal statistics than those generated by methods like probabilistic graphical models. However, it is worth noting that the dataset updating process can encounter convergence issues that achieving convergence may not always be straightforward, making it a challenging aspect of this approach.

3.2 Deep learning-based method

Deep learning (DL) methods are widely utilized for image synthesis, typically handling homogeneous numerical data. However, there is an increasing interest in applying these techniques to tabular data, with adaptations being made to suit this purpose. Table 3 summarizes the proposed methods, each of which is discussed in the following sections.

3.2.1 Autoencoder (AE). The autoencoder is a highly prevalent unsupervised learning model with the primary objective of acquiring a compact data representation, often employed for dimensionality reduction [24, 41]. This neural network architecture operates by simultaneously training an encoder, responsible for converting high-dimensional data points into lower-dimensional representations, and a decoder, tasked with reconstructing high-dimensional data from the compressed representation. This process allows the model to capture essential features within the data while minimizing the overall data volume.

Chen et al. [36] trained a differentially private autoencoder and made the encoder available for generating synthetic data by inputting the user’s own data. The resulting synthetic data is then employed in downstream prediction tasks. However, the generated synthetic dataset is a low-dimensional data representation that differs from the original dataset in terms of data format. Abay et al. [6] applied the expectation maximization function to optimize the output of the encoded data and generate the synthetic data by decoding the encoded data.

Lessons learned and discussion. Such group methods cannot produce arbitrary synthetic datasets. Typically, autoencoders are used in conjunction with other generative models, such as GANs, serving as a preprocessing step to prepare the input data for the generative model.

3.2.2 Variational Autoencoder (VAE). A Variational Autoencoder is a generative model that combines the principles of autoencoders and probabilistic modelling. Different from AE, which only tries to reduce the dimension, VAEs are designed to learn and represent complex, high-dimensional data in a lower-dimensional space while simultaneously capturing the underlying probability distribution of the data [107]. An encoder network in VAE maps the input data to a probability distribution in the latent space, while a decoder network generates data samples from this distribution. VAEs use variational inference to model uncertainty in the latent space, which allows for the generation of not just deterministic reconstructions but also diverse and expressive data samples.

Table 3. Deep learning - based data synthetic methods

DL model	Methods	Privacy mechanism	Constant clipping	Privacy accountant	Utility evaluation	Privacy evaluation
AE	DP-AuGM [36]	DPSGD	Y	MA	classification	Y
	DP-SYN [6]	DPSGD DP-EM	Y	MA	k-way marginal classification agreement rate	N
VAE	DP-VaeGM [36]	DPSGD	Y	MA	classification	Y
	DPGM [8]	DP-k-means DPSGD	N	MA	linear query	N
	P3GM [120]	DP-PCA DP-EM DPSGD	Y	RDP	classification k-way marginal	N
GAN	DPGAN [135]	DPSGD	Y	MA	DWpre, DWpro	N
	dp-GAN [54]	DPSGD	N	MA	DWpre	N
	DPNet [50]	DPSGD	N	MA	realism k-way marginal	N
	PATE-GAN [77]	PATE	-	MA	classification	N
	G-PATE [89]	PATE	-	RDP	classification	N
AE+GAN	RDP-CGAN [124]	DPSGD	Y	RDP	MMD, DWpre	N
	DP-auto-GAN [123]	DPSGD	Y	RDP	DWpre DWpro k-way marginal	N
GN	GEM [87]	Exponential Gaussian	Y	zCDP	k-way marginal	N
	DP-MERF [64]	Gaussian	Y	RDP	classification	N
	DP-HP [131]	Gaussian	Y	RDP	k-way marginal classification	N

MA: moment accountant; RDP: Renyi differential privacy; DWpre: Dimension-wise prediction; DWpro: Dimension-wise probability; GN: Generative Network

Chen et al. [36] employed multiple private VAEs, with each VAE dedicated to generating synthetic data for a specific class. Their empirical findings revealed that training n generative models achieves higher utility compared to training a single model in terms of prediction accuracy. Similarly, Acs et al. [8] divided the data using k-means clustering and subsequently trained separate VAEs for each partition. In line with the conclusion in [36], they find that using multiple models resulted in more accurate synthetic samples, as it prevented the mixture model from generating unrealistic synthetic data that could emerge from improbable combinations of very distinct clusters. Despite these methods offering advantages, the synthetic data might not follow the original distribution. This is because the synthetic data is generated independently across separate datasets. Rather than training multiple VAEs, Takagi et al. [120] expanded the VAE architecture by employing a dimensional reduction function in lieu of embedding. Additionally, they fixed the encoder's mean to a constant value, thereby narrowing the search space and significantly accelerating the convergence speed, which in turn improves the model accuracy with a fixed privacy budget.

Lessons learned and discussion. In a VAE, a common assumption involves employing a Gaussian distribution in the latent space owing to its simplicity and ease of use. However, datasets with notable skewness or non-normal distributions may not conform well to this assumption. While it is possible to adapt the distribution of the latent space, the challenge lies in determining a suitable distribution that aligns with the specific characteristics of the data. In addition, more complex

distributions might capture fine-grained details but could also increase model complexity and computational requirements. Furthermore, VAEs might struggle with imbalanced datasets, potentially leading to difficulties in generating representative samples for minority classes or rare instances.

3.2.3 Generative Adversarial Network (GAN). Generative Adversarial Networks [62] are a class of deep learning models consisting of two neural networks, a generator and a discriminator, engaged in a competitive game. The generator creates synthetic data samples from random noise or other input sources. The discriminator evaluates both the synthetic data generated by the generator and the real data samples to differentiate them. The parameters of the generator and discriminator are updated based on the computed loss to improve the performance. The training process continues until the generator achieves the ability to generate synthetic data that closely resembles real data, and the discriminator reaches a point where its accuracy in distinguishing between the two levels is off. Due to the mode complexity of GAN, the training samples are easily remembered by the model [135]. To ensure privacy, two privacy models are considered in the literature, Differentially Private Stochastic Gradient Descent (DPSGD) [5] and Private Aggregation of Teacher Ensembles (PATE) [105].

DPSGD. The widely used framework DPSGD has been applied to the GAN training process, specifically adding noise to the gradient of the discriminator during training to provide provable privacy protection. Frigerio[54] and Fan et al. [50] optimized this process by reducing the clipping bound for each iteration, in turn, reduces the introduced noise. Fan et al. [50] further improved the performance by privately selecting the best model across all training epochs. Besides, they train an embedding model to capture the relationships between features. However, to save the privacy budget, the embedding model is required to train on a public dataset.

PATE. PATE employs an ensemble of teachers trained on different subsets of data, ensuring that no single model has access to the entire sensitive dataset [88]. In a typical GAN framework, there is a single discriminator trained in direct opposition to the generator. PATE-GAN [77], however, introduces k teacher discriminators alongside a student discriminator. To ensure differential privacy, the student discriminator is only trained on records generated by the generator and labelled by the teacher discriminators. This framework limits the influence of individual samples on the model, providing strong differential privacy guarantees. However, the approach assumes that the generator can cover the entire real data space during training. If most synthetic records are labelled as fake, the student discriminator could be biased and fail to learn the true data distribution. Different from PATE-GAN, Long et al. [89] proposed an ensemble of teacher discriminators to replace the GAN's single discriminator. A differentially private gradient aggregator is incorporated to collect information from these teacher discriminators, which guides the student generator to improve synthetic sample quality. Instead of ensuring differential privacy for the discriminator, noise is added to the flow of information from the teacher discriminators to the student generator.

In addition, GANs can suffer from mode collapse [150], where they generate limited varieties of samples, especially in complex and high-dimensional data spaces like tabular datasets. This can result in a lack of diversity in generated samples, failing to represent the full complexity of the original data distribution [120]. Addressing these challenges often involves modifications to the GAN architecture and exploring novel training methods tailored for the tabular data generation task. Some efforts have been made in the literature, for example, Fan et al. [50] learned an embedding during the training process to capture the relationships between attributes using a public dataset. Conditional GAN [100] was applied to deal with the imbalanced label distribution. Specifically, it encodes the label as a condition vector to guide the generator to generate samples with the label. Long et al. [89] proposed to utilize a small privacy budget to estimate the

class distribution in the training dataset and use the trained differentially private generator to generate data following the estimated class distribution. To overcome the model collapse problem of GAN, a variant of GAN, Wasserstein GAN [16], often abbreviated as WGAN, was adopted [10, 86]. It introduces the Wasserstein distance (also known as the Earth Mover’s Distance) as a more stable and informative metric for training generative models. Liu et al. [86] proposed to simplify the neural network architectures of the discriminator to limit the capacity of the discriminator and then avoid the chance of gradient disappearance of the generator. Xu et al. [140] proposed a mode-specific normalization based on the Gaussian Mixture Model to capture complex data distributions. However, this method does not offer privacy-preserving features.

Lessons learned and discussion. GANs have achieved significant success in image generation [35, 42, 85, 137], which can be represented in a continuous space. However, their application to tabular data is still in the early stages. Unlike image data, tabular data possesses unique characteristics that present challenges for their adoption in this context. First, tabular data often includes mixed data types; second, the attributes of tabular data are usually correlated. However, the majority of work on GANs focuses on making the synthetic data visually resemble real data [112, 125], often overlooking these correlations between features. Third, the datasets may exhibit highly imbalanced data distributions. Without careful consideration, this can result in insufficient training for records with minority labels.

3.2.4 AE+GAN. One challenge with GANs is that they are primarily suited for continuous data types, whereas tabular data often includes a mix of both categorical and numerical data types. Autoencoders can effectively address this issue as they are capable of encoding categorical data into a numerical format using techniques such as one-hot encoding and label encoding. Additionally, embedding layers in autoencoders can transform sparse categorical variables into dense, lower-dimensional representations, making them more amenable to processing by neural networks. Therefore, the AE is used in conjunction with GANs to handle mixed data types in tabular datasets [123]. Besides, autoencoders are designed to compress data into a lower-dimensional latent space while still preserving the key characteristics of the original data, which aids in capturing the correlations between features. Torfi et al. [124] have enhanced this capability by integrating one-dimensional convolutional neural networks (CNNs) within autoencoders.

