Privacy-preserving Medical Data Generation using Adversarial Learning

Pronaya Prosun Das[1][0000−0003−0165−5167], Despina Tawadros[1][0009−0008−5682−5235], and Lena Wiese[1,2][0000−0003−3515−9209]

[1] Fraunhofer Institute for Toxicology and Experimental Medicine, Hannover, Germany
{pronaya.prosun.das|lena.wiese}@item.fraunhofer.de
[2] Institute of Computer Science, Goethe University Frankfurt, Frankfurt a. M., Germany

Abstract. Outstanding performance has been observed in a number of real-world applications such as speech processing and image classification using deep learning models. However, developing these kinds of models in sensitive domains such as healthcare usually necessitates dealing with a specific level of privacy challenges which provide unique concerns. For managing such privacy concerns, a practical method might involve generating feasible synthetic data that not only provides acceptable data quality but also helps to improve the efficiency of the model. Synthetic Data Generation (SDG) innately includes Generative Adversarial Networks (GANs) that have drawn significant interest in this field as a result of their achievement in various other research areas. In the study, a framework safeguarding privacy, which employs Rényi Differential Privacy along with Generative Adversarial Networks and a Variational Autoencoder (RDP-VAEGAN), is introduced. This approach is evaluated and contrasted with other top-tier models having identical privacy constraints, utilizing both unsupervised and supervised methods on two medical datasets that are publicly accessible.

Keywords: Adversarial Learning · Rényi Differential Privacy · GAN · Variational Autoencoders · Synthetic Data Generation · Healthcare · Medical data.

1 Introduction

Deep learning (DL) has shown remarkable achievements in various domains, including natural language processing, information retrieval, and computer vision, thanks to its immense capabilities. However, the effectiveness of deep learning heavily depends on having access to large volumes of training data. Consequently, incorporating deep learning models into industries that prioritize data privacy, such as healthcare, may face obstacles. In order to effectively utilize data-driven approaches in medical fields, it is crucial to address privacy concerns. Typically, personally identifiable information is anonymized to protect the sensitivity of the data. However, these methods can be vulnerable to de-anonymization attacks

[22], leading researchers to explore alternative approaches like privacy-preserving machine learning (ML) [3] to further improve a system's resilience against such attacks. In addition to this primary privacy concern, handling noisy and intricate data further adds complexity to the process, as the data may include different types, such as categorical, discrete, and continuous variables.

Synthetic Data Generation (SDG) stands out as one of the most viable methods for maintaining privacy. By generating synthetic data, privacy concerns can be alleviated, opening up numerous collaborative research opportunities. This process is particularly advantageous in tasks such as pattern identification and predictive model creation. Generative Adversarial Networks (GANs) have drawn substantial interest in the realm of SDG due to their achievements in other areas [10]. Since SDG relies on a generative process, GANs are well-suited for this purpose. Utilizing GANs makes it challenging to differentiate between real samples and those that are generated, due to the underlying distribution of the actual data. Consequently, the results of GANs cannot be reversed using a deterministic function.

On the other hand, the utilization of GANs for SDG solely does not actually ensure the privacy of the system. Depending only on the irreversibility of GANs is not sufficient, as GANs have already been proven to be vulnerable [13]. Therefore, additional steps must be considered to assure the privacy of SDG systems. The usage of private patient data in medical applications increases concerns about the severe consequences of privacy breaches. Consequently, two fundamental questions must be considered: (1) What amount of information becomes revealed during the training stage? and (2) How strong are the system's security measures? Hence, evaluating the system's privacy level is crucial to ensure its commitment to safeguarding privacy.

Differential Privacy (DP) [9] is a mathematical framework that provides a means of ensuring and quantifying the privacy of a system. It has emerged as the standard approach for exploring databases containing sensitive information. DP's strength lies in its ability to provide precise mathematical representation to ensure privacy without limiting statistical reasoning. Additionally, DP allows for the measurement of the privacy level of a system. In the domains of ML and DL, where sensitive data is often employed to enhance predictive accuracy, the role of DP is crucial. The privacy of an ML model can be compromised by various attacks, therefore it's wise to anticipate the existence of a potent adversary with a thorough understanding of the system's entire pipeline, including the model and its training process [27]. In order to shield the privacy of the system, it's imperative to defend from this type of adversary, or at the minimum, quantify the greatest possible extent of privacy intrusion in that context. A system that is entirely differentially private guarantees that the training of the algorithm does not depend on the sensitive information of any individual.

