Computers and Electrical Engineering 126 (2025) 110468

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

A decision framework for privacy-preserving synthetic data generation

Pablo Sanchez-Serrano *, Ruben Rios, Isaac Agudo

NICS Lab, University of Malaga, Malaga, Spain

ARTICLE INFO

Keywords:
Synthetic data
Generative models
Privacy
Utility
Metrics
Tabular data
Taxonomy
Framework

ABSTRACT

Access to realistic data is essential for various purposes, including training machine learning models, conducting simulations, and supporting data-driven decision making across diverse domains. However, the use of real data often raises significant privacy concerns, as it may contain sensitive or personal information. Generative models have emerged as a promising solution to this problem by generating synthetic datasets that closely resemble real data. Nevertheless, these models are typically trained on original datasets, which carries the risk of leaking sensitive information. To mitigate this issue, privacy-preserving generative models have been developed to balance data utility and privacy guarantees. This paper examines existing generative models for synthetic tabular data generation, proposing a taxonomy of solutions based on the privacy guarantees they provide. Additionally, we present a decision framework to aid in selecting the most suitable privacy-preserving generative model for specific scenarios, using privacy and utility metrics as key selection criteria.

1. Introduction

In today's data-driven world, having access to large and high-quality datasets is essential, as they hold immense value for scientific, economic, and social progress. Data is the cornerstone of modern decision-making, enabling organizations to derive insights, train machine learning models, and solve complex real-world problems. However, privacy issues arise by the fact that these data are often associated with individuals or contains sensitive information. Furthermore, the growing demand for data leads to the need to share it, which presents significant challenges, particularly maximizing utility while protecting individual privacy.

When tabular data is to be shared, the most basic approach to protect privacy is to de-identify the data, that is, to remove or modify identifiers so that they cannot be linked to a specific individual. However, there are occasions when this de-identification process is insufficient [1] or done incorrectly [2]. Quasi-identifiers are those database attributes that do not uniquely identify an individual by themselves, but can be used in combination with other sources of information to reveal an individual's identity. Examples of quasi-identifiers include ZIP code, birth year, or gender. This highlights the need for a security notion that measures the level of protection against re-identification by linking information sources.

Traditional methods for protecting privacy in tabular data include [3]: data pseudonymization, which replaces Personally Identifiable Information (PII) with fake identifiers, and data anonymization, which involves generalization, suppression and perturbation techniques that modify attributes in the dataset to obtain a supposedly anonymous dataset. To decrease the risk of re-identification some privacy notions like k-anonymity, l-diversity and t-closeness have been proposed.

* Corresponding author.
E-mail addresses: pablosanserr@uma.es (P. Sanchez-Serrano), ruben.rdp@uma.es (R. Rios), isaac@uma.es (I. Agudo).

https://doi.org/10.1016/j.compeleceng.2025.110468
Received 1 December 2024; Received in revised form 24 April 2025; Accepted 20 May 2025

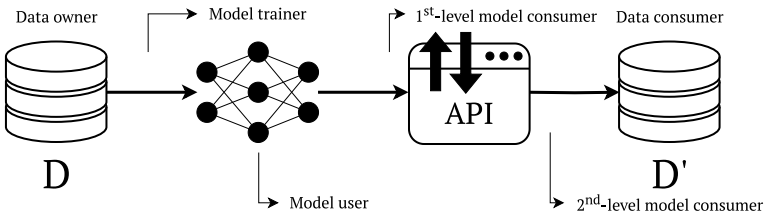


Fig. 1. Different levels of knowledge and access to the trained model.

Recently, *generative models* have emerged as a way to guarantee the privacy of datasets [4]. These models generate synthetic data from real datasets, mimicking the statistical properties of the training data. When dealing with synthetic datasets, there are significant differences in the amount of knowledge and access available to different users (see figure Fig. 1). This involves a range of privacy challenges that need to be considered. Users further to the right of the diagram show a higher level of difficulty in discerning which data were used to generate the synthetic data. The number of barriers will be higher the further to the right the user is located, i.e. the less knowledge and access the user has.

A *model trainer* uses the real data (D) given by the data owner to train a generative model. The model trainer must be careful with possible data leakage due to errors or intermediate outputs. The model trainer could also be malicious, or the data owners may not trust the data owner. Security mechanisms such as homomorphic encryption [5] or federated learning [6] should be implemented. Once the model is trained, the user can have different levels of access to the model. We refer to the user with full access to the model as the *model user*. Despite having completed the training phase, it may be possible to obtain information about D from the model [7]. Conversely, a *model consumer* can only generate samples from the model using an API, but do not have access to the trained model. The amount of information available to this type of users depends on the API. A first-level API allows unlimited samples generation, leading to honest-but-curious users who seeks information while respecting established protocols. On the other hand, a second-level API has some restrictions on data generation, i.e. limited number of requests or attributes that are not allowed to be generated. Membership Inference Attacks (MIAs) [8] can exploit the lack of restrictions on data generation. MIAs take advantage of differences in how models respond to queries from members inside and outside of the training dataset. Finally, the *data consumer* only has access to a synthetic dataset (D') generated by the model, and is unable to generate samples by himself. Although more challenging, it is possible to obtain information about D from D' [9].

Generative models are used to generate different types of data, such as images, text, audio or tabular data. This paper focuses on the latter type of data. Tabular data structured data organized in tuples or rows, where each row represents an element or individual, and each column represents an attribute of that element. Attributes can be continuous, discrete or categorical. These attributes represent different data, such as temperature, age or color.

Implementing privacy techniques can affect the quality and utility of the data. It is important to find a good balance in the trade-off between privacy and utility. This balance will depend on the type of data employed. It is crucial to have metrics for both parameters, privacy and utility. They are necessary to evaluate the quality of a generative model, to compare several models and to select the best one or the one best suited to the specific task or use.

This paper builds on our previous work [10], where we identified GANs focused on generating synthetic tabular data with privacy guarantees. The main contributions of this work are the following:

- **Creation of a taxonomy of tabular data generative models with privacy guarantees**: We developed a comprehensive taxonomy that categorizes generative models based on their underlying generative approaches and the privacy notions they adhere to, such as differential privacy. This taxonomy serves as a guideline for researchers and practitioners to identify the most appropriate generative model for specific use cases.
- **Compilation and classification of commonly used metrics for evaluating privacy and utility**: We compiled and analyzed the most widely used metrics for assessing the privacy and utility of synthetic data generated by tabular data generative models. These metrics provide a way to evaluate the effectiveness of different models in preserving privacy while ensuring that the synthetic data remains useful for its intended applications.
- **Design of a framework for model assessment and selection**: We highlighted the key challenges involved in comparing different generative models and propose a novel privacy-first framework that facilitates the assessment and selection of synthetic data generation models. The proposed framework prioritizes privacy guarantees while enabling the identification of models that provide the highest utility according to a set of predefined metrics. This structured approach simplifies the process of choosing an appropriate model for generating synthetic tabular data in privacy-sensitive scenarios.

This work is organized as follows. Section 2 describes background knowledge on generative models as well as privacy notions and techniques. Section 3 presents the taxonomy of generative models for tabular data with privacy guarantees. Section 4 details privacy and utility metrics. Section 5 provides a framework for selecting the most appropriate privacy-preserving generative model. Section 6 discusses recent studies similar to ours. Finally, Section 7 presents some conclusions.

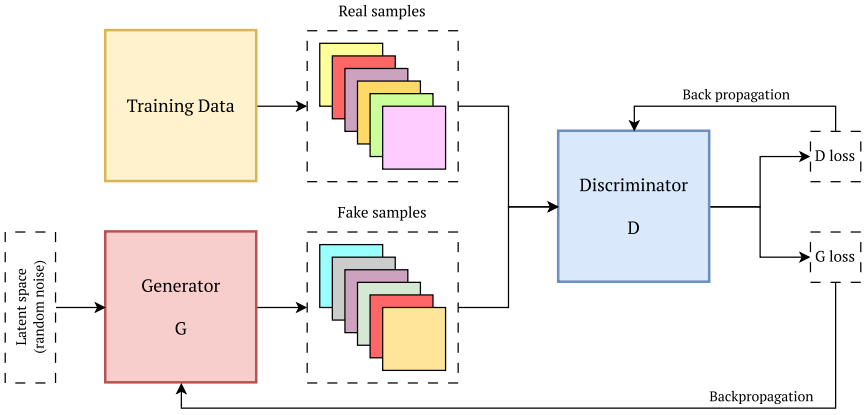


Fig. 2. Vanilla GAN block diagram. G generates samples from noise and uses them to feed D. D tries to distinguish between real and fake data. Losses are obtained from D to train G and D.

2. Background

2.1. Synthetic data generation

Generative models are methods and algorithms designed to create new data samples that mimic the patterns, properties or behavior of a given dataset. In this section we introduce relevant generative models based on various machine learning and statistical techniques.

Generative adversarial network (GAN) [11]. It is a type of machine learning framework where two neural networks are trained simultaneously in a zero-sum game setting. GANs have established themselves as one of the state-of-the-art generative models. GANs consists of two adversarial models, as shown in Fig. 2:

- Generator *G*: takes random noise as input and generates samples. It aims to generate data that imitates a given dataset.
- Discriminator *D*: attempts to differentiate between real data samples taken from the training dataset and fake data samples generated by the generator. It outputs a probability indicating whether a given sample is real or fake.

The generator tries to fool the discriminator by generating realistic data. The discriminator tries to become better at distinguishing real data from fake data. This creates a minimax game between them. The generator aims to maximize the probability of the discriminator misclassifying its outputs as real, and the discriminator aims to minimize the probability of incorrectly classifying real data as fake and vice versa.