Lessons learned and discussion. The combination of AE and GAN allows for better data representation and generation, leveraging the strengths of both models. However, Integrating multiple models might complicate the implementation of differential privacy mechanisms, making it challenging to ensure both privacy and utility.

3.2.5 Generative Network (GN). One stream of research proposes using a generator exclusively to create synthetic datasets. It employs techniques to quantify the distances between the information obtained from real data and synthetic data. The generator is designed to minimize this distance, thereby producing data that closely resembles the real data.

Harder et al. [64] proposed using Maximum Mean Discrepancy (MMD) [63] to quantify the distance between distributions in a Hilbert space [25]. The distributions of real and synthetic datasets are transformed into the Hilbert space using kernel mean embeddings (KME), which compute the mean of kernel evaluations. Gaussian noise is added to the kernel mean embeddings of the real dataset to ensure privacy. Due to the resource-intensive nature of computing KMEs, they approximate the KMEs using the inner product of feature vectors. However, since a large number of random features are required to achieve a good approximation, the noise increases significantly. To address this issue, Vinaroz et al. [131] used Hermite polynomial features, capturing more information with fewer features. Additionally, to tackle the imbalance issue in tabular datasets, they propose allocating part of the privacy budget to obtain the statistics of class counts and modify the released mean embeddings by appropriately weighting the embedding for each class. Liu

et al. [87] defined the distance based on a set of queries. Specifically, they defined the loss function as the distance between the query results on the synthetic data distribution and the noisy query results on the real dataset. The work focuses on query release, optimizing the distribution to provide accurate query answers. The sampled synthetic data may not accurately represent the original data characteristics.

Lessons learned and discussion. Using a generative network with only a generator to produce synthetic data has several advantages. The architecture is simpler, avoiding the complexities and potential instability issues present in more complex generative models like GANs, and it makes privacy noise addition more straightforward. However, designing an effective loss function that accurately captures the data distribution differences can be challenging due to the limited feedback.

4 DISTRIBUTED DATA SYNTHESIS WITH DP

Decentralized data synthesis involves the collaboration and synthesis of data from multiple parties or sources. It becomes especially valuable when one party lacks a sufficient dataset for meaningful analysis or insights. In a decentralized setup, a semi-trust server typically coordinates the learning process. Rather than sending the raw dataset to the server, each client shares statistical or intermediate results from their local models. Together, they collaboratively learn a synthetic data generative model. To safeguard clients' data, differential privacy noise is incorporated into the information transmitted to the server. We categorize the existing works into two types: vertical data synthesis and horizontal data synthesis. Table 4 summarizes these works.

Table 4. Distributed data synthesis with differential privacy

Data synthesis method	Data partition	Model		Architecture		Privacy	Key limitation
		Statistics	DL	MD	FL		
DistDiffGen[101]	V	CT				ϵ -DP	Only suitable for two-party scenario
DPLT[122]	V	BN				ϵ -DP	Limited to discrete attributes
VertiGAN[73]	V		GAN	✓		$(\alpha, \epsilon(\alpha))$ -RDP	Cannot deal with imbalanced data
GTV[156]	V		GAN		✓	-	Server cannot collude with any client
DP-SUBN [118]	H	BN				ϵ -DP	Split privacy budget needs too much
Fed-TGAN[154]	H		GAN		✓	-	Joint distribution is not considered
DP-CTGAN [51]	H		GAN		✓	(ϵ, δ) -DP	No privacy for the feature statistics
HT-Fed-GAN [46]	H		GAN		✓	(ϵ, δ) -DP	High communication cost
ATLAS [132]	H		GAN	✓		$(\alpha, \epsilon(\alpha))$ -RDP	Cannot deal with the skewed dataset

V: Vertical; H: Horizontal; CT: Contingency table; BN: Bayesian Network; MD: Multi-discriminator structure; FL: Federated learning structure

4.1 Vertical data synthesis

Vertically partitioned data refers to scenarios where different subsets of attributes of the same dataset are stored across different parties or locations. It is quite common in various fields, including healthcare, finance, retail and more. Many organizations naturally organize their data based on different departments, divisions, or functional areas. For example, in a healthcare scenario where patient information is vertically distributed across various departments within a hospital. Merging these segregated datasets helps to generate a comprehensive patient profile. In another scenario, datasets may originate from distinct organizations, like a bank with customer income records and an e-commerce company with customer purchase histories. By combining these disparate datasets, each organization gains valuable insights that augment their respective analyses by leveraging diverse information across sectors. Specifically, within the context of

vertical data synthesis, each client holds a local dataset containing distinct features related to the same individuals. A specific challenge in vertical data synthesis is capturing the relationships between columns across datasets from different clients.

Statistical method. Mohammed et al. [101] introduced the initial method for releasing vertically partitioned data that forms an integrated data table. Their approach involves using a predefined attribute taxonomy tree as publicly available information, along with the distributed exponential mechanism to produce the generalized data table. Additionally, they apply Laplace noise to the true count at leaf nodes to guarantee differential privacy. However, this method is only suitable for a two-party scenario due to the limitations of various underlying cryptographic primitives. And the data utility deteriorates quickly with the increase in the number of attributes. Tang et al. [122] expanded the algorithm's capacity to handle a larger number of attributes by leveraging a latent tree model [148]. They utilized this model to capture the dependencies among attributes, followed by privatizing these latent tree parameters using a distributed Laplace protocol. The acquired model was then employed to generate synthetic data. The proposed method enhances data utility to a certain extent, yet the rise in data dimensionality could still result in significant utility loss. Moreover, its applicability remains restricted to discrete attributes.

Deep learning-based method. The Distributed GAN framework has been employed for collaborative model training within a vertical setting. Two architectural variations, depicted in Figure 2, have been investigated. In the multi-discriminator structure, the server maintains a global generator, while each client possesses a discriminator; together, they collaboratively train a global generator. In the federated learning structure, the server initializes both a global generator and discriminator. Each client trains its local generator and discriminator on its dataset, updating the global models by aggregating parameter transformations from all clients.

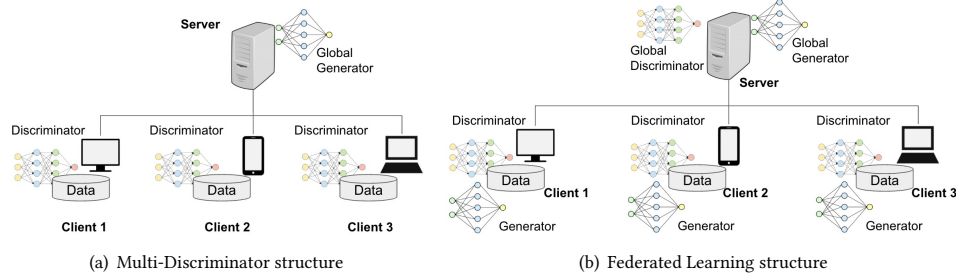


Fig. 2. Architecture of Distributed GAN. The left figure shows the multi-discriminator structure, where the server maintains a global generator and each client possesses a discriminator. The right figure shows the federated learning structure, where the server initializes both a global generator and discriminator and each client trains its local generator and discriminator on its dataset.

Jiang et al. [73] adopted a distributed GAN architecture consisting of one global generator on the server side and multiple discriminators on the client side. The global generator processes random latent features to generate synthetic data tailored for each local client. Meanwhile, the local discriminators are trained within each client's data to differentiate between real and synthetic data. DP perturbation is applied to the discriminator to protect data privacy. The proposed method successfully addressed the issue of attribute inconsistency and established connections among attribute columns across different clients. However, it does not account for the commonly occurring problem of data imbalance in tabular data. Consequently, its performance cannot be guaranteed when dealing with such imbalanced data distributions.

Zhao et al. [156] solved the data imbalance issue by applying a conditional GAN. It manages the generation of data by leveraging a conditional vector that controls the class of the generated information. However, this approach comes with inherent risks without appropriate perturbation. The server can potentially reconstruct client data by using the constructed conditional vector and the indices of selected training data. Even though a strategy is proposed to shuffle clients' datasets using the same random seed, there remains a significant vulnerability: if a single client collaborates with the server, the information can easily be exposed. The proposed method doesn't employ differential privacy due to concerns about data utility. However, they did note that ensuring privacy could be achieved directly by introducing noise into the communicated intermediate result.

Lessons learned and discussion. The predominant emphasis of current research lies in the implementation of such data synthesis within a vertical framework. However, there's a notable scarcity in addressing specific challenges on tabular datasets, like handling imbalanced data distributions and managing the high dimensionality of attributes under differential privacy constraints. Furthermore, existing works commonly assume that all clients possess an identical set of records/individuals or can establish this through a secure set intersection protocol. This assumption presents a challenge since obtaining a substantial amount of identical user records from multiple parties for learning purposes might be impractical.

4.2 Horizontal data synthesis

In a horizontal data distribution scenario, data records are spread across multiple nodes or servers, typically in a manner that divides data rows or entities across these nodes. For instance, various banks might hold the same categories of data about their customers, such as names, addresses, transaction histories, account balances, and so forth. Each bank can store this kind of information within its database, specifically for its customers, while maintaining separate entities from other banks. Specifically, within the context of horizontal data synthesis, each client holds a local dataset containing the same features related to different individuals. The particular challenge in synthesizing horizontal data is accurately estimating the overall data distribution, considering that each client's distribution may be skewed compared to the entire dataset.