Differentially Private Stochastic Gradient Descent (DP-SGD), introduced in [1], is a widely adopted technique that ensures differential privacy while maintaining model accuracy within a given privacy budget. DP-SGD serves as the foundation for numerous research studies [31, 2]. In essence, the DP-SGD approach consists of three primary steps. Firstly, it limits the gradients to set the algorithm's sensitivity to individual data. Secondly, it introduces Gaussian noise into the data. Finally, it performs gradient descent optimization. Currently, the utilization of SGD within the DP-SGD framework is regarded as a very important technique for preserving privacy without sacrificing accuracy in ML models.

Generating synthetic data in the medical domain is faced with a number of challenges. The first challenge is ensuring privacy during the model training, which is often not directly addressed in current works, and instead, statistical or machine learning-based techniques are used. The second difficulty involves handling discrete data, a task with which GAN-based techniques often grapple, as these are designed for continuous data. The third challenge involves evaluating the quality of generated data within realistic, real-world contexts, which is particularly important in the healthcare sector. Here, inferior synthetic data can result in serious repercussions, potentially endangering human lives. The fourth challenge is the integration of local and temporal correlations among features, something that is frequently overlooked but is vital in the medical field. This is because patient histories and disease occurrences frequently display coherent patterns, and acknowledging these interdependencies can substantially enhance the reliability of synthetic data.

In this paper, we present our model that produces higher-quality synthetic data while working within similar privacy constraints. Additionally, our model provides not just high-quality synthetic data but also superior privacy safeguards.

The layout of the paper is arranged in the following manner: In Section 2, previous research on synthetic data generation, GAN, and their challenges are discussed. Section 3 presents the proposed algorithmic framework for privacy-preserving medical data generation. The experimental results are outlined in Section 4. Section 5 offers a conclusion of the research.

2 Background

2.1 Related Works

Many studies utilize Differential Privacy to generate synthetic data, often following a method described in [1]. This approach involves training a neural network while maintaining differential privacy by adding noise and using gradient clipping to restrict the norms of the gradients, in accordance with the standard procedure introduced in [9]. One of the major contributions of [1] is the introduction of the privacy accountant that monitors privacy loss. Inspired by the effectiveness of this method, we expand our privacy-preserving framework by adopting Rényi Differential Privacy (RDP) [20] as a novel notion of DP to estimate privacy loss.

Recent research has focused on tackling the challenges related to generating synthetic healthcare data [5, 11]. MedGAN is an early system used to generate synthetic medical data, which only relies on GAN models and does not implement any privacy-preserving mechanism [8]. While the method exhibits strong performance in data generation, it lacks privacy assurance, as GANs can be susceptible to attacks. In contrast, MedGAN utilizes denoising autoencoders to produce discrete data [8]. Tools like Synthea, synthetic patient generator, are not widely utilized as they depend solely on conventional specifications and do not account for factors vital to predictive analysis [30, 7]. Other research, such as CorGAN [19] and TableGAN [23], employs convolutional GANs for generating sequences of longitudinal events, while CTGAN [32] is specifically designed to handle tabular data comprising both continuous and discrete features. However, none of these approaches ensures any privacy during data generation. This lack of privacy protection makes these models vulnerable in practice and could compromise the privacy of original medical data. This paper will delve into different strategies for preserving privacy.

2.2 Differential Privacy

Differential Privacy ensures the safeguarding of individual privacy by quantifying the privacy loss that takes place when information is disclosed from a database, relying on a defined mathematical concept [9]. The most commonly used definition of Differential Privacy is (ϵ, δ)-differential privacy.

Definition 1 ((ϵ, δ)-DP). *If a randomized algorithm A that takes a dataset X as input and returns a query outcome Q satisfies the definition of (ϵ, δ)-differential privacy, then it ensures individuals' privacy for all possible query outcomes Q and all neighbouring datasets D and D'.*

$$\Pr[A(D) \in Q] \leq e^{\epsilon} \Pr[A(D') \in Q] + \delta \tag{1}$$

Datasets D and D', which differ only by one record, are referred to as neighbor datasets, highlighting the importance of maintaining individual privacy. The parameters (ϵ, δ) are used to represent the privacy budget, meaning that differentially private algorithms do not guarantee absolute privacy but only indicate the level of confidence in privacy preservation for the given (ϵ, δ) values. The smaller the values of (ϵ, δ), the more confident we are about the algorithm's privacy. The value of (ϵ, δ) with $\delta = 0$ is known as ϵ-DP, which is the original definition [9] and provides a stronger promise of privacy since even a small value of δ can lead to privacy violations due to the shift in the distribution. The use of (ϵ, δ)-DP is common because it allows for advanced composition theorem to be applied.