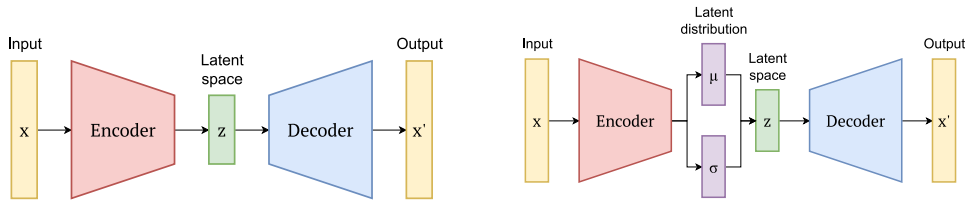
There is a wide variety of GANs, each one specialized in generating certain kinds of data, such as images, video, network traffic, tabular data, etc. In addition, over time, certain variants have emerged that improve training or the quality of the generated data, such as Wasserstein GANs (WGANs) [12], which use the Wasserstein distance to stabilize model training, Conditional GANs [13], which allow control of the data generation process, or Convolutional GANs [14], which introduce convolutional layers into their models.

In the context of tabular data generation, GANs represent the most prevalent generative model. It performs particularly well when there is a large amount of data, and it captures diversity better than other types of generative approaches. Moreover, they are proficient at capturing complex, non-linear dependencies. However, their performance is reduced when the training dataset is small. One example of a dataset where GANs would be particularly suitable is a large dataset with different types of attributes, heterogeneous distributions, and mixed data types. Typical datasets with these characteristics are EHRs, which relate various information about patients, such as personal data, symptoms, diseases, etc.

Autoencoder (AE) [15]. Unsupervised deep learning model that is trained by reconstructing its own inputs through its two main components: the encoder and the decoder. In essence, it is trained to encode the input data into a compressed representation and then decompress it. Fig. 3(a) shows the model of an autoencoder. The input data passes through the encoder and at the output is the latent space of the input data with a dimension d smaller than the input data. This latent space captures the most important features of the original data. Later, the decoder transform this compressed representation of the original data back to its original form.

Autoencoders are used for various tasks such as dimensionality reduction [16], anomaly detection [17], synthetic data generation, or denoising [18], among others.

Variational Autoencoders (VAEs) [19,20] are a type of AE specialized in the generation of synthetic data. They follow the encoder–decoder structure, but with certain modifications, as shown in Fig. 3(b). The encoder does not output a latent space, but generates a distribution over the latent space, i.e., a mean μ and a standard deviation sigma for each z dimension. The value of z is



(a) Autoencoder diagram. (b) Variational Autoencoder diagram.

Fig. 3. Autoencoder structure. The encoder compresses the input into a latent space, and the decoder reconstructs the input from that latent representation.

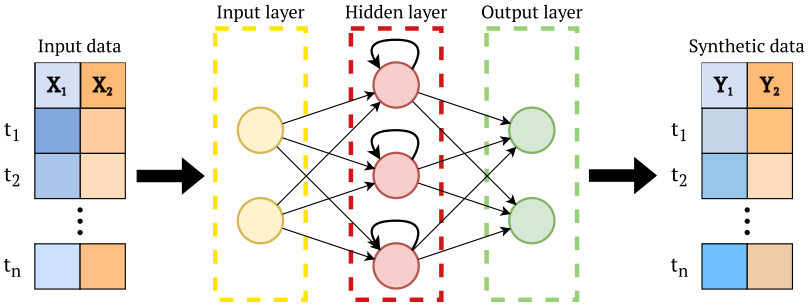


Fig. 4. RNN diagram consisting of three layers of neurons. The input data consists of n rows, each corresponding to a time point t_i of a sequence. The hidden layer receives input from both the input layer's outputs and its own previous state. This feedback mechanism enables the hidden layer to learn temporal and sequential dependencies within the data.

then obtained from this latent distribution. During training, a sample of $N(\mu, \sigma^2)$ is not taken directly, but the "reparameterization trick" is used: an auxiliary variable ϵ, which follows an $N(0, 1)$ distribution, is taken and reparameterized by the following expression:

$$z = \mu + \sigma \cdot \epsilon \quad \text{with} \quad \epsilon \sim N(0, 1) \tag{1}$$

Once the model is trained, the decoder can be used to generate synthetic data. First, the latent space is sampled with a z vector of dimension d, where each z_i value is a sample of $N(0, 1)$. The vector z is passed to the VAE decoder and a synthetic data sample is obtained.

A use case for generating synthetic data with VAE would be when one wishes to model a dataset with a high dimensionality. For example, one could use it with a dataset containing the occurrence of words or characters in emails to train an email classifier as spam or non-spam. In the event that there are a large number of words for which the occurrence is being counted, the number of attributes is very large.

Although AEs are not as widely used as generative models by themselves (in the form of VAE), they are used as auxiliary tools in other models. For example, in GANs, they can be used to reduce the dimensionality of the input data when it has many attributes.

Recurrent Neural Network (RNN) [21]. An RNN is a type of neural network designed to process sequential data due to its ability to capture temporal dependencies. This capability makes RNNs well-suited for generating synthetic data that preserve temporal patterns. Fig. 4 illustrates the structure of a simple RNN. The hidden layer feeds itself with its output from the previous iteration. As a result, RNNs can effectively model sequences where past information influences future predictions.

By maintaining a hidden state through self-recurrent connections, RNNs retain information from previous time steps, acting as a form of memory. During training, the network updates its weights using backpropagation through time (BPTT) [22], allowing it to learn and preserve past dependencies. Nevertheless, vanilla RNNs struggle to capture long-term dependencies due to the problem of vanishing gradients [23]. Long short-term memory units (LSTMs) [24] have been developed to address this problem. RNNs can be used not only independently, but also as part of another model, such as a GAN [25]. In this way, other models can be given the ability to generate reasonably realistic sequences.

To generate synthetic data, the model is first trained. Once the model is trained, an initial event is created, and then the RNN is used to generate subsequent events. The generation of events can continue until it is deemed necessary or until some sort of "termination event" occurs. An example of using RNNs to generate synthetic data is a patient's medical appointment history. The sequentiality of the data is given by the nature of this type of dataset. For instance, a patient must first have attended an appointment and then come in to receive some type of test, such as an X-ray. After training the model with actual data, a first event would be created in which a patient attends a first appointment with their doctor, and data would be generated from this event with the trained model.

Probabilistic Graphical Models (PGM) [26]. It is a representation of the structure of dependencies between random variables using graphs. A graph is a data structure consisting of a set of nodes $\mathcal{X} = X_1, \ldots, X_n$ and a set of edges ε. In a PGM, a pair of nodes

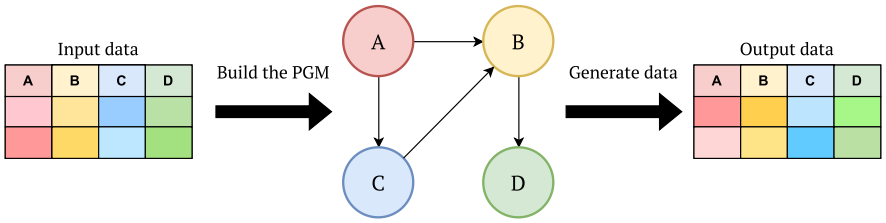


Fig. 5. PGM diagram: Each node A, B, C, and D represents an attribute of the input data, and their dependencies are represented by arrows.

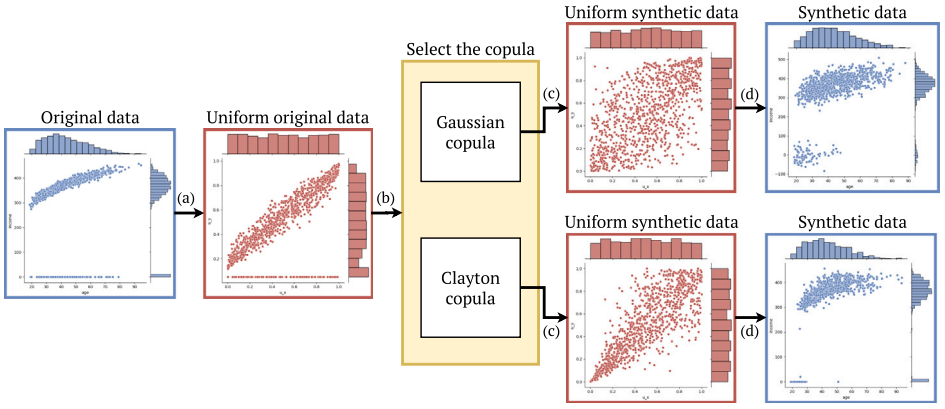


Fig. 6. Bivariate copula workflow. (a) Original data is transformed to uniform using the corresponding CDFs. (b) Select a copula that resembles the uniform transformation and fit its parameters. (c) Generate samples using the fitted copula on the uniform interval. (d) Use inverse CDFs to transform the uniform samples into the original distributions.

can be connected by a *direct edge* $X_i \rightarrow X_j$ or an *undirected edge* X_i—X_j. A graph is *directed* if all edges are *directed edges*. For each $X_i \rightarrow X_j$ from ε, X_j is said to be the *child* of X_i. A graph is *undirected* if all edges are *undirected edges*. For any X_i—X_j from ε, X_i is a *neighbor* of X_j (and vice versa). Bayesian Networks [26] are the most prevalent type of directed PGM. Similarly, Markov random fields (MRF) [27] represent the most common graphical model based on undirected graphs.

In the context of synthetic tabular data generation, each node represents one of the attributes A_i from the set $\mathcal{A} = A_1, \ldots, A_d$. Fig. 5 shows an example of PGM whose nodes represent attributes $\mathcal{A} = A, B, C, D$. The real dataset is used to construct a graph that models the joint probability distribution over the cross-product of the attribute domains in $\mathcal{A}$. Synthetic data is generated from the modeled graph, typically by sampling according to the learned distribution. This type of model aims to generate high quality data in terms of dependencies between attributes.

This type of model is particularly well-suited for the generation of data with a hierarchical structure or a clear causal structure. It is important to note, however, that this type of model is not as effective when dealing with large datasets due to inherent scalability issues [28]. A typical example that works well with PGM is a dataset containing diseases, risk factors, and symptoms, where attribute dependencies are easily identified. In such a dataset, certain risk factors may be identified as causes of diseases, which in turn might be shown to lead to certain symptoms.