Statistical method. Su et al. [118] implemented multi-party data synthesis by collaboratively constructing a Bayesian Network. The parties and the server worked together to quantify correlations between attribute pairs and initialized a Bayesian network. Each party then sequentially updated it based on their local dataset. To manage noise and reduce communication costs, they constructed the search frontier of the network using only strongly correlated pairs. The sequential update method maximized the use of dependency information from previous parties, further reducing the number of candidate attribute pairs. Besides the inherent limitations of using Bayesian networks to generate synthetic data, it necessitates multiple uploads of information by each party, splitting the privacy budget into four pieces, thereby significantly impacting statistical accuracy.

Deep learning-based method. Zhao et al. [154] adopted the Federated learning collaborative training framework using CT-GAN [140]. To efficiently encode the features, they suggested gathering private frequency information related to categorical attributes and parameters of a Variational Gaussian Mixtures (VGM) model [18] from individual users for each column. Following model initialization, this statistical information is reintegrated to compute weights for each client, aiding smoother convergence in scenarios of imbalanced data across different clients. Yet, the weight calculation relies on individual columns, disregarding the joint distribution of attributes. Also, it didn't explain how the concept

of differential privacy could be used in the suggested method, even though it mentioned its potential application. It claimed that privacy is ensured by the transfer of statistical information rather than raw, original data. Fang et al. [51] incorporated differential privacy to CT-GAN by adding noise to the discriminator and deploying it in a federated setting as well. However, the feature encoding process is protected, which can still disclose some information about the original data. Besides the VGM model, the Variational Bayesian Gaussian Mixture Model (VB-GMM) [39] is also considered [46] for modelling the distribution of the data column. In contrast to VGM, VB-GMM takes a Bayesian approach to estimate the model parameters and doesn't require specifying the number of clusters in advance. Duan et al. [46] implemented the data encoding process using Homomorphic Encryption [15]. To prevent the discriminator from memorizing the private data during training, they introduced noise during the aggregation step of the discriminator. To well balance the privacy and data utility, Wang et al. [132] proposed to adaptively adjust the noise scale to reduce the impact of gradient perturbation. Additionally, they utilize the discriminator to filter out more realistic synthetic data records during the data generation process.

Lessons learned and discussion. Few studies explore the use of statistical models for distributed data synthesis. Further research into advanced methods is needed to address high-dimensional challenges in collaborative learning of these models. For deep learning approaches, CT-GAN is used as the main tool in synthesizing tabular data through horizontal federated learning. To safeguard information during the feature encoding process, other privacy techniques like HE are employed. While it helps with privacy, it does increase communication and computation costs. Developing better ways to allocate privacy budgets and aggregate information could strike a better balance between privacy and data utility.

5 SYNTHETIC DATA EVALUATION

Evaluating synthetic data is essential to ensure that it serves its intended purpose effectively while safeguarding sensitive information. Many evaluation techniques are designed for specific types of data or particular domains [112]. In this section, we focus on evaluation methods for tabular data, with the taxonomy shown in Fig. 3.

5.1 Fidelity evaluation

Fidelity evaluation for synthetic data refers to the process of assessing how well the generated synthetic data matches the properties and patterns of the real data it aims to replicate. The primary goal is to ensure that the synthetic data is a reliable and accurate representation of the real data in terms of statistical, structural, and distributional characteristics.

5.1.1 Human intervention. This method is the most straightforward way to evaluate the fidelity of synthetic data. The metrics are concerned with how accurately the synthetic data replicates or aligns with the real-world.

Human inspection. This method requires the experts to judge whether the data is real or synthetic. Since the experts have professional knowledge, it helps them to identify fake data. For example, considering a clinical dataset, the fake data record might include a medical history with contradictory information, such as a patient having two mutually exclusive medical conditions or treatments that are incompatible [23].

Realism. Realism is similar to Human inspections, involving the verification of its alignment with domain-specific knowledge. It works by identifying a specific test, for each test, specifying the criteria that must be satisfied to classify the dataset as realistic. For instance, when evaluating a synthetic network traffic dataset [50], if a flow describes normal user behavior and involves source or destination ports 80 (HTTP) or 443 (HTTPS), the transport protocol must be TCP.

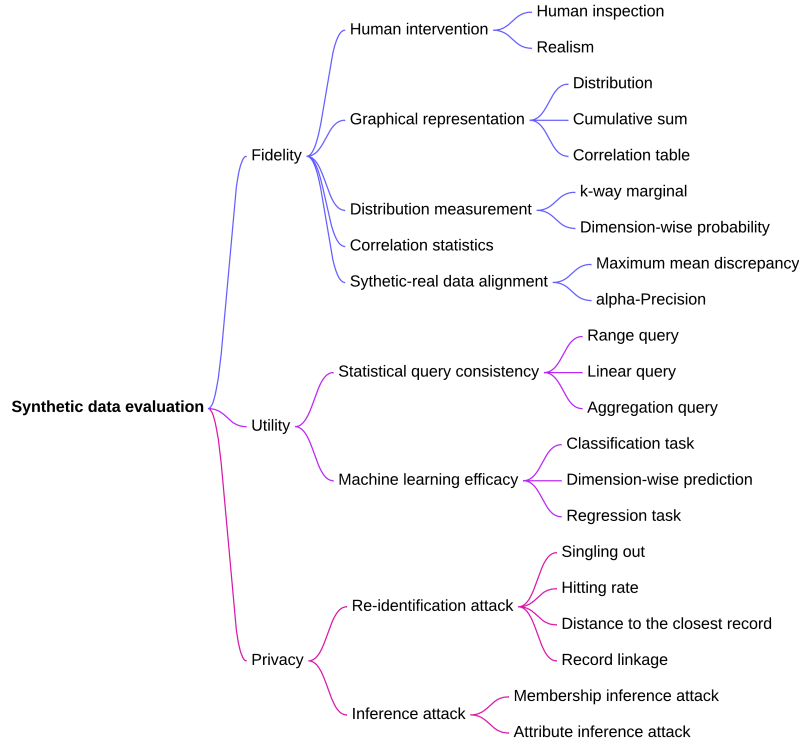


Fig. 3. The taxonomy tree of the synthetic data evaluation methods

5.1.2 Graphical representation. One method to evaluate the fidelity of synthetic data is through graphical representations. These visual assessments allow for easy verification of results and recognition of similar patterns between the real and synthetic data.

Distribution. It visualizes the distribution of each column or multiple columns of the real and synthetic data [94], allowing for a visual assessment of how well the synthetic data matches the original data. A Q-Q (Quantile-Quantile) plot is an effective graphical method that compares the quantiles of the distribution of a synthetic dataset to the quantiles of the distribution of the original dataset. By plotting these quantiles against each other, the Q-Q plot helps identify whether the synthetic data preserves the statistical properties of the original data, highlighting any discrepancies between the two datasets.

Cumulative sum. Visualizing the cumulative sum of each column for real and synthetic data can reveal the similarity between their distributions as well. This method effectively demonstrates the distribution of both continuous and categorical values, facilitating a thorough comparison between the real and synthetic datasets.

Correlation table. A correlation table displays the correlation coefficients between multiple variables in a dataset, with each cell representing the correlation between two variables. Comparing the correlation metrics can indicate how well the synthetic data captures the relationships between the columns [61], thereby assessing the preservation of

statistical dependencies in the synthetic data. These correlation coefficients can also be visually represented using a heatmap or a correlation matrix plot.

5.1.3 Distribution measurement. Distribution evaluation for synthetic data involves assessing how well the statistical properties of the synthetic data align with those of the real data.

k -way marginal. k -way marginal, which refers to a marginal that consists of k attributes, is the most commonly used metric to evaluate the statistical consistency between the synthetic dataset and the real datasets. As k increases, higher-order attribute correlations are measured. Various metrics can be used for quantifying the statistical errors, including L_1 -distance, L_2 -distance and L_∞ -distance. Besides, Hardt et al. [65] utilized Kullback-Leibler (KL) divergence to measure entropy differences. While Gambs et al. [56] employed Kolmogorov-Smirnov distance for CDF comparisons. Du and Li [44] proposed using Wasserstein distance to measure the distribution discrepancies. Additionally, total variation distance [79] and the Kolmogorov-Smirnov test [146] are commonly used to assess distribution similarity for categorical and numerical attributes, respectively.

Dimension-wise probability [38]. Dimension-wise probability analysis is a fundamental validation technique used to assess a model's understanding of the distribution of data in each dimension. It is used for evaluating the model performance for binary variables. The process involves training the model on a real dataset and generating an equal number of synthetic samples. For every dimension k , it calculates the Bernoulli success probability (p_k) to estimate the likelihood of a specific outcome in that dimension. The p_k values for each dimension in the real dataset are then compared to those in the synthetic samples. This comparison serves as a sanity check to determine whether the model has successfully captured the dimension-wise data distribution, ensuring the model's reliability.

5.1.4 Correlation statistics. Correlation evaluation for synthetic data focuses on assessing how well the relationships between variables in the synthetic data match those in the real data. Correlations are crucial for maintaining the structural integrity of the data, as they capture dependencies and associations between features. Researchers evaluate the similarity of pairwise relationships between variables in synthetic and real datasets by comparing correlation scores. Commonly used correlation measures include Theil's uncertainty coefficient [155], Pearson correlation [146], and the correlation ratio [79]. Additionally, Gambs et al. [56] introduced the Mean Correlation Delta, a metric designed to quantify the differences in correlation coefficients between the correlation matrices of the real and synthetic datasets.

5.1.5 Synthetic-real data alignment. The group of metrics evaluate how closely the synthetic data captures the patterns and characteristics of the real data, ensuring a strong alignment between the two.

Maximun mean discrepancy (MMD). MMD is a statistical measure that compares the similarity or difference between two sets of data points. Given two sets of data points, often denoted as P and Q , first represent them in a higher-dimensional space using a kernel function. The choice of the kernel function affects how data points are mapped. MMD calculates the difference in mean values between the two sets in this higher-dimensional space. If the mean values are close, it suggests that the two sets are similar, and if they are significantly different, it suggests dissimilarity. In a recent study [141], MMD demonstrated most of the desired features of evaluation metrics, especially for GAN.