Theorem 1 (Advanced Composition [9]). *Suppose we apply an adaptive composition of a (ϵ, δ)-DP mechanism k times. Then, the resulting composite mechanism will be $(\epsilon', k\delta' + \delta) - DP$ with respect to δ', where the parameter ϵ' is defined as $\epsilon' = \sqrt{2k \ln\left(\frac{1}{\delta'}\right)}\epsilon + k\epsilon(e^{\epsilon} - 1)$.*

2.3 Rényi Differential Privacy

Compared to the basic composition theorem, Theorem 1, known as strong or advanced composition, establishes a more precise upper limit for the privacy loss in $(\epsilon, \delta) - \text{DP}$ compositions. However, the strong composition theorem has the drawback of rapidly increasing privacy parameters as the theorem is used repeatedly, leading to a selection of possible $(\epsilon(\delta), \delta)$ values. A novel approach called Rényi Differential Privacy (RDP) was introduced in [20] to overcome some of the constraints associated with $(\epsilon, \delta) - \text{DP}$. This approach is grounded in the idea of Rényi divergence, as detailed in Equation 2.

Definition 2 (Rényi divergence of order α [25]). *Rényi divergence of order α, which quantifies the difference between two probability distributions P and P', is specified as follows:*

$$D_\alpha(P||P') = \frac{1}{\alpha - 1} \log \left(\sum_{x \in X} \left(P(x)^\alpha P'(x)^{1-\alpha} \right) \right) \tag{2}$$

The Rényi divergence, which is a more general form of the Kullback-Leibler divergence, is equivalent to the Kullback-Leibler divergence when α is equal to 1. When α is equal to infinity, the special case is:

$$D_\infty(P||P') = \log \left(\sup_{x \in X} \frac{P(x)}{P'(x)} \right) \tag{3}$$

The value given by the logarithm of the highest ratio of probabilities for a given x over $P'(x)$ is used to calculate the Rényi divergence. The relationship between ϵ-DP and Rényi divergence is established when the value of $\alpha = \infty$. In case a randomized mechanism $\mathcal{A}$ demonstrates ϵ-differential privacy, then for a pair of datasets D and D', differing by a single record, the condition depicted in Equation 4 has to be satisfied. Using the definitions discussed earlier, the work [20] unveiled a novel concept in differential privacy known as RDP.

Definition 3 (RDP [20]). *The $(\alpha, \epsilon)-RDP$ is a randomized algorithm $\mathcal{A} : \mathcal{D} \to U$, and it is defined as satisfying the condition that for all neighbour datasets D and D', the following condition is satisfied:*

$$D_\alpha(\mathcal{A}(D)||\mathcal{A}(D')) \leq \epsilon \tag{4}$$

Two essential characteristics of the RDP definition (Definition 3) must be taken into account.

Proposition 1 (Composition of RDP [20]) *Assuming $\mathcal{A}$ is a randomized function that maps from a set $\mathcal{X}$ to a set U_1 and conforms to $(\alpha,\ \epsilon_1) - RDP$, and $\mathcal{B}$ is a randomized function that maps from $U_1 \times \mathcal{X}$ to a set U_2 and conforms to $(\alpha,\ \epsilon_2) - RDP$. Then, by applying $\mathcal{A}$ to $\mathcal{X}$, we obtain M_1, and by applying $\mathcal{B}$ to M_2 and $\mathcal{X}$, we obtain M_2. The resulting mechanism (M_1, M_2) meets the conditions of $(\alpha, \epsilon_1 + \epsilon_2)$-RDP.*

Proposition 2 *Assuming $\mathcal{A}$ is a randomized function that maps from a set $\mathcal{X}$ to a set $\mathcal{U}$ and adheres to $(\alpha,\ \epsilon) - RDP$, then it must also comply with $(\epsilon + \frac{\log(\frac{1}{\delta})}{(\alpha-1)},\ \delta) - DP$ for any value of δ that falls between 0 and 1.*

The two propositions mentioned above are fundamental to our privacy preservation approach. Proposition 1 deals with computing the privacy cost by combining the autoencoder and GAN structures. Proposition 2 is useful for evaluating the level of differential privacy in our system using the standard $(\epsilon,\ \delta)$ definition (as defined in Definition 1).

3 Algorithmic Framework

We have developed a GAN model that secures privacy through the implementation of Rényi differential privacy. GAN models have historically faced challenges when generating non-continuous data [15], prompting us to integrate autoencoders [18] to establish a continuous feature space representative of the input. This enables the GAN to produce high-quality synthetic data, simultaneously protecting privacy. No matter the type of input data, whether continuous, discrete, or a combination of both, the autoencoder can convert the input space into a continuous one. The autoencoder functions as a conduit between the non-continuous data and the GAN model which is often common in medical domains.