Copula-based models [29] A copula is a cumulative distribution function (CDF) that models the correlation between random variables into a multivariate joint distribution. The marginal distributions are modeled independently and each has a uniform distributions on the unit interval. Copula functions are typically used to simulate correlated data.

Let $(X_1, \ldots, X_d)$ represent a vector of d-dimensional random variables, with $F_1, \ldots, F_d$ denoting the CDFs of their respective marginal distributions. By applying the CDF of each marginal to its corresponding variable, we obtain a transformed vector $(U_1, \ldots, U_d)$, where each U_i follows a uniform distribution on $[0, 1]$. In essence, the CDF transforms any distribution into the uniform distribution. Similarly, it is possible to transform a uniform distribution to any other distribution using the inverse CDF. Based on this, a copula function can be defined as follows:

Definition 1 (*Copula Function*)**.** A d-dimensional copula is a function C: $[0, 1]^d \rightarrow [0, 1]$ defined as the joint cumulative distribution function of $(U_1, \ldots, U_d)$ on the unit cube $[0, 1]^d$, where $C(u_1, \ldots, u_d) = P[U_1 \leq u_1, \ldots, U_d \leq u_d]$.

There are various types of copula functions, including the Gaussian copula, the t-Student copula, the Clayton copula or the Gumbel copula, among others. Each copula function is suited to a specific type of correlation between variables. Fig. 6 illustrates the workflow for generating synthetic correlated data using a copula function.

Note that not all the models reviewed in this section work equally well with all types of data. Certain models might perform better than others in terms of utility.

2.2. Privacy notions and techniques

In the introduction, we highlighted the need for a privacy notion that measures the level of protection against re-identification by linking information sources. L. Sweeney proposes k-anonymity [30], which guarantees that each person has at least k-1 others with the same quasi-identifiers. The formal definition of k-anonymity is as follows:

Definition 2 (*k-Anonymity*)**.** Let D be a dataset and QI_D be the quasi-identifier associated with it. D is said to satisfy k-anonymity if and only if each sequence of values in D[QI_D] appears with at least k occurrences in D[QI_D].

l-diversity [31] is an improvement on k-anonymity. In a k-anonymized dataset, an attacker can discover the values of sensitive attributes when the diversity in sensitive attributes is low. Furthermore, A. Machanavajjhala et al. [31] demonstrate that k-anonymity does not guarantee privacy against attackers who have background knowledge. l-diversity solves these problems by ensuring that, within each k-anonymized group, there are at least l different values for the sensitive attribute. This means that, even if an attacker knows to which group an individual belongs, he cannot be sure of his sensitive attribute because of the diversity of possible values.

Definition 3 (*l-Diversity*)**.** Let equivalence class be a k-anonymized group of registers k-anonymized. An equivalence class is said to be l-diverse if it contains at least l well-represented values for the sensitive attributes. A table is said to have l-diversity if every equivalence classes have l-diversity.

N. Li et al. [32] highlight a number of shortcomings of l-diversity and present t-closeness: a new principle that overcomes specific limitations of the previous ones. The idea behind t-closeness is that within each group (or equivalence class), the frequency with which each sensitive value occurs should closely resemble the overall frequency of values in the entire table. N. Li et al. defines t-closeness as follows:

Definition 4 (*t-Closeness*)**.** An equivalence class is said to have t-closeness if the distance between the distribution of a sensitive attribute in this class and the distribution of the attribute in the whole table is no more than a threshold t. A table is said to have t-closeness if all equivalence classes have t-closeness.

These privacy notions are applied to the data through generalization or aggregation techniques, as the authors show in their respective papers.

The notions explained so far focus on protecting against re-identification and attribute inference attacks. However, there are other privacy notions that need to be addressed. MIAs [8] are attacks that exploit the lack of constraints on data generation. These attacks focus on finding out whether a particular individual or element is used in the original database used to train a model. The mechanism applied to the real database may support a loss of privacy. To deal with this, the notion of differential privacy (DP) arises. DP [33] is a mathematical framework designed to provide privacy guarantees for data entries within a dataset. DP ensures that the inclusion or exclusion of a single individual's data does not significantly affect the outcome of any analysis, thereby protecting the individual's privacy.

Definition 5 (*Neighboring Datasets*)**.** Two datasets, D and D′, are neighboring, if and only if D′ differs from D in only one entry.

Definition 6 ((ε, δ)*-Differential Privacy*)**.** For a non-negative privacy budget ε and a non-negative relaxation term δ, an algorithm, M, satisfies (ε, δ)-differential privacy if for any pair of neighboring datasets D, D′ and $S \subseteq Range(M)$

$$Pr[M(D) \in S] \leq exp(\varepsilon) \cdot Pr[M(D') \in S] + \delta \tag{2}$$

where Pr is taken with respect to the randomness of M. δ is a relaxation term to ε-differential privacy. There are a variety of techniques for achieving differential privacy. Essentially, the algorithm M perturbs the input with some noise distribution, i.e. normal distribution, based on ε and δ.

The following expression is obtained by clearing ε from expression Eq. (2):

$$\varepsilon \geq ln\left(\frac{Pr[M(D) \in S] - \delta}{Pr[M(D') \in S]}\right) \tag{3}$$

A lower value of ε implies a higher level of privacy because inequality (3) is more restrictive. However, decreasing ε increases the noise that needs to be added to satisfy Definition 6.

There are some variations or extensions of the definition of differential privacy. Rényi differential privacy (RDP) [34] is a relaxation of DP based on the Rényi divergence, which is defined as follows:

Definition 7 (*Rényi Divergence*)**.** For two probability distributions P and Q defined over R, the Rényi divergence of order $\alpha > 1$ is

$$D_\alpha(P \parallel Q) \triangleq \frac{1}{\alpha - 1} \log \mathbb{E}_{x \sim Q}\left(\frac{P(x)}{Q(x)}\right)^\alpha \tag{4}$$

where $P(x)$ is the density of P at x.

Starting from the definition of Rényi divergence, the following relaxation of differential privacy is defined:

Definition 8 (*(α, ε)-Rényi Differential Privacy*)**.** A randomized mechanism $f : \mathcal{D} \mapsto \mathcal{R}$ is said to have ε-Rény differential privacy of order α, or (α, ε)-RDP, if for any neighboring datasets D, D′ $\in \mathcal{D}$ it holds that

$$D_\alpha \left(f(D) \parallel f(D') \right) \leq \varepsilon. \tag{5}$$

It should be noted that the Rényi divergence can be defined for $\alpha < 1$, including negative values. However, these values are not used in the definition of RDP.

Local Differential Privacy (LDP) [35] is a modification of DP notion. LDP arises in a setting where data owners trust no one, not even the model trainer. To achieve privacy guarantees, each of the data owners applies differential privacy to its data and then shares it.

Let $X_1, \ldots, X_n \in \mathcal{X}$ be samples from an unknown distribution P and $Z_1, \ldots, Z_n \in \mathcal{Z}$ be the privatized views of the original data. The random variables of the original and privatized data are linked by a family of conditional distributions $Q\left(Z_i \mid X_i = x, Z_j = z_j, j \neq i\right)$. This family of conditional distributions is called *channel distribution*. The formal definition of LDP is as follows:

Let $\alpha \geq 0$, Z_i is said to be an alpha-differentially locally private view of X_i if

$$\sup \frac{Q(S \mid X_i = x, Z_j = z_j, j \neq i)}{Q(S \mid X_i = x', Z_j = z_j, j \neq i)} \leq \exp(\alpha) \tag{6}$$

where the supremum is taken over $S \in \sigma(Z)$, $zj \in Z$, and $x, x \in X$; and $\sigma(Z)$ denotes an appropriate σ-field on Z.

As mentioned above, privacy techniques consist of adding noise in one way or another. In general, this noise usually follows a Laplace or Gaussian distribution. The Gaussian noise mechanism has been shown to perform better than the Laplace mechanism in machine learning models with DP guarantees [34]. Some of the techniques used to guarantee differential privacy include:

- DP-SGD [36] is a modification of the traditional Stochastic Gradient Descent (SGD) algorithm that incorporates DP to protect the privacy of the training data. The core idea behind DP-SGD is to add noise to the gradients computed during the training process. It helps prevent the leakage of sensitive information from the training data.
- Differentially Private Expectation Maximization (DP-EM) [37] is a variant of Expectation Maximization (EM) algorithm designed to maintain the privacy of individual data points during the iterative process of parameter estimation. It uses moment accountant [36] and zero-concentrated differential privacy (zCDP) [38] to bound the moment generating function of the privacy loss random variable and achieve a refined tail bound, which effectively reduces the amount of additive noise.
- Private Aggregation of Teacher Ensembles (PATE) [39,40] is a robust privacy-preserving technique for training machine learning models involving an ensemble of n teacher models and a student model. First, the set of teachers is trained on disjoint subsets of the sensitive data, D. Then, the student model is trained on public data labeled by the aggregated output of the ensemble. To protect the privacy of individuals, noise is added to the aggregated teacher output. In this way, differential privacy is guaranteed.

3. Taxonomy of generative models with privacy guarantees

In [10] we conducted a systematic literature review of the different GANs with privacy guarantees using the PICOC methodology [41]. We have extended this work to analyze also other type of models, with the aim to cover the whole landscape of generative data models: Generative Adversary Networks (GANs), Autoencoders (AEs), Probabilistic Graphical Models (PGMs), Recurrent Neural Networks (RNNs), and Copula-based models. One of the main contributions of this work is a taxonomy of tabular data generative models with privacy guarantees that can help decide with model to use for depending on each particular scenario. Fig. 7 shows this taxonomy. The white boxes represent each of the 19 models we have analyzed, while the gray boxes represent the framework or model type into which the different models fall. Those models that integrate an autoencoder as a component of their model are in the blue box. Similarly, conditional models data are in the green box. Note that DP-GAN, which is connected to the green box (conditional models box), is a particular case. Although it is possible to introduce conditions on one of its components, it does not fall within the definition of a conditional GAN [13]. Therefore, it is placed outside the category of conditional GANs.