α -Precision. α -Precision is proposed in [9]. It is an enhanced version of the conventional Precision metric. While Precision checks how often generated data matches real data, α -Precision goes further by focusing on a specific portion of the real data distribution. It evaluates whether a synthetic data point falls within the most important or central part

of the real data distribution, defined by a parameter α . In doing so, it measures the fidelity of synthetic data, ensuring that it captures the core characteristics of the real data, rather than just any random part. The core part refers to the region of the data distribution where the majority of real data points are concentrated, representing the most significant or typical patterns within the data.

5.1.6 Discussion. Human evaluation methods, though practical, are rarely used in academic literature to assess synthetic data quality due to their cost and time demands. Graphical evaluation visually compares real and synthetic data, aiding similarity assessment, but becomes resource-intensive and complex with large, high-dimensional datasets. The simplest method to evaluate the fidelity of synthetic data is by using basic statistics such as mean, median, and variance [34]. The closer these statistics are between the real and synthetic data, the more similar the datasets are considered. However, this approach may not capture complex relationships within the data. Francis Anscombe demonstrated this with his famous “Anscombe’s quartet” [14], where four datasets had nearly identical basic descriptive statistics but vastly different distributions. This illustrates the need for more sophisticated and in-depth statistical analysis to evaluate synthetic data.

5.2 Utility evaluation

Utility evaluation is the process of assessing how useful synthetic data is for specific tasks or applications, typically by comparing its performance with that of real data. The goal is to ensure that synthetic data preserves enough meaningful information from the original data to be effectively used in analysis, decision-making, or machine learning models.

5.2.1 Statistical query consistency. One common metric used to evaluate the utility of synthetic data is through statistical queries. The focus is on assessing how well the synthetic data preserves the statistical properties of the real data when responding to specific queries. This process determines whether the synthetic data retains enough accuracy and relevance to be used for meaningful analysis, ensuring its utility for downstream tasks.

Range query. A range query is a type of database query that retrieves all records where the values in a specific field fall within a specified range. This is particularly useful for querying numerical, date, or other ordered data types. The range query can involve one or more attributes. For instance, it might inquire about the percentage of data records with ages between 40 and 60 and salaries between 1000 and 2000 dollars. Range queries are commonly used in evaluating synthetic data [82, 84].

Linear query. Linear query, in the context of synthetic data generation or data analysis, typically refers to a query that can be expressed as a linear combination of features or attributes in a dataset [138]. Notably, in the work of Acs et al. [7], a linear query is defined as a predicate function that calculates the count of instances in the dataset that meet the specified predicate criteria.

Aggregation query. An aggregation query is a query that processes and summarizes data to produce a single result or a set of results based on grouped data. These queries are commonly used to compute aggregate values like sums, averages, counts, minimums, maximums, and other statistical metrics. Fan et al. [49] ran the same aggregation queries with aggregate functions, such as count, average and sum, on both original and synthetic datasets, and then measured the relative error of the results.

5.2.2 Machine learning efficacy. Machine learning-based evaluation involves using machine learning models to assess the utility of synthetic data. This evaluation can be done by performing specific predictive tasks, and comparing the performance of these models when trained on real data versus synthetic data.

Classification task. Classification tasks are commonly employed to assess the preservation of attribute correlations in relation to attribute prediction. This involves training classifiers on both the original and synthetic datasets and then testing them on a shared test set. Performance is usually compared using metrics like misclassification rate or F1 score. Besides, Bindschaedler et al. [26] introduced the agreement rate, which measures the percentage of identical predictions. Du and Li [44] proposed Machine Learning Affinity to measure the relative performance discrepancy across models. Ideally, the classifier trained on the synthetic dataset should exhibit classification performance similar to that of the one trained on the original dataset.

Dimension-wise prediction. Dimension-wise prediction was proposed by Choi et al. [38] to quantify the binary variables. This approach seamlessly extends to handling multiple variables. Differing slightly from traditional classification methods, this approach allows the flexibility of selecting any attribute as the label for prediction.

Regression task. The regression task has been employed to assess synthetic data as well in the literature [56]. This involved training two predictors separately on synthetic and real datasets. Subsequently, these predictors were utilized to forecast values for a specific attribute within the test dataset. The ensuing step involved a comparative analysis of the predicted values generated by the two distinct predictors.

5.2.3 Discussion. Query-based metrics are typically computationally efficient and do not require complex models or large-scale processing. However, they only evaluate specific statistics or properties being queried, which means they may not fully capture the performance of synthetic data in task-specific scenarios. Machine learning-based evaluation directly assesses how well synthetic data supports specific tasks, providing clear and quantitative metrics for comparison. However, training and evaluating machine learning models can be computationally intensive, particularly with large datasets or complex models such as deep neural networks. Besides, it is well known that tabular data demonstrates varying performance across different machine learning models, with no single model consistently achieving optimal results on all datasets [44]. Consequently, making fair comparisons using machine learning-based methods becomes challenging and needs careful design.

5.3 Privacy evaluation

Privacy evaluation for synthetic data assesses how well it preserves the privacy of individuals in the original dataset. Even with differential privacy techniques, residual risks may persist, making accurate risk quantification essential. Limited research exists on privacy risks in differentially private synthetic data, but attack-based assessments offer valuable insights into protection levels. The generic privacy risk evaluation assumes the synthetic data generation process is a black box. In this scenario, the attacker can only access the generated synthetic data and attempt to deduce sensitive information by executing various attacks on the synthetic dataset. The attacker might possess some background knowledge about the individuals involved.

5.3.1 Re-identification attack. One group of evaluation aims to quantify the risk associated with re-identifying the original data records.

Singling out. The intuition behind this method is that individual data records can be isolated if their attributes or attribute connections are unique in the synthetic data. In the case of numerical attributes, Xue et al. [60] employed the minimum and maximum values of each attribute to establish predicates based on values falling below the minimum or exceeding the maximum, enabling the identification of outliers.

Hitting rate. The hitting rate measures the number of records in the original dataset that a synthetic record can match. Two records are considered similar only if all categorical attribute values are identical, and the discrepancy between numerical attribute values falls within a defined threshold [49].

Distance to the closest record. This metric assesses the proximity of a synthetic data record to the nearest real data record. A distance of 0 indicates that the synthetic data perfectly replicates a real data record, potentially revealing actual information. Conversely, a greater distance signifies stronger privacy protection, indicating a larger divergence from the real data [90].

Record linkage. The assessment of privacy risk relies on the success rate of identifying connections between synthetic data records and their corresponding original data records. Various record linkage methods, including probabilistic [72] and distance-based approaches [95], can be employed for this evaluation. Lu et al. [90] utilized the Python Record Linkage Toolkit to compare record pairs.

5.3.2 Inference attack. Another group of evaluations seeks to measure the synthetic dataset’s capacity to deduce information about the membership and specific sensitive attributes.

Membership inference attack. A membership inference attack [113] on synthetic data aims to determine if a particular individual’s data contributed to training the synthetic generative model. Stadler et al. [116] attempted to attack synthetic data by utilizing handcrafted features from the synthetic data distribution to train shadow models. Hyeong et al. [71] estimated the likelihood of target records using density estimation and applied this as a confidence score for membership inference. Gambs et al. [56] introduced the Monte Carlo attack, which calculates the count of synthetic data records neighbors of the queried data record, thereby inferring its potential presence in the training dataset.

Attribute inference attack. Attribute inference operates under the assumption that the attacker possesses knowledge about certain attributes of the victim. Leveraging this background information and the synthetic dataset, Giomi et al. [60] deduced sensitive attributes by searching for the closest data record containing the known attributes. The sensitive attributes in the nearest data record constitute the attacker’s estimated information. Stadler et al. [115] performed the attribute attack by training a machine learning model on the synthetic dataset and subsequently predicting the sensitive attribute of the victim based on the known attribute values. Annamalai et al. [12] proposed a linear reconstruction attack method that generates a series of marginal queries to infer the sensitive attribute by minimizing the query error based on the known victim attributes.

5.3.3 Discussion. Beyond generic attacks, if an attacker gains access to the data generation model, traditional membership [69] and attribute [149] attacks for machine learning could be employed to assess privacy risks. While most of these attacks traditionally focus on image data, exploring effective methods to evaluate the risks of tabular data utilizing such attacks could be an intriguing study area. Besides, developing a unified framework to quantify the privacy risk could benefit society. It can inform the development of privacy regulations and policies by providing a structured approach to assessing and mitigating risks, aiding in establishing effective guidelines and standards.

6 RESEARCH GAPS AND FUTURE DIRECTIONS

This section explores current research gaps, highlights areas where current solutions fall short, and identifies potential future research directions. It focuses on design, privacy, evaluation methods, and emerging application scenarios that present promising opportunities for future research advancements

6.1 Tailored DL Methods for Tabular Data Synthesis.

Deep learning-based data synthesis methods offer a powerful and flexible approach to generating synthetic data. These methods can efficiently handle high-dimensional datasets and scale well. While they show great potential for tabular data synthesis, most existing studies have overlooked the unique characteristics of tabular data. These include mixed data types, unbalanced data distributions, and column interdependencies. Some strategies have been proposed to address these challenges [123, 156]. For instance, autoencoders are commonly used to handle mixed data types, while some deep learning models introduce conditional GANs to manage unbalanced data distributions. However, research in this area is still in its early stages, and more effective strategies need to be explored to enhance the capability of deep learning models for tabular data synthesis. These strategies may include the structural design of neural networks and the integration of specific statistical components to capture the distribution and dependencies of the dataset.