The framework we propose is depicted in Figure 1. In this configuration, random noise $z \in \mathbb{R}^r$, which follows a normal distribution $\mathcal{N}(0,1)$, is taken by the generator G and mapped to the generator's domain D_g^d. The discriminator D takes real data $x \in \mathbb{R}^n$ and maps it to the discriminator domain D_d, typically a set of binary values or values within the range $[-1,1]$. In our RDPVAEGAN framework, the synthetic data undergoes decoding before being inputted into the discriminator, which deviates from the standard training process used in traditional GANs. This decoding step involves guiding the artificial data through a pre-trained variational autoencoder.

3.1 GAN

Most of the studies in synthetic data generation [8] overlook the local or temporal correlations of the features. Multilayer perceptrons are commonly used, though they don't correspond well with real-life situations such as the development of diseases. To overcome this limitation, we deploy one-dimensional convolutional neural networks (CNNs) in our variational autoencoder as well as in the generator and discriminator components of our GAN architecture. CNNs are capable of recognizing patterns in correlated input features as well as capturing temporal information [12].

The procedure for training the GAN is described in Algorithm 2, referred to as the GAN Training Step. Remarkably, differential privacy is exclusively applied to the discriminator, as it is the pivotal component with access to authentic data. To effectively thwart mode collapse during training, we harness the

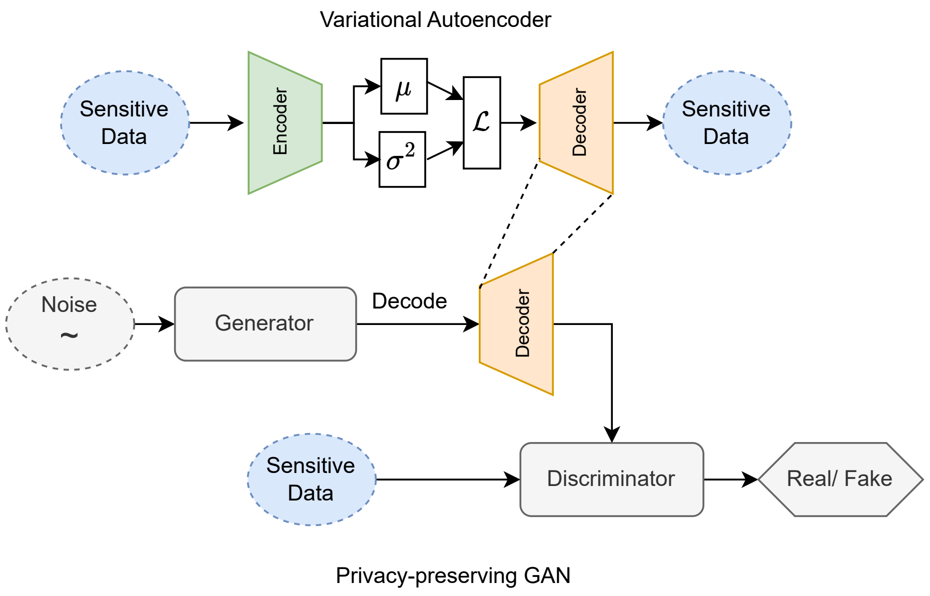

Fig. 1. The overall block diagram of GAN framework.

remarkable capabilities of the Wasserstein GAN [4], which adeptly approximates the Earth Mover's (EM) distance. The Wasserstein GAN has garnered acclaim for its unparalleled effectiveness in evading mode collapse. Distinguished as the Wasserstein-1 distance, the EM distance signifies the minimal expenditure required to seamlessly convert a synthesized data distribution $\mathbb{P}_g$ into an authentic data distribution $\mathbb{P}_x$.

$$W\left(\mathbb{P}_x,\ \mathbb{P}_g\right) = \inf_{v \epsilon \prod \mathbb{P}_x,\ \mathbb{P}_g} E_{(x,y)\sim\ v}\left[\|x-y\|\right] \tag{5}$$

Here, $\prod \mathbb{P}_x,\ \mathbb{P}_g$ represents the collection of all joint distributions $\vartheta(x,\ y)$ with marginals given by $\mathbb{P}_x$ and $\mathbb{P}_g$, respectively. The function $\vartheta(x,\ y)$ measures the amount of "mass" that needs to be shifted from x to y to effectively transform $\mathbb{P}_x$ into $\mathbb{P}_g$.