As mentioned before, one the user has a clear understanding of the privacy notion they want to achieve with their synthetic data, this taxonomy can help one determine which models one should try to generate synthetic data from their original data. This guide would be useful for testing multiple models, as one model may perform better for your specific case than the one that might have been considered more appropriate *a priori*. Also keep in mind that in some contexts, *ad hoc* privacy notions defined for a particular scenario may give better results. In order to compare models, it is necessary to fully understand utility and privacy metrics, which are discussed in detail in the next section.

3.1. GANs

Most of the generative models for tabular data generation with privacy guarantees fall into this category. Table 1 shows the list of GANs from the taxonomy. It shows the publication year, the privacy notion and some relevant information about each model architecture. Some of them are conditional GANs and there are GANs with some convolutional layer. In addition, there are some GANs that include an autoencoder in the model. Approximately half of them use conditional GAN.

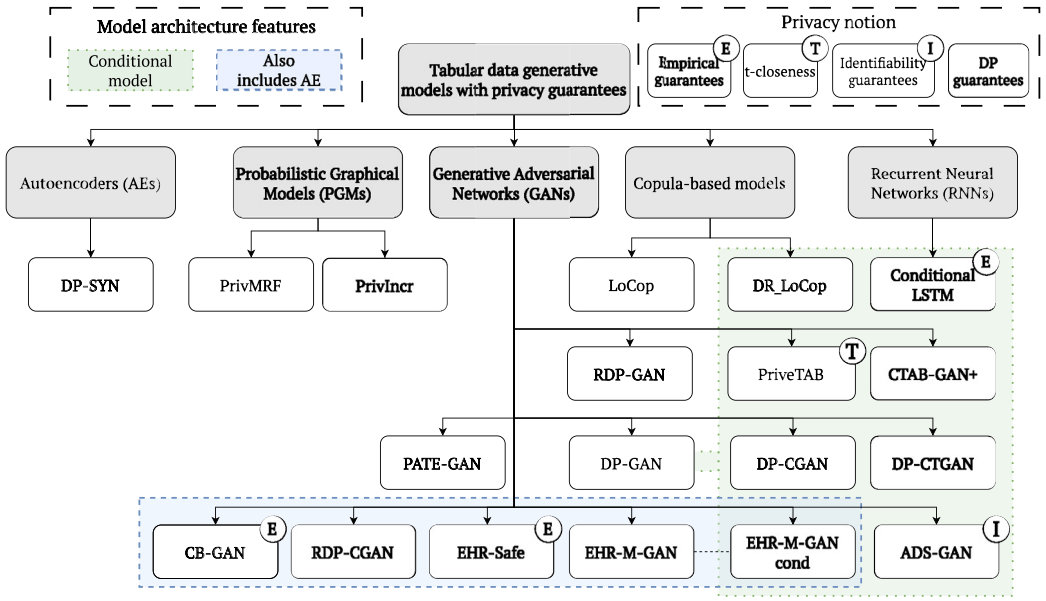


Fig. 7. Privacy-preserving tabular data generative models taxonomy.

Table 1
Comparison between GAN models.

Model	Year	Privacy notion	Cond.	Conv.	AE
PATE-GAN [42]	2019	DP	–	–	–
ADS-GAN [43]	2020	Identifiability	✓	–	–
DP-GAN [44]	2021	DP	*	–	–
RDP-CGAN [45]	2022	(R)DP	–	✓	✓
DP-CTGAN [46]	2022	DP	✓	–	–
PriveTAB [47]	2023	t-closeness	✓	–	–
DP-CGAN [48]	2023	(R)DP	✓	–	–
RDP-GAN [49]	2023	(R)DP	–	✓	–
CB-GAN [50]	2023	Empirical	–	✓	✓
EHR-M-GAN [51]	2023	DP	–	–	✓
EHR-M-GANcond [51]	2023	DP	✓	–	✓
EHR-Safe [52]	2023	Empirical	–	–	✓
CTAB-GAN+ [53]	2024	(R)DP	✓	✓	–

In a vanilla GAN there is no control over the data generation process. It generates synthetic data from the real data without allowing any further conditions or requirements. Conditional Generative Adversarial Networks (cGANs) [13] are used to address this problem. With cGANs, a condition can be included to control the data generation process. ADS-GAN is a cGAN with identifiability guarantees. Identifiability is defined in ADS-GAN paper [43] to measure and limit the risk of re-identification. PriveTAB is a cGAN with t-closeness guarantees. It generates data with CTGAN [54], a non privacy-preserving data generator, in a way that ensures t-closeness. The following cGANs provide DP guarantees: CTAB-GAN+ is an improvement on CTAB-GAN [55] and combines cGAN with convolutional layers. DP-CGAN, DP-CTGAN and EHR-M-GAN-cond are cGANs designed specifically for EHR data. The last one, EHR-M-cond, is the conditional version of EHR-M-GAN.

There are other ways to create synthetic data with privacy guarantees beyond cGANs. The following GAN models provide DP guarantees but are not conditional: EHR-M-GAN, mentioned below, uses a dual-VAE to map heterogeneous EHR data into a shared latent representations. It helps to model mixed-data type datasets. RDP-CGAN also focuses on EHR data and uses an AE in a similar way to EHR-M-GAN. This model and RDP-GAN use a convolutional network and focus specifically on RDP rather than DP. As mentioned before, DPGAN allows to introduce conditions on one of its components, it does not fall within the definition of a conditional GAN [13]. It is used for achieving privacy and not for controlling the data generation process. Lastly, PATE-GAN modifies PATE framework [39] to ensure DP [42].

There are also some models that do not theoretically guarantee privacy, but rather focus on an empirical approach to measuring privacy. In particular, CB-GAN and EHR-Safe are GANs with empirical guarantees.

Table 2
Comparison between other generative models.

Model	Year	Framework	Privacy notion	Cond.
DP-SYN [56]	2019	VAE	DP	✓
Conditional LSTM [57]	2020	RRN	Empirical	–
PrivMRF [58]	2021	PGM	DP	–
LoCop [59]	2022	Copula	(L)DP	–
Dr_LoCop [59]	2022	Copula	(L)DP	✓
PrivIncr [60]	2023	PGM	(L)DP	–

3.2. Other generative models

The taxonomy in Fig. 7 collects other types of generative models apart from GANs. These models are collected in Table 2, which shows the framework of the model, the privacy notion applied and whether or not it is a conditional generative model.

DP-SYN is a VAE with DP guarantees. Conditional LSTM is an RNN that empirically guarantees privacy by testing the effectiveness of its model against various attacks. PrivIncr and PrivMRF are PGMs that generate synthetic data with DP guarantees. PrivIncr specifically applies the concept of LDP. Finally, LoCop and its conditional version, DR_LoCop, are copula-based models that also guarantee DP. DR_LoCop also achieves dimension reduction by dividing high-dimensional attributes into several compact cliques.

4. Analysis of privacy and utility metrics

When generating tabular data, there is a clear trade-off between privacy of the training data and the utility synthetic data. The use of privacy techniques, such as reducing the level of detail in the data or adding noise, helps to protect sensitive information and the identity of individuals, but can introduce distortions in the underlying data distributions. When generating synthetic data with privacy guarantees, it is necessary to be aware of this trade-off and to know the degree of utility degradation that these techniques introduce. In order to measure the quality of the data generated by a model, it is necessary to obtain metrics that measure both the level of privacy achieved by the privacy techniques used and the utility of the data, in order to verify how much it has been degraded by the application of the privacy techniques. One of the most common applications of synthetic data generation is data augmentation for machine learning training. Therefore, it is important to have metrics to measure the performance of generative models in such tasks.

The selection of appropriate metrics is contingent upon the type of training data and the intended application. This section provides a comprehensive overview of the privacy and utility metrics most commonly employed in the models under examination. Furthermore, this section assists in identifying metrics that align closely with the requirements of the problem to be solved, adapting them to the type of data and its context.

4.1. Privacy metrics

One way to measure the privacy of a model is to measure its performance against specific attacks. The identity disclosure risk is measured by a re-identification attack [61]. This attack attempts to match the anonymized data to an original individual. One way to do this is to use external information. This is known as a linkage attack. The attribute inference attack [62] is a type of attack that focuses on inferring specific attributes of an individual. Through some attacks like this, attribute disclosure risk [63] is measured. However, the attribute disclosure risk can theoretically be measured too by metrics such as identifiability [43].

Not only is there a risk of revealing sensitive information. There is also the possibility of incorrectly attributing certain information from the data generated. This is the attribution disclosure risk [64]. Both the accurate or inaccurate attribution of attributes has the potential to cause harm [65]. The MIA [8] consists of determining whether an individual is part of the training data set used to train the generative model. This attack serves as a metric for measuring the membership disclosure risk and is one of the most important privacy metrics.

In summary, the main attacks against synthetic tabular data that can be used to empirically measure the privacy of generative models are as follows:

- **Singling out attack** [30]: Re-identification based on a single combination of quasi-identifiers.
- **Linkage attack** [30]: Re-identification through linkage of data from different datasets.
- **Background knowledge attack** [31]: Inference of sensitive data using prior knowledge of an individual or group.
- **Homogeneity attack** [31]: Attribute inference due to the lack of diversity of an anonymized group.
- **Similarity attack** [32]: Inference of sensitive information when the anonymized group has a variety of sensitive attribute values, but their values are semantically similar or numerically close.
- **Skewness attack** [32]: Attribute inference due to a skewed overall distribution of sensitive attributes.
- **Membership inference attack (MIA)** [8]: To infer whether or not an individual is a member of the training data set.