6.2 Privacy risk evaluation.

While DP offers a provable level of privacy protection, it remains unclear how much privacy risk remains at a fixed privacy level ϵ . In other words, it is uncertain how large a privacy budget can provide sufficient privacy protection. A study [115] suggested that generative models trained without privacy-preserving techniques (non-private generative models) offer limited protection against inference attacks. Furthermore, the researchers discovered that training generative models with differential privacy did not significantly enhance protection against inference attacks. Another recent study [71] found that synthetic datasets generated by DP-GAN exhibit better resistance to black-box attacks, but white-box attackers can still accurately infer membership. This raises questions about the effectiveness of other synthetic data generation models, such as marginal-based methods, in safeguarding privacy. We may be interested in assessing the effectiveness of privacy protection against various attacks or exploring methods to strike a balance between preserving privacy and maximizing utility.

6.3 Differentially private tabular data synthesis with the diffusion models.

Diffusion models have recently been explored for tabular data synthesis since their flexible probabilistic framework can effectively model complex data distributions and capture the dependencies. Several works [79, 81, 109, 119, 146] addressed the challenges of synthesizing tabular data with mixed data types and varied distributions. Yang et al. [145] focused on ensuring fairness in synthetic data generation using diffusion models. Villaizan et al. [130] and Jolicoeur et al. [74] explored generating and imputing tabular data with diffusion models. Additionally, some studies propose using federated learning for tabular data generation while maintaining data privacy. Sattarov et al. [110] combined denoising diffusion probabilistic models with federated learning to enhance data privacy in the horizontal scenario. Meanwhile, Shankar et al. [111] investigated data synthesis using diffusion models with vertically partitioned data. However, the use of diffusion models for synthetic data generation has not been explored in conjunction with differential privacy, either in centralized or distributed settings. Investigating the challenges and balancing privacy and utility when integrating differential privacy into the diffusion model, which offers robust protection during the data synthesis process, would be a valuable area of research.

6.4 Disparity effect of DP.

DP noise often leads to negative values and introduces inconsistencies within marginal distributions. Various post-processing techniques are commonly employed to enhance the accuracy of statistical results [143]. However, it is

important to note that while post-processing techniques can improve statistical accuracy, they may introduce some bias in the data [142, 157]. The bias introduced during the post-processing of noisy data can propagate and become more pronounced in downstream tasks using synthetic data. This observation is demonstrated in the research conducted by Ganey et al. [57]. Their recent study empirically illustrates the differential privacy’s varying impact on synthetic datasets. Particularly, classifying tasks on the private synthetic dataset influences the gaps between the majority and minority subgroups. PrivBayes reduces this disparity, while another GAN-based model exacerbates it. Research on the disparity effects of applying differential privacy in synthetic data generation is still limited. Further exploration of contributing factors and developing mitigation strategies are crucial areas of study.

6.5 Multi-party data synthesis with different users.

In distributed data synthesis scenarios, particularly when datasets are vertically partitioned, the literature often assumes that the two parties have precisely the same user groups or that overlapping users can be identified through Private Set Intersection (PSI) techniques [102]. The synthetic data learning process is then conducted on these overlapping users. However, it is rare for different organizations to have the same users in practice. A significant issue arises when the size of the overlapping user group is small. On the one hand, it becomes challenging to train an effective generative model; on the other hand, it leads to substantial data resource wastage, as most data cannot be utilized for model learning. Therefore, developing efficient collaborative data generation methods that consider scenarios where parties hold different user groups is promising and essential. Such methods should utilize all available data to improve the quality of the generated models.

6.6 Differentially private tabular data synthesis with LLM.

The popularity of Large Language Models (LLMs) has surged in recent years, driven by their remarkable capabilities in natural language processing. While initially designed for text-based tasks, LLMs have been adapted to work with structured data like tables [27, 52]. Borisov et al. [28] proposed using pre-trained LLMs for synthesizing tabular data in a non-private setting. By treating each row as a sequence of tokens, LLMs can generate tabular data like generating text, showing promising results that outperform GANs in synthesizing tabular data. However, when differential privacy is applied using DPSGD, Tran and Xiong [126] demonstrated that traditional DP fine-tuned LLMs struggle to generate tabular data with format compliance due to the injected noise. To address this, they proposed a two-phase fine-tuning method that first learns the format and then fine-tunes the model to capture the feature distributions and dependencies of the dataset. The use of LLMs for generating synthetic data, especially under differential privacy constraints, is still in its early stages. Since LLMs are primarily designed for unstructured text, adapting them to generate structured or tabular data presents significant challenges, especially when it comes to capturing the complex relationships between columns and rows. Additionally, achieving the right balance between ensuring privacy and maintaining the utility of the generated data remains an ongoing challenge.

6.7 Multi-relational data synthesis

Research on tabular data synthesis primarily focuses on single-table scenarios, while in practice, data is often stored in multiple interconnected tables as relational databases. The main challenge in multi-table synthesis is capturing the long-range dependencies caused by foreign-key relationships. Few studies have explored this area. For instance, Mami et al. [93] employed graph variational autoencoders to model the synthesis process, while Patki et al. [106] used Gaussian copulas to capture parent-child relationships. Another work [33] leveraged the controlled generation

capabilities of diffusion models, utilizing clustering labels as intermediaries to model relationships between tables. To the best of our knowledge, only one study addresses the synthesis of relational data with foreign keys under differential privacy. The key idea is to model the data distribution using graphical models, incorporating latent variables to capture inter-relational correlations induced by foreign keys. Calibrated noise is injected into the model training algorithm to ensure differential privacy. Incorporating differential privacy into multi-relational data synthesis is still an open problem, particularly for large-scale databases with complex dependencies among hundreds of interconnected tables. DP noise makes it difficult to capture cross-table correlations, and distributing the privacy budget effectively across interconnected tables while still ensuring the privacy of sensitive information presents another significant challenge.

7 CONCLUSION

In this paper, we conducted a comprehensive review of existing DP methods for synthesizing tabular data, a widely used data type in finance, healthcare, and other domains. We categorized these methods into statistical methods and deep learning-based approaches based on the data generation models, discussed in both centralized and decentralized settings. We examined and compared the methods in each category, highlighting their strengths and weaknesses regarding utility, privacy, and computational complexity. Furthermore, we provided a comprehensive overview of evaluation techniques for data synthesis, including fidelity, utility and privacy. From this analysis and discussion, we identified the research gaps and several potential directions for future research.

REFERENCES

- [1] 2018. 2018 Differential Privacy Synthetic Data Challenge. <https://www.nist.gov/ctl/pscr/open-innovation-prize-challenges/past-prize-challenges/2018-differential-privacy-synthetic>
- [2] 2024. EU AI Act: first regulation on artificial intelligence. <https://www.europarl.europa.eu/topics/en/article/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence>
- [3] 2024. The Home of the U.S. Government’s Open Data. <https://data.gov/> Last updated: 2024-10-25.
- [4] 2024. Kaggle datasets. <https://www.kaggle.com/datasets> Accessed: 2014-9-20.
- [5] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 308–318.
- [6] Nazmiye Ceren Abay, Yan Zhou, Murat Kantarcioglu, Bhavani Thuraisingham, and Latanya Sweeney. 2019. Privacy preserving synthetic data release using deep learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I* 18, Springer, 510–526.
- [7] Gergely Acs, Claude Castelluccia, and Rui Chen. 2012. Differentially private histogram publishing through lossy compression. In *2012 IEEE 12th International Conference on Data Mining*. IEEE, 1–10.
- [8] Gergely Acs, Luca Melis, Claude Castelluccia, and Emiliano De Cristofaro. 2018. Differentially private mixture of generative neural networks. *IEEE Transactions on Knowledge and Data Engineering* 31, 6 (2018), 1109–1121.
- [9] Ahmed Alaa, Boris Van Breugel, Evgeny S Saveliev, and Mihaela van der Schaar. 2022. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. In *International Conference on Machine Learning*, PMLR, 290–306.
- [10] Moustafa Alzantot and Mani Srivastava. 2019. Differential Privacy Synthetic Data Generation using WGANs. https://github.com/nesl/nist_differential_privacy_synthetic_data_challenge/
- [11] Seongbin An, Trang Doan, Juhee Lee, Jiwoo Kim, Yong Jae Kim, Yunji Kim, Changwon Yoon, Sungkyu Jung, Dongha Kim, Sunghoon Kwon, et al. 2023. A comparison of synthetic data approaches using utility and disclosure risk measures. *The Korean Journal of Applied Statistics* 36, 2 (2023), 141–166.
- [12] Meenatchi Sundaram Muthu Selva Annamalai, Andrea Gadotti, and Luc Rocher. 2023. A linear reconstruction approach for attribute inference attacks against synthetic data. *arXiv preprint arXiv:2301.10053* (2023).
- [13] Meenatchi Sundaram Muthu Selva Annamalai, Andrea Gadotti, and Luc Rocher. 2024. A linear reconstruction approach for attribute inference attacks against synthetic data. (2024).
- [14] Francis J Anscombe. 1973. Graphs in statistical analysis. *The american statistician* 27, 1 (1973), 17–21.
- [15] Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shihō Moriai, et al. 2017. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE transactions on information forensics and security* 13, 5 (2017), 1333–1345.

- [16] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*. PMLR, 214–223.
- [17] Hassan Jameel Asghar, Ming Ding, Thierry Rakotoarivelo, Sirine Mrabet, and Dali Kaafar. 2020. Differentially private release of datasets using Gaussian copula. *Journal of Privacy and Confidentiality* 10, 2 (2020).
- [18] Hagai Attias. 2013. Inferring parameters and structure of latent variable models by variational Bayes. *arXiv preprint arXiv:1301.6676* (2013).
- [19] Sergul Aydore, William Brown, Michael Kearns, Krishnamurthy Kenthapadi, Luca Melis, Aaron Roth, and Ankit A Siva. 2021. Differentially private query release through adaptive projection. In *International Conference on Machine Learning*. PMLR, 457–467.
- [20] Borja Balle and Yu-Xiang Wang. 2018. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*. PMLR, 394–403.
- [21] Ergute Bao, Xiaokui Xiao, Jun Zhao, Dongping Zhang, and Bolin Ding. 2021. Synthetic data generation with differential privacy via Bayesian networks. *Journal of Privacy and Confidentiality* 11, 3 (2021).
- [22] André Bauer, Simon Trapp, Michael Stenger, Robert Leppich, Samuel Kounev, Mark Leznik, Kyle Chard, and Ian Foster. 2024. Comprehensive exploration of synthetic data generation: A survey. *arXiv preprint arXiv:2401.02524* (2024).
- [23] Brett K Beaulieu-Jones, Zhiwei Steven Wu, Chris Williams, Ran Lee, Sanjeev P Bhavnani, James Brian Byrd, and Casey S Greene. 2019. Privacy-preserving generative deep neural networks support clinical data sharing. *Circulation: Cardiovascular Quality and Outcomes* 12, 7 (2019).
- [24] Kamal Berahmand, Fatemeh Daneshfar, Elaheh Sadat Salehi, Yuefeng Li, and Yue Xu. 2024. Autoencoders and their applications in machine learning: a survey. *Artificial Intelligence Review* 57, 2 (2024), 28.
- [25] Alain Berlinet and Christine Thomas-Agnan. 2011. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media.
- [26] Vincent Bindschaedler, Reza Shokri, and Carl A Gunter. 2017. Plausible deniability for privacy-preserving data synthesis. *VLDB* (2017).
- [27] Sebastian Bordt, Harsha Nori, Vanessa Rodrigues, Besmira Nushi, and Rich Caruana. 2024. Elephants Never Forget: Memorization and Learning of Tabular Data in Large Language Models. *arXiv preprint arXiv:2404.06209* (2024).
- [28] Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. 2022. Language models are realistic tabular data generators. *arXiv preprint arXiv:2210.06280* (2022).
- [29] Stavroula Bourou, Andreas El Saer, Terpsichori-Helen Velivassaki, Artemis Voulkidis, and Theodore Zahariadis. 2021. A review of tabular data synthesis using GANs on an IDS dataset. *Information* 12, 09 (2021), 375.
- [30] Claire McKay Bowen and Fang Liu. 2020. Comparative study of differentially private data synthesis methods. (2020).
- [31] Mark Bun and Thomas Steinke. 2016. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of cryptography conference*. Springer, 635–658.
- [32] Kuntai Cai, Xiaoyu Lei, Jianxin Wei, and Xiaokui Xiao. 2021. Data Synthesis via Differentially Private Markov Random Fields. *Proceedings of the VLDB Endowment* 14, 11 (2021), 2190–2202.
- [33] Kuntai Cai, Xiaokui Xiao, and Graham Cormode. 2023. Privlava: synthesizing relational data with foreign keys under differential privacy. *Proceedings of the ACM on Management of Data* 1, 2 (2023), 1–25.
- [34] Anne-Sophie Charest. 2011. How can we analyze differentially-private synthetic datasets? *Journal of Privacy and Confidentiality* 2, 2 (2011).
- [35] Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. 2020. Gs-wgan: A gradient-sanitized approach for learning differentially private generators. *Advances in Neural Information Processing Systems* 33 (2020), 12673–12684.
- [36] Qingrong Chen, Chong Xiang, Minhui Xue, Bo Li, Nikita Borisov, Dali Kaafar, and Haojin Zhu. 2018. Differentially private data generative models. *arXiv preprint arXiv:1812.02274* (2018).
- [37] Rui Chen, Qian Xiao, Yu Zhang, and Jianliang Xu. 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*. 129–138.
- [38] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. 2017. Generating multi-label discrete patient records using generative adversarial networks. In *Machine learning for healthcare conference*. PMLR, 286–305.
- [39] Adrian Corduneanu and Christopher M Bishop. 2001. Variational Bayesian model selection for mixture distributions. In *Artificial intelligence and Statistics*, Vol. 2001. Morgan Kaufmann Waltham, MA, 27–34.
- [40] Graham Cormode, Tejas Kulkarni, and Divesh Srivastava. 2018. Marginal release under local differential privacy. In *Proceedings of the 2018 International Conference on Management of Data*. 131–146.
- [41] Daniel González Cortés, Enrique Onieva, Iker Pastor López, Laura Trinchera, and Jian Wu. 2024. Autoencoder-Enhanced Clustering: A Dimensionality Reduction Approach to Financial Time Series. *IEEE Access* (2024).
- [42] Vinicius Luis Trevisan De Souza, Bruno Augusto Dorta Marques, Harlen Costa Batagelo, and João Paulo Gois. 2023. A review on generative adversarial networks for image generation. *Computers & Graphics* 114 (2023), 13–25.
- [43] Irit Dinur and Kobbi Nissim. 2003. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 202–210.
- [44] Yuntao Du and Ninghui Li. 2024. Towards principled assessment of tabular data synthesis algorithms. *arXiv preprint arXiv:2402.06806* (2024).
- [45] Dheeru Dua and Casey Graff. 2019. UCI Machine Learning Repository. <https://archive.ics.uci.edu/>
- [46] Shaoming Duan, Chuanyi Liu, Peiyi Han, Xiaopeng Jin, Xinyi Zhang, Tianyu He, Hezhong Pan, and Xiayu Xiang. 2022. HT-Fed-GAN: Federated Generative Model for Decentralized Tabular Data Synthesis. *Entropy* 25, 1 (2022), 88.

- [47] C Dwork. 2006. Differential privacy [C] *IIProc of the 33rd International Colloquium on Automata, Languages and Programming*.
- [48] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* 9, 3-4 (2014), 211–407.
- [49] Ju Fan, Tongyu Liu, Guoliang Li, Junyou Chen, Yuwei Shen, and Xiaoyong Du. 2020. Relational data synthesis using generative adversarial networks: A design space exploration. *arXiv preprint arXiv:2008.12763* (2020).
- [50] Liyue Fan and Akarsh Pokkunuru. 2021. DPNeT: Differentially private network traffic synthesis with generative adversarial networks. In *IFIP Annual Conference on Data and Applications Security and Privacy*. Springer, 3–21.
- [51] Mei Ling Fang, Devendra Singh Dhami, and Kristian Kersting. 2022. Dp-ctgan: Differentially private medical data generation using ctgans. In *International Conference on Artificial Intelligence in Medicine*. Springer, 178–188.
- [52] Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Jane Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, Christos Faloutsos, et al. 2024. Large language models (LLMs) on tabular data: Prediction, generation, and understanding-a survey. (2024).
- [53] Alvaro Figueira and Bruno Vaz. 2022. Survey on synthetic data generation, evaluation methods and GANs. *Mathematics* 10, 15 (2022), 2733.
- [54] Lorenzo Frigerio, Anderson Santana de Oliveira, Laurent Gomez, and Patrick Duverger. 2019. Differentially private generative adversarial networks for time series, continuous, and discrete open data. In *ICT Systems Security and Privacy Protection: 34th IFIP TC 11 International Conference, SEC 2019, Lisbon, Portugal, June 25-27, 2019, Proceedings 34*. Springer, 151–164.
- [55] Marco Gaboardi, Emilio Jesús Gallego Arias, Justin Hsu, Aaron Roth, and Zhiwei Steven Wu. 2014. Dual query: Practical private query release for high dimensional data. In *International Conference on Machine Learning*. PMLR, 1170–1178.
- [56] Sébastien Gambs, Frédéric Ladouceur, Antoine Laurent, and Alexandre Roy-Gaumont. 2021. Growing synthetic data through differentially-private vine copulas. *Proc. Priv. Enhancing Technol.* 2021, 3 (2021), 122–141.
- [57] Georgi Ganev, Bristena Oprisanu, and Emiliano De Cristofaro. 2022. Robin hood and matthew effects: Differential privacy has disparate impact on synthetic data. In *International Conference on Machine Learning*. PMLR, 6944–6959.
- [58] Simson Garfinkel, John M Abowd, and Christian Martindale. 2019. Understanding database reconstruction attacks on public data. *Commun. ACM* 62, 3 (2019), 46–53.
- [59] Debolina Ghatak and Kouichi Sakurai. 2022. A Survey on Privacy Preserving Synthetic Data Generation and a Discussion on a Privacy-Utility Trade-off Problem. In *International Conference on Science of Cyber Security*. Springer, 167–180.
- [60] Matteo Gioni, Franziska Boenisch, Christoph Wehmeyer, and Borbála Tasnádi. 2022. A Unified Framework for Quantifying Privacy Risk in Synthetic Data. *arXiv preprint arXiv:2211.10459* (2022).
- [61] Andre Goncalves, Priyadip Ray, Braden Soper, Jennifer Stevens, Linda Coyle, and Ana Paula Sales. 2020. Generation and evaluation of synthetic patient data. *BMC medical research methodology* 20 (2020), 1–40.
- [62] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.
- [63] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research* 13, 1 (2012), 723–773.
- [64] Frederik Harder, Kamil Adamczewski, and Mijung Park. 2021. Dp-merf: Differentially private mean embeddings with random features for practical privacy-preserving data generation. In *International conference on artificial intelligence and statistics*. PMLR, 1819–1827.
- [65] Moritz Hardt, Katrina Ligett, and Frank McSherry. 2012. A simple and practical algorithm for differentially private data release. *Advances in neural information processing systems* 25 (2012).
- [66] Moritz Hardt and Guy N Rothblum. 2010. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st annual symposium on foundations of computer science*. IEEE, 61–70.
- [67] Conor Hassan, Robert Salomone, and Kerrie Mengersen. 2023. Deep Generative Models, Synthetic Tabular Data, and Differential Privacy: An Overview and Synthesis. *arXiv preprint arXiv:2307.15424* (2023).
- [68] Justin Hsu, Aaron Roth, and Jonathan Ullman. 2013. Differential privacy for the analyst via private equilibrium computation. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*. 341–350.
- [69] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. 2022. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)* 54, 11s (2022), 1–37.
- [70] Yuzheng Hu, Fan Wu, Qinbin Li, Yunhui Long, Gonzalo Munilla Garrido, Chang Ge, Bolin Ding, David Forsyth, Bo Li, and Dawn Song. 2024. SoK: Privacy-Preserving Data Synthesis. In *2024 IEEE Symposium on Security and Privacy (SP)*. 4696–4713. <https://doi.org/10.1109/SP54263.2024.00002>
- [71] Jihyeon Hyeon, Jayoung Kim, Noseong Park, and Sushil Jajodia. 2022. An empirical study on the membership inference attack against tabular data synthesis models. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 4064–4068.
- [72] Matthew A Jaro. 1989. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *J. Amer. Statist. Assoc.* 84, 406 (1989), 414–420.
- [73] Xue Jiang, Yufei Zhang, Xuebing Zhou, and Jens Grossklags. 2023. Distributed GAN-Based Privacy-Preserving Publication of Vertically-Partitioned Data. *Proceedings on Privacy Enhancing Technologies* 2 (2023), 236–250.
- [74] Alexia Jolicoeur-Martineau, Kilian Fatras, and Tal Kachman. 2024. Generating and imputing tabular data via diffusion and flow-based gradient-boosted trees. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 1288–1296.