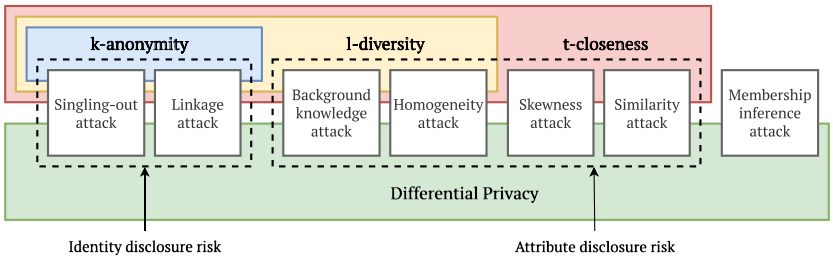


Fig. 8. Attacks on privacy and what privacy notions protect against them.

Table 3
Privacy notions parameters and the impact of their values on privacy.

Privacy notion	Parameter	Privacy impact of parameters
k-anonymity [30]	$k \in [1, n]$	The higher the value of k, the larger the group in which the individual is indistinguishable by the combination of its quasi-identifiers, so privacy is higher.
l-diversity [31]	$l \in [1, \min(k, \lvert V \rvert)]$	The higher the value of l, the greater the diversity of sensitive values, which reduces the homogeneity of the data and the chances of using background information to infer sensitive information. The value of l is limited by k and $\lvert V \rvert$, the number of possible values of the sensitive attribute
t-closeness [32]	$t \in [0, 1]$	The lower the value of t, the more similar the distribution of sensitive values in each equivalence group will be to the overall distribution, reducing the ability to make inferences in unbalanced groups.
DP [33]	$\varepsilon, \delta \in [0, \infty)$	The lower the value of ε and δ, the more noise must be added. This reduces the influence of individual data on the model parameters, which reduces the membership inference risk.

It is important to note that not all the privacy notions aims to minimize the risk against all the attacks. Fig. 8 shows the attacks from which each notion of privacy protects. Since t-closeness is an improvement over l-diversity and the l-diversity is an improvement over k-anonymity, it can be seen that if one of these notions protects against an attack, then the notion that extends it also protects against the same attack.

The application of techniques to ensure t-closeness can also serve as a theoretical measure of privacy. Since t-closeness is a technique that mitigates both identity disclosure risk and attribute disclosure risk, it can also be considered for use as a metric of these disclosure risks.

By its own definition, DP is a notion of privacy that serves as a privacy metric when a DP technique is applied. It measures the level of privacy by its ε and δ values. Similarly, RDP serves as a privacy metric through its α and ε values. MIAs, mentioned above, are the main empirical test to which generative models with differential privacy techniques are subjected.

It is important to understand what the parameters of the privacy notions represent. Table 3 lists these parameters, the possible values they can have and the impact their values have on the privacy of the generated data.

When discussing the measurement of DP, there is another widely used concept: Moments accountant. Privacy accounting concept indicates that there is a need of some "accountant" procedure that computes the privacy cost at each access to the training data, and accumulates this cost as the training progress. The strong composition theorem [66] is a general bound on the cumulative privacy loss after multiple applications of private differential mechanisms, no matter what type of noise or specific mechanism is used. It is always valid, but it tends to be conservative and therefore may overestimate the cumulative privacy loss.

Moments accountant [36] provides a tighter and more accurate estimate of cumulative privacy loss for specific cases, such as when the differential privacy mechanism is Gaussian noise aggregation (e.g., the DP-SGD algorithm). The privacy analysis of some differential privacy techniques uses the moments accountant approach to keep track of privacy costs over multiple iterations. This concept can also be used to measure privacy degradation with increasing number of queries. One way to compensate for this progressive loss of privacy would be to progressively increase the noise.

4.2. Utility metrics

Essentially, metrics that evaluate the usefulness of synthetic data can be divided into two main groups. On the one hand, there are metrics that directly compare synthetic data to training data. On the other hand, there are metrics that compare the performance of synthetic data with the performance of training data in other tasks or applications.

A robust selection of utility metrics is essential for a comprehensive assessment of the model's performance. Therefore, it is crucial to identify an appropriate set of metrics based on the characteristics of the data and the intended application.

Table 4
Categorization of utility metrics. Downward (↘) and upward (↗) arrows indicate that the metric value is better when it is lower or higher, respectively. The (=) symbol indicates that the key aspect is for the metric values of the synthetic and real data to be as similar as possible. (*) Dimension-wise probability is a combination of two metrics (RMSE ↘ and Correlation coefficient ↗).

Category	Subcategory	Metric
Statistical	Distributions comparison	MAE (↘) RMSE (↘) MMD (↘) JSD (↘) EMD (↘)
	Correlation	Pearson correlation (=) Cramer's V coefficient (=)
	Context specific	Patient trajectories (=) Dimension-wise probability (∗) Transition matrices (=) Other metrics
ML utility	Classification task	Accuracy (=) Precision (=) Recall (=) F1-score (=) AUROC (=) AUPRC (=) Agreement rate (↗)
	Regression models	MAPE (=) EVS (=) R^2 (=)

Table 4 contains a number of utility metrics of different types. This section explains the different types and classifications and provides a detailed description of each metric.

4.2.1. Statistical metrics

There are several metrics that evaluate the similarity between probability distributions. The probability mass functions (PMFs) or the probability density function (PDF) of the original data and the synthetic data can be directly compared using classical error functions such as the Mean Absolute Error (MAE) or the Root Mean Squared Error (RMSE). In addition, the Maximum Mean Discrepancy (MMD) [67] measures the difference between two distributions by calculating the mean of their measures.

Kullback–Leibler (KL) Divergence [68] also measures the difference between two probability distributions. This metric calculates the similarity of the marginal PMF for each variable independently. Thus, it calculates the similarity of two PMFs, that of the real data and that of the synthetic data. One of the problems with KL is that it is not symmetric. For example, $KL(A, B)$ is different from $KL(B, A)$, which means that it does not serve as a "distance" function. Jensen–Shannon Divergence (JSD) is a variation of KL that is symmetric, so it gives the same measure regardless of the order of the distributions being compared.

Earth Mover's Distance (EMD) [69] or Wasserstein Distance[1] is a metric that measures the "effort" required to transform one distribution into another, making it similar to the amount of earth that would have to be moved to transform one pile of earth into another. It is one of the most commonly used metrics.

The goal of generative models is not only to ensure that the distributions are as similar as possible, but also that the correlations between the different attributes are maintained. To see how well they are maintained in the synthetic data, Pearson correlation matrices are used to obtain the correlation between each pair of columns. This process of obtaining all Pearson correlations [70], pair by pair, is also called Pearson Pairwise Correlation. The most common procedure is to obtain the correlation matrix for the original and synthetic data, and to obtain the matrix resulting from the difference between the two. Sometimes Pearson correlation is used for continuous variables, while Cramer's V coefficient [71] is used for the correlation of discrete variables.

In some datasets, tabular data contains temporal information. The evolution of the data over time is a key aspect for this type of database. The autocorrelation function (ACF) measures the correlation of a time series with itself at different points in time. To evaluate how well the generative model captures these temporal dependencies, the RMSE of the ACF is calculated.

In addition to all these general statistical metrics, more specific utility metrics can be taken depending on the specific context. For example, J. Li et al. [51] propose two specific metrics for their context of EHR data generation: Patient Trajectories and dimension-wise probability. On the one hand, Patient Trajectories is a metric that evaluates the mean and standard deviation of attributes at certain time points along the data sequence. In this way, it is possible to see whether temporal changes are preserved in the synthetic data. On the other hand, to obtain the dimension-wise probability, a Bernoulli probability distribution is modeled for

[1] Specifically, EMD is equivalent to Wasserstein-1 distance. In the context of generative models, when talking about Wasserstein distance, it is usually referred to as Wasserstein-1 distance.

Table 5
Confusion matrix of a classifier.

	It is positive	It is negative
Predicted positive	TP	FP
Predicted negative	FN	TN

certain attributes and their similarities are quantified using the correlation coefficient and the RMSE. Depending on the type of generative model, it may also be interesting to use more specific metrics. For example, for an RNN, transition matrices can be compared, as well as the conditional LSTM utility evaluation [57].

4.2.2. Machine learning utility metrics

One of the main reasons for generating synthetic data is to increase the number of samples in the training dataset of other machine learning models. Therefore, it is crucial to have metrics that allow us to compare the performance of a model trained on synthetic data with the performance of the same model trained on the original data. Most of these metrics are measured in classification or regression models.

A confusion matrix is a tool for visualizing the performance of an algorithm used in supervised learning. In the case of a binary classifier, the confusion matrix has four possible values. As shown in Table 5, the value predicted by the model is compared to the actual classification of the data. A true positive (TP) or true negative (TN) is a correct classification, while a false negative (FN) or false positive (FP) is an incorrect classification.

The following three metrics can be obtained from the confusion matrix: Accuracy measures the ratio of correct predictions to total predictions made. Although less common, the concept of an misclassification rate is also employed. It is the complement of accuracy, so its value is $1 - accuracy$.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

Precision measures the ratio of true positives to total positive predictions. It is particularly important in situations where false positives are particularly damaging.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

Recall measures the ratio of true positives to the total number of true positives. It is relevant when it is important to detect all positives and reduce the number of false negatives.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{9}$$

F1-score is a balanced metric that represents the harmonic mean between precision and recall. It is employed when a balance between precision and recall is required.

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

Area Under the ROC curve (AUROC) [72] measures the ability of a model to discriminate between positive and negative classes. The ROC curve is the plot of the True Positive Rate (TPR) against the False Positive Rate (FPR). A value of AUROC close to 1 indicates good discriminative power, a value of 0.5 indicates no discriminative power (equivalent to random guessing), and a rate of less than 0.5 indicates that the classification is worse than random classification.

Area Under the Precision–Recall Curve (AUPRC) [73] is similar to AUROC, but pits precision against recall. A value close to 1 indicates that the model is very good at identifying true positives, but not many false positives. A low value indicates the opposite. AUPRC represents the performance of an algorithm better than AUROC when the dataset is unbalanced. This difference is especially relevant in GANs, where mode collapse [74] is one of the most common problems they have to deal with.