- [75] James Jordon, Daniel Jarrett, Evgeny Saveliev, Jinsung Yoon, Paul Elbers, Patrick Thorat, Ari Ercole, Cheng Zhang, Danielle Belgrave, and Mihaela van der Schaar. 2021. Hide-and-seek privacy challenge: Synthetic data generation vs. patient re-identification. In *NeurIPS 2020 Competition and Demonstration Track*. PMLR, 206–215.
- [76] James Jordon, Lukasz Szpruch, Florimond Houssiau, Mirko Bottarelli, Giovanni Cherubin, Carsten Maple, Samuel N Cohen, and Adrian Weller. 2022. Synthetic Data—what, why and how? *arXiv preprint arXiv:2205.03257* (2022).
- [77] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. 2018. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*.
- [78] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. 2015. The composition theorem for differential privacy. In *International conference on machine learning*. PMLR, 1376–1385.
- [79] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. 2023. Tabddpm: Modelling tabular data with diffusion models. In *International Conference on Machine Learning*. PMLR, 17564–17579.
- [80] Christopher Krapu, Robert Stewart, and Amy Rose. 2023. A review of Bayesian networks for spatial data. *ACM Transactions on Spatial Algorithms and Systems* 9, 1 (2023), 1–21.
- [81] Chaejeong Lee, Jayoung Kim, and Noseong Park. 2023. Codi: Co-evolving contrastive diffusion models for mixed-type tabular synthesis. In *International Conference on Machine Learning*. PMLR, 18940–18956.
- [82] Haoran Li, Li Xiong, and Xiaoqian Jiang. 2014. Differentially private synthesization of multi-dimensional data using copula functions. In *Advances in database technology: proceedings. International conference on extending database technology*, Vol. 2014. NIH Public Access, 475.
- [83] Haoran Li, Li Xiong, Lifan Zhang, and Xiaoqian Jiang. 2014. DPSynthesizer: differentially private data synthesizer for privacy preserving data sharing. In *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases*, Vol. 7. NIH Public Access, 1677.
- [84] Seng Pei Liew, Tsubasa Takahashi, and Michihiko Ueno. 2021. Pearl: Data synthesis via private embeddings and adversarial reconstruction learning. *ICLR* (2021).
- [85] Ji Liu, Zhenyu Weng, and Yuesheng Zhu. 2024. Precise region semantics-assisted GAN for pose-guided person image generation. *CAAI Transactions on Intelligence Technology* 9, 3 (2024), 665–678.
- [86] Tongyu Liu, Ju Fan, Guoliang Li, Nan Tang, and Xiaoyong Du. 2024. Tabular data synthesis with generative adversarial networks: design space and optimizations. *The VLDB Journal* 33, 2 (2024), 255–280.
- [87] Terrance Liu, Giuseppe Vietri, and Steven Z Wu. 2021. Iterative methods for private synthetic data: Unifying framework and new methods. *Advances in Neural Information Processing Systems* 34 (2021), 690–702.
- [88] Ziyao Liu, Jiale Guo, Mengmeng Yang, Wenzhuo Yang, Jiani Fan, and Kwok-Yan Lam. 2023. Privacy-Enhanced Knowledge Transfer with Collaborative Split Learning over Teacher Ensembles. In *Proceedings of the 2023 Secure and Trustworthy Deep Learning Systems Workshop*. 1–13.
- [89] Yunhui Long, Boxin Wang, Zhuolin Yang, Bhavya Kailkhura, Aston Zhang, Carl Gunter, and Bo Li. 2021. G-pate: Scalable differentially private data generator via private aggregation of teacher discriminators. *Advances in Neural Information Processing Systems* 34 (2021), 2965–2977.
- [90] Pei-Hsuan Lu, Pang-Chieh Wang, and Chia-Mu Yu. 2019. Empirical evaluation on synthetic data generation with generative adversarial network. In *Proceedings of the 9th International Conference on Web Intelligence, Mining and Semantics*. 1–6.
- [91] Yingzhou Lu, Huazheng Wang, and Wenqi Wei. 2023. Machine Learning for Synthetic Data Generation: a Review. *arXiv preprint arXiv:2302.04062* (2023).
- [92] Xuebin Ma, Xuejian Qi, Yulei Meng, and Tao Yang. 2023. Improved Bayesian network differential privacy data-releasing method based on junction tree. In *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 759–764.
- [93] Ciro Antonio Mami, Andrea Coser, Eric Medvet, Alexander TP Boudewijn, Marco Volpe, Michael Whitworth, Borut Svara, Gabriele Sgroi, Daniele Panfilo, and Sebastiano Saccani. 2022. Generating realistic synthetic relational data through graph variational autoencoders. *arXiv preprint arXiv:2211.16889* (2022).
- [94] Javier Marin. 2022. Evaluating Synthetically Generated Data from Small Sample Sizes: An Experimental Study. *arXiv preprint arXiv:2211.10760* (2022).
- [95] Josep Maria Mateo-Sanz, Francesc Sebé, and Josep Domingo-Ferrer. 2004. Outlier protection in continuous microdata masking. In *International Workshop on Privacy in Statistical Databases*. Springer, 201–215.
- [96] Ryan McKenna, Gerome Miklau, and Daniel Sheldon. 2021. Winning the nist contest: A scalable and general approach to differentially private synthetic data. *arXiv preprint arXiv:2108.04978* (2021).
- [97] Ryan McKenna, Brett Mullins, Daniel Sheldon, and Gerome Miklau. 2022. AIM: An Adaptive and Iterative Mechanism for Differentially Private Synthetic Data. *arXiv preprint arXiv:2201.12677* (2022).
- [98] Ryan McKenna, Daniel Sheldon, and Gerome Miklau. 2019. Graphical-model based estimation and inference for differential privacy. In *International Conference on Machine Learning*. PMLR, 4435–4444.
- [99] Ilya Mironov. 2017. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. IEEE, 263–275.
- [100] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).
- [101] Noman Mohammed, Dima Alhadidi, Benjamin CM Fung, and Mourad Debbabi. 2013. Secure two-party differentially private data release for vertically partitioned data. *IEEE transactions on dependable and secure computing* 11, 1 (2013), 59–71.