Agreement rate [75] indicates the percentage of matching predictions between two different classifiers for the same task, regardless of whether the prediction is correct. In the case of generative models, one model is trained with the original data and another model with the synthetic data to obtain this metric. This approach reveals whether the classifier trained with synthetic data has learned the same model as the classifier trained with real data.

$$\text{Agreement Rate} = \frac{\text{Number of matching predictions}}{\text{Total number of predictions}} \times 100 \tag{11}$$

In addition to the metrics related to classifiers, there are other metrics related to regression models. A regression model is a tool used to predict or explain the value of a variable based on other variables. The following metrics are to the performance of this type of model, which are commonly used to measure the utility of synthetic data generated by generative models: Mean Absolute Percentage Error (MAPE) measures the degree of accuracy and is useful when you want to interpret the error relatively, but it suffers when the values are very close to zero because the relative error becomes very large. Explained Variance Score (EVS) is a metric that measures the proportion of the total variance of the data that is explained by the model. It is useful to see which model best captures the variability of the data. Finally, R^2 represents the proportion of variance in the dependent variable of the model. $R^2 \in (-\infty, 1]$, where 1 indicates a perfect fit.

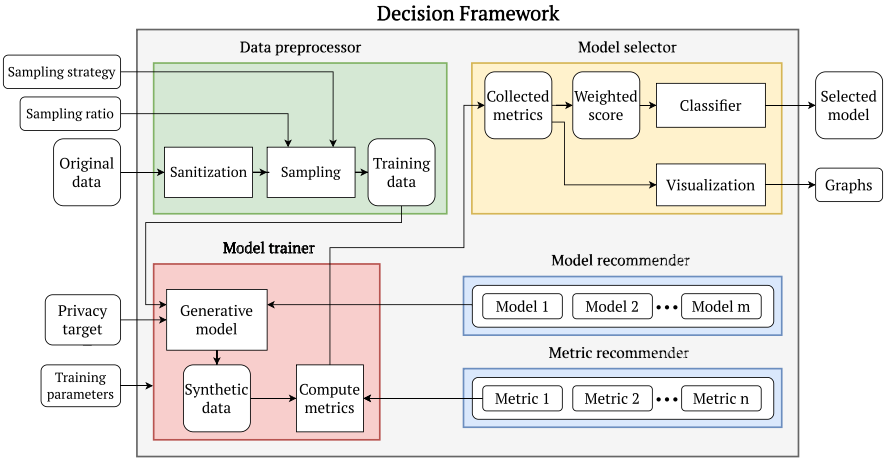


Fig. 9. Framework components diagram. The inputs are on the left and the outputs are on the right. The large colored boxes represent the different components. Rectangles with rounded corners represent variables or data, and rectangles with straight corners represent functions or processes.

5. Decision framework for privacy preserving synthetic data generation

There is a variety of generative tabular data models with privacy guarantees. The diversity in the architectures of these models, the utility and privacy evaluation metrics used, and the specific implementations make it difficult to make direct comparisons between them. In addition, each model addresses different privacy notions and are designed with different purposes in mind. For example, as shown in Section 3, some models are designed specifically for EHR data generation. Also, some models are tailored to generate high-quality mixed-type data, while others are more concerned with establishing narrower differential privacy boundaries. Thus, it is difficult to make a fair comparison between existing models. This can lead to biased interpretations of results and confusion about which model is most appropriate for particular use cases or data types.

In this section, we propose a framework for making informed selections of a generative model for a specific context. Fig. 9 presents a diagram of the framework, which is designed to assist in determining the most suitable model for a given context, rather than relying on intuition alone.

It is essential to recognize the inherent trade-off between privacy and data utility, as increasing privacy inevitably results in a reduction in utility. Therefore, to achieve an optimal balance between privacy and utility, we opt for fixing one variable and try to maximize the other. Given that ensuring a certain level of privacy is often a primary concern, we propose a privacy-first approach to address this need.

The privacy-first approach implies that one of the inputs to the framework is the privacy target to be set. Since DP is the most widely used privacy notion and provides more privacy guarantees than other traditional privacy notions such as t-closeness, it is the recommended choice but not all generative models support it. In addition to the privacy target, the system has other inputs: the original data, the training parameters, and, optionally, a sampling strategy and ratio. The framework inputs are illustrated in Fig. 9, on the left side of the diagram. In addition, the selection of metrics and experimental conditions must be in balanced — being sophisticated enough to enable meaningful quality comparisons without becoming overly intricate in a way that inadvertently favors or disadvantages certain models under specific conditions.

5.1. Models and metrics recommendation

The first step in generating synthetic data is selecting a model that accurately captures the characteristics of the original dataset, ensuring that the generated samples reflect its underlying patterns and relationships.

The analysis of the various types of models in the taxonomy from Section 3 allowed us to draw conclusions regarding their suitability for different contexts. Specifically:

- **Temporal Dependencies**: If the data exhibits sequential or temporal dependencies, an RNN is the most suitable choice, as it is specifically designed to capture such relationships.
- **Attribute Dependencies**:
  - If dependencies between attributes are important, models such as PGMs or Copulas are well-suited for capturing them.
  - If attribute dependencies are not critical, simpler or more flexible models, such as VAEs, may be considered. However, even if dependencies are not essential, models capable of capturing them should not be excluded.

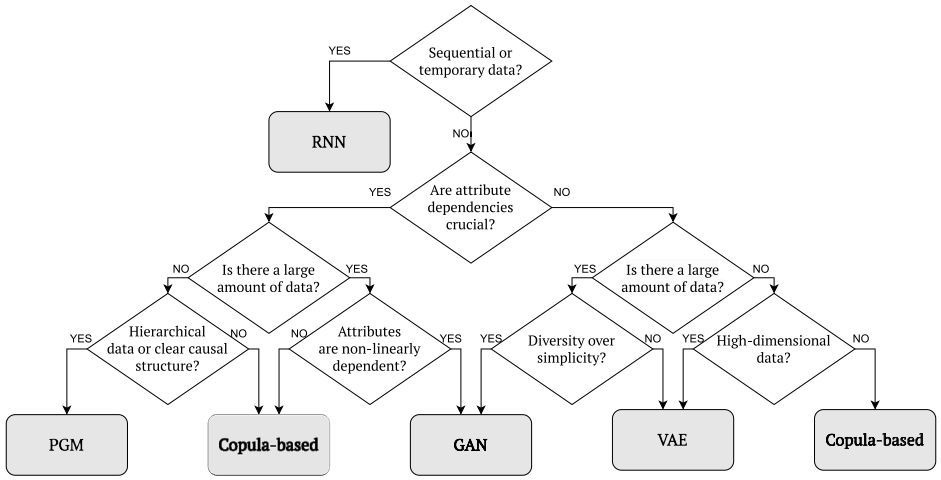


Fig. 10. Decision tree for model selection. The questions guide the identification of the model that, a priori, is most suitable for generating synthetic data that resembles the original data.

- **Data Volume**:
  - For large datasets, models like GANs, which require substantial data to learn distributions effectively, are suitable.
  - For smaller datasets, models such as copulas or PGMs, which can perform well with limited data, may be more appropriate.

Fig. 10 summarizes the conclusions from our analysis in the form of a decision flow diagram, which progresses from general to more specific questions about the dataset's characteristics. This diagram ultimately leads to the selection of the most suitable model based on the answers to these questions.

The framework includes a *model recommender* component that includes a library of generative models with privacy guarantees. This system provides an initial selection of models based on user input, the decision tree illustrated in Fig. 10 and the taxonomy of models presented in Section 3. Through a series of questions, it can identify the models that, in principle, best suit the user's requirements and data characteristics. Note that this recommendation is not restrictive; the user can still consider models that are not recommended by the decision tree, but may be viable based on the user's criteria.

Similarly, the framework includes a *metric recommender* component that contains a library of metrics that can be used to compare the models selected by the *model recommender*. Based on the data in Table 4 and user input, this component suggests a set of recommended metrics with a short description of each metric. In this way, the user can choose which metrics to include in the evaluation of the models and assign a specific weight to them for the construction of the aggregated score. As in the *model recommender*, these recommendations are indicative and do not represent a strict selection, allowing the user to make a free choice of the metrics to be used to evaluate the models.

5.2. Data preprocessing and generation

Once the models and metrics are selected, the framework initiates a preprocessing phase. This includes data sanitization and optional data sampling, which is especially useful when the amount of data is huge and training multiple models with all the data would be too costly in terms of time and/or resources. There are several types of sampling, including stratified and random [76].

Once training data is prepared, all selected models are trained with this data and the selected privacy and training parameters. Standardized experimental conditions are essential for this comparison. For a fair comparison, the models should be evaluated in the same environments to prevent software or hardware differences from introducing undesirable biases into the results. As far as possible, it should be ensured that the training parameters, such as the number of epochs, are the same. The goal of the framework is to find an appropriate balance according to the user's requirements, so no particular emphasis is placed on the computational complexity of the models. However, since some models are inherently more complex than others, limits can be set on the number of epochs or execution time in order to prevent infinite training loops. Since there are models of different types, it may not make sense to compare certain parameters, as not all of them are deep learning models, for example. The training of each model is independent, so this process can be parallelized. For each model, once the data is generated, the metrics previously selected as targets are obtained and stored in the *model selector*. Note that not only can multiple types of models be tested, but the same type can also be trained multiple times. This is especially interesting if a sampling strategy was used during data preprocessing to test the model on different subsets of the original data set.

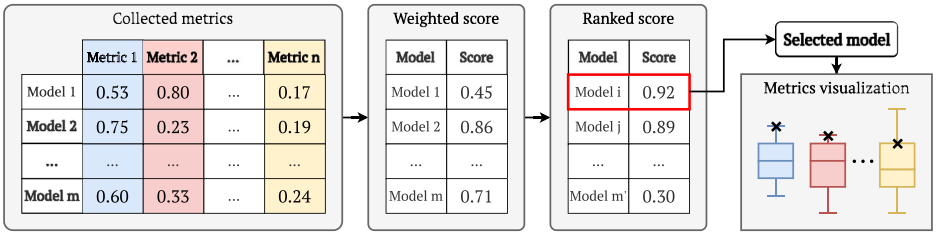


Fig. 11. *Model selector* diagram: The collected metrics are used to assign to each model an aggregated score based on user defined weights that can be used to select the best model. Additionally, the component provides visualizations to compare each model's metrics against the others.

5.3. Model selection and visualization

When the process of generating synthetic data and obtaining metrics is complete, the next step is to select the model that produced the best data. Fig. 11 shows a more detailed diagram of the *model selector*. Once the metrics of all the models have been stored, the next step is to obtain their scores through weights. The weights assigned to the metrics depend on the importance of the data characteristics for each context. For example, depending on the application to be given to the data, it may be more important to maximize the preservation of correlation or the utility in ML tasks, so the weights must be adjusted to such circumstances.

When selecting the best model, optimizing a single metric may seem like the most straightforward approach. However, in many cases, it is preferable to maximize multiple metrics simultaneously to achieve a more accurate evaluation. To this end, we introduce a weighted scoring approach, where the overall score is computed as the weighted sum of previously normalized metrics.

For example, if the primary objective is to preserve data correlations, a higher weight should be assigned to a metric that quantifies this, such as Pearson's correlation. Conversely, a metric like RMSE, which evaluates the preservation of statistical properties, could be assigned a lower weight. If machine learning utility is also relevant, an additional metric, such as R^2 or AUROC, could be assigned an intermediate weight.

A possible combination of weights in this scenario could be $w_1 = 0.6$, $w_2 = 0.1$, and $w_3 = 0.3$ for Pearson's correlation, RMSE, and R^2, respectively. Assuming these three metrics are used with the specified weights for Model 1 in Fig. 11, the overall score would be computed as:

$$score_{model1} = 0.6 \cdot 0.53 + 0.1 \cdot 0.8 + 0.3 \cdot 0.17 = 0.449$$

Once the model utility scores are obtained, they are sorted to produce a list of models ranked according to their utility. As can be seen in Fig. 11, a visualization of the obtained metrics can be made using box plot type graphs. These graphs will be particularly useful in cases where the differences between the models with the highest utility are very small and support in the graphs is needed to decide which one to use. If a sampling technique has been used, it is very important to test the model on the entire data set to ensure that the sampling strategy has resulted in a sample that accurately reflects the performance of the generative model on the entire set. Finally, the selected model can be used to generate synthetic data with privacy guarantees that mimic the original data set.

6. Related work

In recent years, there has been a growing interest in applying AI techniques to generate synthetic tabular data as a means of preserving privacy. There are still few works that provide systematic methods for comparing or selecting the most suitable mechanisms for specific use cases. In the following, we review some of the most relevant contributions in this area and compare them with our proposed approach.

SDV [77] provides a framework for computing various metrics to assess data utility, not the model itself. While this tool is valuable for obtaining a wide range of metrics and visualizations, it mists to offer a comprehensive ecosystem that assists in selecting the most suitable generative model for each specific case while ensuring privacy guarantees. Table Evaluator [78] is more recent proposal in the same line.

Yan et al. [79] propose a specialized benchmarking system for evaluating synthetic medical data. Consequently, the utility metrics included in this framework are domain-specific rather than general-purpose. In contrast, our approach incorporates general-purpose metrics while allowing for the integration of domain-specific metrics when needed. Similar to our system, their framework aggregates multiple metrics into a composite score. However, instead of computing a weighted sum of the obtained metrics, they use a weighted sum of the model's ranking positions across different metrics.

SynthEval [80] is a framework designed to evaluate both the utility and privacy of synthetic datasets. As a recent development, it addresses several limitations of previous evaluation ecosystems. Moreover, like our framework, it is easily extendable with custom metrics for specific use cases. While SynthEval serves as a tool for evaluating synthetic datasets by determining which dataset performs better based on a set of metrics, it does not compare generative models with privacy guarantees, as our framework does.

Synthcity [81] is a platform designed for comparing generative models for tabular data. Unlike SynthEval, it enables direct comparison of generative models and provides tools for training, evaluating and comparing models across different contexts.

However, although it includes some models with privacy guarantees and incorporates certain privacy metrics, it does not adopt a privacy-first approach nor does it specifically focus on generating tabular data with privacy guarantees. Additionally, rather than aggregating different metrics into a composite score for each model, it presents individual metrics, allowing users to determine the most suitable model based on whether their priority is fidelity, utility, or privacy. The paper includes tables that can help to select the models to compare, but it does not offer automated guidance or tools equivalent to the *model recommender* and *metric recommender* of our framework. This type of step-by-step or automated initial recommendation is not found in any of the papers mentioned in this section.

7. Conclusions

The need to access large amounts of realistic data for scientific, social, and economic progress often conflicts with privacy concerns. In this context, generative models emerge as a promising solution to generate synthetic tabular data while ensuring privacy guarantees. However, selecting the most appropriate model for each application presents significant challenges.

In this work we have analyzed 19 different tabular data generation models with privacy guarantees, each of them with a different way of measuring privacy and utility, and classify them in a taxonomy that can help deciding the right model for each particular scenario. The main conclusions that can be drawn from the taxonomy are: GANs are the most widely used generative model, many of the models integrate an AE into their architecture, the vast majority of models focus on guaranteeing differential privacy and it is very challenging to define a standard metric for measuring privacy and utility that works in all scenarios.

To address the challenge of comparing and selecting the most appropriate generative model for each case, we propose a privacy-first model selection framework. This framework allows the selection of a model from a predefined list that maximizes data utility according to selected utility metrics, while ensuring compliance with specified privacy requirements.

Overall, our work stands out by focusing exclusively on synthetic tabular data generation with formal privacy guarantees, adopting a privacy-first perspective. Unlike other frameworks, our study compares generative models rather than datasets, providing a model and metric recommender to guide the initial selection process based on specific use cases.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Pablo Sanchez-Serrano reports financial support was provided by Cybersecurity National Institute. Pablo Sanchez-Serrano reports a relationship with Cybersecurity National Institute that includes: employment. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This publication is part of the project “CiberIA: Investigación e Innovación para la Integración de Ciberseguridad e Inteligencia Artificial” (Proyecto C079/23), financed by “European Union NextGeneration-EU”, the Recovery Plan, Transformation and Resilience, through “INCIBE”. It has also been partially supported by the project SecAI (PID2022-139268OB-I00) funded by the “Spanish Ministerio de Ciencia e Innovacion”, and “Agencia Estatal de Investigacion.”.

Data availability

No data was used for the research described in the article.

References

[1] Narayanan A, Shmatikov V. Robust de-anonymization of large sparse datasets. In: Symposium on security and privacy. IEEE s&p, 2008, p. 111–25.
[2] Barbaro M, Zeller T, Hansell S. A Face is Exposed for AOL Searcher No. 4417749. New York Times; 2006, https://www.nytimes.com/2006/08/09/technology/09aol.html.
[3] Majeed A, Lee S. Anonymization techniques for privacy preserving data publishing: A comprehensive survey. IEEE Access 2021;9:8512–45.
[4] Figueira A, Vaz B. Survey on synthetic data generation, evaluation methods and GANs. Mathematics 2022;10(15).
[5] Armknecht F, Boyd C, Carr C, Gjøsteen K, Jäschke A, Reuter CA, Strand M. A guide to fully homomorphic encryption. 2015, Cryptology ePrint Archive, Paper 2015/1192.
[6] Zhang C, Xie Y, Bai H, Yu B, Li W, Gao Y. A survey on federated learning. Knowl-Based Syst 2021;216:106775.
[7] Zhu L, Liu Z, Han S. Deep leakage from gradients. In: Advances in neural information processing systems. Vol. 32, Curran Associates, Inc.; 2019.
[8] Shokri R, Stronati M, Song C, Shmatikov V. Membership inference attacks against machine learning models. In: Symposium on security and privacy. IEEE s&p, 2017, p. 3–18.
[9] van Breugel B, Sun H, Qian Z, van der Schaar M. Membership inference attacks against synthetic data through overfitting detection. 2023.
[10] Sanchez-Serrano P, Rios R, Agudo I. Privacy-preserving tabular data generation: Systematic literature review. In: Computer Security. ESORICS 2024 International Workshops. Cham: Springer Nature Switzerland; 2025, p. 170–80.
[11] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial networks. Commun ACM 2020;63(11):139–44.
[12] Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks. In: Precup D, Teh YW, editors. Proceedings of the 34th international conference on machine learning. Proceedings of machine learning research, vol. 70, PMLR; 2017, p. 214–23.