- [102] Daniel Morales, Isaac Agudo, and Javier Lopez. 2023. Private set intersection: A systematic literature review. *Computer Science Review* 49 (2023), 100567.
- [103] Roger B Nelsen. 2007. *An introduction to copulas*. Springer Science & Business Media.
- [104] Aleksandar Nikolov, Kunal Talwar, and Li Zhang. 2013. The geometry of differential privacy: the sparse and approximate cases. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*. 351–360.
- [105] Nicolas Papernot, Martín Abadi, Ulfr Erlingsson, Ian Goodfellow, and Kunal Talwar. 2017. Semi-supervised knowledge transfer for deep learning from private training data. *ICLR* (2017).
- [106] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. 2016. The synthetic data vault. In *2016 IEEE international conference on data science and advanced analytics (DSAA)*. IEEE, 399–410.
- [107] Lucas Pinheiro Cinelli, Matheus Araújo Marins, Eduardo Ant3nio Barros da Silva, and S3rgio Lima Netto. 2021. Variational autoencoder. In *Variational Methods for Machine Learning with Applications to Deep Networks*. Springer, 111–149.
- [108] Wabbeh Qardaji, Weining Yang, and Ninghui Li. 2014. Priview: practical differentially private release of marginal contingency tables. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. 1435–1446.
- [109] Timur Sattarov, Marco Schreyer, and Damian Borth. 2023. Findiff: Diffusion models for financial tabular data generation. In *Proceedings of the Fourth ACM International Conference on AI in Finance*. 64–72.
- [110] Timur Sattarov, Marco Schreyer, and Damian Borth. 2024. FedTabDiff: Federated Learning of Diffusion Probabilistic Models for Synthetic Mixed-Type Tabular Data Generation. *arXiv preprint arXiv:2401.06263* (2024).
- [111] Aditya Shankar, Hans Brouwer, Rihan Hai, and Lydia Chen. 2024. SiloFuse: Cross-silo Synthetic Data Generation with Latent Tabular Diffusion Models. *arXiv preprint arXiv:2404.03299* (2024).
- [112] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. 2018. How good is my GAN?. In *Proceedings of the European conference on computer vision (ECCV)*. 213–229.
- [113] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 3–18.
- [114] M Sklar. 1959. Fonctions de r3partition 3 n dimensions et leurs marges. In *Annales de l’ISUP*, Vol. 8. 229–231.
- [115] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. 2020. Synthetic data-A privacy mirage. *arXiv preprint arXiv:2011.07018* (2020).
- [116] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. 2022. Synthetic data–anonymisation groundhog day. In *31st USENIX Security Symposium (USENIX Security 22)*. 1451–1468.
- [117] Todd Andrew Stephenson. 2000. *An introduction to Bayesian network theory and usage*. (2000).
- [118] Sen Su, Peng Tang, Xiang Cheng, Rui Chen, and Zequn Wu. 2016. Differentially private multi-party high-dimensional data publishing. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 205–216.
- [119] Namjoon Suh, Xiaofeng Lin, Din-Yin Hsieh, Merhdad Honarkhah, and Guang Cheng. 2023. Autodiff: combining auto-encoder and diffusion model for tabular data synthesizing. *arXiv preprint arXiv:2310.15479* (2023).
- [120] Shun Takagi, Tsubasa Takahashi, Yang Cao, and Masatoshi Yoshikawa. 2021. P3gm: Private high-dimensional data release via privacy preserving phased generative model. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 169–180. <https://arxiv.org/abs/2006.12101v4>
- [121] Peng Tang, Rui Chen, Chongshi Jin, Gaoyuan Liu, and Shanjing Guo. 2022. Marginal release under multi-party personalized differential privacy. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 555–571.
- [122] Peng Tang, Xiang Cheng, Sen Su, Rui Chen, and Huaxi Shao. 2019. Differentially private publication of vertically partitioned data. *IEEE transactions on dependable and secure computing* 18, 2 (2019), 780–795.
- [123] Uthaiapon Tao Tantipongpipat, Chris Waites, Digvijay Boob, Amaresh Ankit Siva, and Rachel Cummings. 2021. Differentially private synthetic mixed-type data generation for unsupervised learning. *Intelligent Decision Technologies* 15, 4 (2021), 779–807.
- [124] Amirsina Torfi, Edward A Fox, and Chandan K Reddy. 2022. Differentially private synthetic medical data generation using convolutional GANs. *Information Sciences* 586 (2022), 485–500.
- [125] Ngoc-Trung Tran, Viet-Hung Tran, Ngoc-Bao Nguyen, Trung-Kien Nguyen, and Ngai-Man Cheung. 2021. On data augmentation for GAN training. *IEEE Transactions on Image Processing* 30 (2021), 1882–1897.
- [126] Toan V Tran and Li Xiong. 2024. Differentially Private Tabular Data Synthesis using Large Language Models. *arXiv preprint arXiv:2406.01457* (2024).
- [127] Carolina Trindade, Lu3s Antunes, T3nia Carvalho, and Nuno Moniz. 2024. Synthetic Data Outliers: Navigating Identity Disclosure. *arXiv preprint arXiv:2406.02736* (2024).
- [128] Giuseppe Vietri, Cedric Archambeau, Sergul Aydore, William Brown, Michael Kearns, Aaron Roth, Ankit Siva, Shuai Tang, and Steven Z Wu. 2022. Private synthetic data for multitask learning and marginal queries. *Advances in Neural Information Processing Systems* 35 (2022), 18282–18295.
- [129] Giuseppe Vietri, Grace Tian, Mark Bun, Thomas Steinke, and Steven Wu. 2020. New oracle-efficient algorithms for private synthetic data release. In *International Conference on Machine Learning*. PMLR, 9765–9774.
- [130] Mario Villaiz3n-Valladolid, Matteo Salvatori, Carlos Segura, and Ioannis Arapakis. 2024. Diffusion Models for Tabular Data Imputation and Synthetic Data Generation. *arXiv preprint arXiv:2407.02549* (2024).

- [131] Margarita Vinaroz, Mohammad-Amin Charusaie, Frederik Harder, Kamil Adamczewski, and Mi Jung Park. 2022. Hermite polynomial features for private data generation. In International Conference on Machine Learning. PMLR, 22300–22324.
- [132] Zhenya Wang, Xiang Cheng, Sen Su, Jintao Liang, and Haocheng Yang. 2023. ATLAS: GAN-based Differentially Private Multi-party Data Sharing. IEEE Transactions on Big Data (2023).
- [133] Qing Xiao and Shaowu Zhou. 2019. Matching a correlation coefficient by a Gaussian copula. Communications in Statistics-Theory and Methods 48, 7 (2019), 1728–1747.
- [134] Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. 2010. Differential privacy via wavelet transforms. IEEE Transactions on knowledge and data engineering 23, 8 (2010), 1200–1214.
- [135] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. 2018. Differentially private generative adversarial network. arXiv preprint arXiv:1802.06739 (2018).
- [136] Xiaodan Xing, Huanjun Wu, Lichao Wang, Iain Stenson, May Yong, Javier Del Ser, Simon Walsh, and Guang Yang. 2022. Non-Imaging Medical Data Synthesis for Trustworthy AI: A Comprehensive Survey. Comput. Surveys (2022).
- [137] Chugui Xu, Ju Ren, Deyu Zhang, Yaoxue Zhang, Zhan Qin, and Kui Ren. 2019. GANobfuscator: Mitigating information leakage under GAN via differential privacy. IEEE Transactions on Information Forensics and Security 14, 9 (2019), 2358–2371.
- [138] Chugui Xu, Ju Ren, Yaoxue Zhang, Zhan Qin, and Kui Ren. 2017. DPPro: Differentially private high-dimensional data release via random projection. IEEE Transactions on Information Forensics and Security 12, 12 (2017), 3081–3093.
- [139] Jia Xu, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, Ge Yu, and Marianne Winslett. 2013. Differentially private histogram publication. The VLDB journal 22 (2013), 797–822.
- [140] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling tabular data using conditional gan. Advances in neural information processing systems 32 (2019).
- [141] Qiantong Xu, Gao Huang, Yang Yuan, Chuan Guo, Yu Sun, Felix Wu, and Kilian Weinberger. 2018. An empirical study on evaluation metrics of generative adversarial networks. arXiv preprint arXiv:1806.07755 (2018).
- [142] Mengmeng Yang, Ming Ding, Youyang Qu, Wei Ni, David Smith, and Thierry Rakotoarivelo. 2024. Privacy at a Price: Exploring its Dual Impact on AI Fairness. arXiv preprint arXiv:2404.09391 (2024).
- [143] Mengmeng Yang, Ivan Tjuawinata, Kwok Yan Lam, Jun Zhao, and Lin Sun. 2022. Secure Hot Path Crowdsourcing With Local Differential Privacy Under Fog Computing Architecture. IEEE Transactions on Services Computing 15, 4 (2022), 2188–2201. <https://doi.org/10.1109/TSC.2020.3039336>
- [144] Mengmeng Yang, Ivan Tjuawinata, Kwok-Yan Lam, Tianqing Zhu, and Jun Zhao. 2022. Differentially Private Distributed Frequency Estimation. IEEE Transactions on Dependable and Secure Computing 20, 5 (2022), 3910–3926.
- [145] Zeyu Yang, Peikun Guo, Khadija Zanna, and Akane Sano. 2024. Balanced Mixed-Type Tabular Data Synthesis with Diffusion Models. arXiv preprint arXiv:2404.08254 (2024).
- [146] Hengrui Zhang, Jiani Zhang, Balasubramaniam Srinivasan, Zhengyuan Shen, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. 2023. Mixed-type tabular data synthesis with score-based diffusion in latent space. arXiv preprint arXiv:2310.09656 (2023).
- [147] Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. 2017. Privbayes: Private data release via bayesian networks. ACM Transactions on Database Systems (TODS) 42, 4 (2017), 1–41.
- [148] Nevin L Zhang. 2004. Hierarchical latent class models for cluster analysis. The Journal of Machine Learning Research 5 (2004), 697–723.
- [149] Xiaoyu Zhang, Chao Chen, Yi Xie, Xiaofeng Chen, Jun Zhang, and Yang Xiang. 2023. A survey on privacy inference attacks and defenses in cloud-based Deep Neural Network. Computer Standards & Interfaces 83 (2023), 103672.
- [150] Zhaoyu Zhang, Mengyan Li, and Jun Yu. 2018. On the convergence and mode collapse of GAN. In SIGGRAPH Asia 2018 Technical Briefs. 1–4.
- [151] Zhikun Zhang, Tianhao Wang, Ninghui Li, Shibo He, and Jiming Chen. 2018. CALM: Consistent adaptive local marginal for marginal release under local differential privacy. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 212–229.
- [152] Zhikun Zhang, Tianhao Wang, Ninghui Li, Jean Honorio, Michael Backes, Shibo He, Jiming Chen, and Yang Zhang. 2021. Privsyn: Differentially private data synthesis. In 30th {USENIX} Security Symposium ({USENIX} Security 21).
- [153] Ziqi Zhang, Chao Yan, and Bradley A Malin. 2022. Membership inference attacks against synthetic health data. Journal of biomedical informatics 125 (2022), 103977.
- [154] Zilong Zhao, Robert Birke, Aditya Kunar, and Lydia Y Chen. 2021. Fed-tgan: Federated learning framework for synthesizing tabular data. arXiv preprint arXiv:2108.07927 (2021).
- [155] Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y Chen. 2021. Ctab-gan: Effective table data synthesizing. In Asian Conference on Machine Learning. PMLR, 97–112.
- [156] Zilong Zhao, Han Wu, Aad Van Moorsel, and Lydia Y Chen. 2023. Gtv: Generating tabular data via vertical federated learning. arXiv preprint arXiv:2302.01706 (2023).
- [157] Keyu Zhu, Pascal Van Hentenryck, and Ferdinando Fioretto. 2021. Bias and variance of post-processing in differential privacy. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35. 11177–11184.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009