[13] Mirza M, Osindero S. Conditional generative adversarial nets. 2014, arXiv:1411.1784.
[14] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. 2016, arXiv:1511.06434.
[15] Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science 2006;313(5786):504–7.
[16] Wang Y, Yao H, Zhao S. Auto-encoder based dimensionality reduction. Neurocomputing 2016;184:232–42, RoLoD: Robust Local Descriptors for Computer Vision 2014.
[17] An J, Cho S. Variational autoencoder based anomaly detection using reconstruction probability. Spec Lect IE 2015;2(1):1–18.
[18] Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P-A, Bottou L. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.. J Mach Learn Res 2010;11(12).
[19] Kingma DP, Welling M. Auto-encoding variational Bayes. 2022, arXiv:1312.6114.
[20] Doersch C. Tutorial on variational autoencoders. 2021, arXiv:1606.05908.
[21] Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Phys D: Nonlinear Phenom 2020;404:132306.
[22] Werbos P. Backpropagation through time: what it does and how to do it. Proc IEEE 1990;78(10):1550–60.
[23] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 1994;5(2):157–66.
[24] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80.
[25] Mogren O. C-RNN-GAN: Continuous recurrent neural networks with adversarial training. 2016, arXiv:1611.09904.
[26] Koller D, Friedman N. Probabilistic graphical models: principles and techniques. MIT Press; 2009.
[27] Wainwright MJ, Jordan MI. Graphical models, exponential families, and variational inference. Found Trends® Mach Learn 2008;1(1–2):1–305.
[28] Chickering DM. Learning Bayesian networks is NP-complete. In: Learning from data: artificial intelligence and statistics v. New York, NY: Springer New York; 1996, p. 121–30.
[29] Nelsen RB. An introduction to copulas, 2. 2006, http://dx.doi.org/10.1007/0-387-28678-0.
[30] Sweeney L. K-anonymity: a model for protecting privacy. Internat J Uncertain Fuzziness Knowledge-Based Systems 2002;10(05):557–70.
[31] Machanavajjhala A, Kifer D, Gehrke J, Venkitasubramaniam M. L-diversity: Privacy beyond k-anonymity. ACM Trans Knowl Discov Data 2007;1(1):3–es.
[32] Li N, Li T, Venkatasubramanian S. T-closeness: Privacy beyond k-anonymity and l-diversity. In: 2007 IEEE 23rd international conference on data engineering. 2007, p. 106–15.
[33] Dwork C. Differential privacy. In: Automata, languages and programming. Berlin, Heidelberg: Springer Berlin Heidelberg; 2006, p. 1–12.
[34] Mironov I. Rényi differential privacy. In: IEEE computer security foundations symposium. CSF, 2017, p. 263–75.
[35] Duchi JC, Jordan MI, Wainwright MJ. Local privacy and statistical minimax rates. In: 2013 IEEE 54th annual symposium on foundations of computer science. 2013, p. 429–38.
[36] Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, Zhang L. Deep learning with differential privacy. In: Conference on computer and communications security. CCS, ACM; 2016, p. 308–18.
[37] Park M, Foulds J, Choudhary K, Welling M. DP-EM: Differentially Private Expectation Maximization. In: International conference on artificial intelligence and statistics. Vol. 54, PMLR; 2017, p. 896–904.
[38] Bun M, Steinke T. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In: Theory of cryptography. Berlin, Heidelberg: Springer Berlin Heidelberg; 2016, p. 635–58.
[39] Papernot N, Abadi M, Erlingsson Ú, Goodfellow I, Talwar K. Semi-supervised knowledge transfer for deep learning from private training data. 2017, arXiv:1610.05755.
[40] Papernot N, Song S, Mironov I, Raghunathan A, Talwar K, Erlingsson Ú. Scalable private learning with PATE. In: Int. conf. on learning representations. ICLR, 2018.
[41] Keele S. Guidelines for performing systematic literature reviews in software engineering. Technical report, Vol. 5, EBSE; 2007.
[42] Yoon J, Jordon J, van der Schaar M. PATE-GAN: Generating synthetic data with differential privacy guarantees. In: Int. conf. on learning representations. ICLR, 2019.
[43] Yoon J, Drumright LN, van der Schaar M. Anonymization through data synthesis using generative adversarial networks (ADS-GAN). IEEE J Biomed Heal Inform 2020;24(8):2378–88.
[44] Ho S, Qu Y, Gu B, Gao L, Li J, Xiang Y. DP-GAN: Differentially private consecutive data publishing using generative adversarial nets. J Netw Comput Appl 2021;185:103066.
[45] Torfi A, Fox EA, Reddy CK. Differentially private synthetic medical data generation using convolutional GANs. Inform Sci 2022;586:485–500.
[46] Fang ML, Dhami DS, Kersting K. DP-CTGAN: Differentially private medical data generation using CTGANs. In: Artificial intelligence in medicine. Springer; 2022, p. 178–88.
[47] Kotal A, Piplai A, Chukkapalli SSL, Joshi A. PriveTAB: Secure and privacy-preserving sharing of tabular data. In: International workshop on security and privacy analytics. IWSPA, ACM; 2022, p. 35–45.
[48] Sun C, van Soest J, Dumontier M. Generating synthetic personal health data using conditional generative adversarial networks combining with differential privacy. J Biomed Inform 2023;143:104404.
[49] Ma C, Li J, Ding M, Liu B, Wei K, Weng J, Poor HV. RDP-GAN: A rényi-differential privacy based generative adversarial network. IEEE Trans Depend Secur Comput 2023;20(6):4838–52.
[50] Hu R, Li D, Ng S-K, Zheng Z. CB-GAN: Generate sensitive data with a convolutional bidirectional generative adversarial networks. In: Database systems for advanced applications. Springer Nature; 2023, p. 159–74.
[51] Li J, Cairns BJ, Li J, Zhu T. Generating synthetic mixed-type longitudinal electronic health records for artificial intelligent applications. NPJ Digit Med 2023;6(1):98.
[52] Yoon J, Mizrahi M, Ghalaty NF, et al. EHR-Safe: generating high-fidelity and privacy-preserving synthetic electronic health records. NPJ Digit Med 2023;6(1):141.
[53] Zhao Z, Kunar A, Birke R, Van der Scheer H, Chen LY. CTAB-GAN+: enhancing tabular data synthesis. Front Big Data 2024;6.
[54] Xu L, Skoularidou M, et al. Modeling tabular data using conditional GAN. In: Adv. in neural information processing systems. Vol. 32, Curran Ass, Inc.; 2019.
[55] Zhao Z, Kunar A, Birke R, Chen LY. CTAB-GAN: Effective table data synthesizing. In: Proceedings of the 13th Asian conference on machine learning. Proceedings of machine learning research, vol. 157, PMLR; 2021, p. 97–112.
[56] Abay NC, Zhou Y, Kantarcioglu M, Thuraisingham B, Sweeney L. Privacy preserving synthetic data release using deep learning. In: Machine learning and knowledge discovery in databases. Springer; 2019, p. 510–26.
[57] Mosquera L, El Emam K, Ding L, et al. A method for generating synthetic longitudinal health data. BMC Med Res Methodol 2023;23(1):67.
[58] Cai K, Lei X, Wei J, Xiao X. Data synthesis via differentially private markov random fields. Proc VLDB Endow 2021;14(11):2190–202.
[59] Wang T, Yang X, Ren X, Yu W, Yang S. Locally private high-dimensional crowdsourced data release based on copula functions. IEEE Trans Serv Comput 2022;15(2):778–92.
[60] Liu G, Tang P, Hu C, Jin C, Guo S, Stoyanovich J, Teubner J, Mamoulis N, Pitoura E, Mühlig J. Multi-dimensional data publishing with local differential privacy.. In: EDBT. 2023, p. 183–94.
[61] Su D, Huynh HT, Chen Z, Lu Y, Lu W. Re-identification attack to privacy-preserving data analysis with noisy sample-mean. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. KDD '20, 2020, p. 1045–53.

[62] Mehnaz S, Dibbo SV, Kabir E, Li N, Bertino E. Are your sensitive attributes private? Novel model inversion attribute inference attacks on classification models. In: 31st USeNIX security symposium. USeNIX security 22, Boston, MA: USENIX Association; 2022, p. 4579–96.
[63] El Emam K, Mosquera L, Bass J. Evaluating identity disclosure risk in fully synthetic health data: Model development and validation. J Med Internet Res 2020;22(11):e23139.
[64] Taub J, Elliot M, Pampaka M, Smith D. Differential correct attribution probability for synthetic data: An exploration. In: Privacy in statistical databases. Cham: Springer International Publishing; 2018, p. 122–37.
[65] Lambert D. Measures of disclosure risk and harm. J Off Stat- Stock 1993;9. 313–313.
[66] Dwork C, Rothblum GN, Vadhan S. Boosting and differential privacy. In: 2010 IEEE 51st annual symposium on foundations of computer science. 2010, p. 51–60.
[67] Gretton A, Borgwardt K, Rasch M, Schölkopf B, Smola A. A kernel method for the two-sample-problem. In: Advances in neural information processing systems. Vol. 19, MIT Press; 2006.
[68] Hershey JR, Olsen PA. Approximating the Kullback Leibler divergence between Gaussian mixture models. In: 2007 IEEE international conference on acoustics, speech and signal processing - ICASSP '07. Vol. 4, 2007, p. IV–317–IV–320.
[69] Rubner Y, Tomasi C, Guibas L. A metric for distributions with applications to image databases. In: Sixth international conference on computer vision. IEEE cat. no.98CH36271, 1998, p. 59–66.
[70] Freedman D, Pisani R, Purves R. Statistics. International Student Ed. 4th ed.. New York: WW Norton & Company; 2007,
[71] Cramér H. Mathematical methods of statistics. Vol. 26, Princeton University Press; 1999.
[72] Bradley AP. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognit 1997;30(7):1145–59.
[73] Davis J, Goadrich M. The relationship between precision-recall and ROC curves. In: Proceedings of the 23rd international conference on machine learning. ICML '06, New York, NY, USA: Association for Computing Machinery; 2006, p. 233–40.
[74] Kossale Y, Airaj M, Darouichi A. Mode collapse in generative adversarial networks: An overview. In: 2022 8th international conference on optimization and applications. ICOA, 2022, p. 1–6.
[75] Bindschaedler V, Shokri R, Gunter CA. Plausible deniability for privacy-preserving data synthesis. 2017.
[76] Berndt AE. Sampling methods. J Hum Lact 2020;36(2):224–6.
[77] Patki N, Wedge R, Veeramachaneni K. The synthetic data vault. In: IEEE international conference on data science and advanced analytics. DSAA, 2016, p. 399–410.
[78] Brenninkmeijer B. Table evaluator. 2023, URL https://github.com/Baukebrenninkmeijer/table-evaluator, (Accessed 17 March 2025).
[79] Yan C, Yan Y, Wan Z, Zhang Z, Omberg L, Guinney J, Mooney SD, Malin BA. A multifaceted benchmarking of synthetic electronic health record generation models. Nat Commun 2022;13(1):7609.
[80] Lautrup AD, Hyrup T, Zimek A, Schneider-Kamp P. Syntheval: a framework for detailed utility and privacy evaluation of tabular synthetic data. Data Min Knowl Discov 2025;39(1):1–25.
[81] Qian Z, Cebere B-C, van der Schaar M. Synthcity: facilitating innovative use cases of synthetic data in different data modalities. 2023, arXiv:2301.07573.