To tackle the challenging infimum in equation 5, WGAN employs an optimization strategy outlined in equation 6, inspired by the Kantorovich-Rubinstein duality [29], reflecting the intrinsic difficulty of the issue.

$$W\left(\mathbb{P}_x,\ \mathbb{P}_g\right) = \sup_{\|f\|_L \leq 1} E_{x\sim\mathbb{P}_x}\left[f(x)\right] - E_{x\sim\mathbb{P}_g}\left[f(x)\right] \tag{6}$$

To offer a more simplified explanation, the terms "supremum" and "infimum" correspond to the smallest upper bound and the largest lower bound, respectively.

Definition 4 (1-Lipschitz functions). *If we have two metric spaces* $(X,\ d_X)$ *and* $(Y,\ d_Y)$ *where 'd' is the distance metric, the function* $f:\ X\ \rightarrow\ Y$ *is known as* $K-Lipschitz$ *when:*

$$\forall(x,\ x') \in \mathbb{X},\ \exists K\in \mathbb{R} : d_Y\ (f(x),\ f(x'))\ \leq\ Kd_X(x,\ x') \tag{7}$$

With $K\ =\ 1$ *and using the distance metric, equation 7 can be expressed as:*

$$\forall(x,\ x')\ \in\ \mathbb{X}\ :\ |f(x)\ -\ f(x')| \leq\ |x\ -\ x'| \tag{8}$$

In order to compute the Wasserstein distance, it becomes imperative to discover a function that adheres to the 1-Lipschitz constraint (as specified in Definition 4 and Equation 8). To accomplish this, a neural model is constructed to acquire knowledge of the function. This entails the development of a discriminator D, which deviates from employing the Sigmoid function and instead produces a scalar output, rather than a probability of confidence.

When it comes to privacy issues, it should be highlighted that the generator is not granted direct access to the actual data, yet it is able to access the discriminator's gradients. Solely the discriminator is granted access to real data, and we propose training it while maintaining differential privacy. Fundamentally, this approach is rooted in the post-processing theorem of differential privacy, which presents the following recommended methodology.

Theorem 2 (Post-processing [9]). *Suppose we have an algorithm* $\mathbf{D}:\ \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{D}$ *that satisfies* $(\epsilon,\ \delta)-$*differential privacy, and we have an arbitrary function* $\mathbf{G}$: $\mathbb{D} \rightarrow \mathbb{O}$*. If we compose* G *with* D*, i.e.,* $\mathbf{G} \circ \mathbf{D}\ :\ \mathbb{N}^{|\mathcal{X}|}\rightarrow \mathbb{O}$*, then the resulting algorithm also satisfies* $(\epsilon,\ \delta)$*-differential privacy, as per the post-processing theorem [9].*

Drawing from the aforementioned rationale, we consider the generator and discriminator as G and D, respectively. By considering the generator as a random mapping that's layered over the discriminator, ensuring differential privacy solely on the discriminator ensures the entire system maintains differential privacy. This makes it unnecessary to train a generator that maintains privacy. During the generator's training process, as demonstrated in lines 17-20 of Algorithm 2, a private generator is unnecessary, and we can employ the conventional loss function for the generator within the WGAN framework [4].

3.2 Variational Autoencoders

The autoencoder is architected to realize multiple goals at the same time, such as detecting correlations among neighbouring features, creating a compressed feature space, converting discrete records into a continuous domain, and handling both discrete and continuous data. Autoencoders are a type of neural network design composed of an encoder and a decoder. The encoding function $Enc(\cdot)$: $\mathbb{R}^n \rightarrow \mathbb{R}^d$ is crafted to map the input $x \in \mathbb{R}^n$ into the latent space $\mathcal{L} \in \mathbb{R}^d$, equipped with weights and biases θ. Alternatively, the decoding function $Dec(\cdot)$:

Algorithm 1 Pre-Training of Variational Autoencoder

```
Require: Real dataset $X = \{x_i\}_{i=1}^{N}$, weights of the network $\theta, \phi$, learning rate $\eta$,
    number of epochs $n_{\text{vae}}$ and standard deviation of the additive noise $\sigma_{\text{vae}}$.

 2: for $k = 1 \dots n_{\text{vae}}$ do
        $Sample\ a\ mini-batch\ of\ n\ examples.\ \mathcal{X} = \{x_i\}_{i=1}^{n}$
 4:     Split X into $\mathcal{X}_1, \dots, \mathcal{X}_r$ $where\ r = \left\lfloor \frac{n}{k} \right\rfloor$
        for $l = 1 \dots r$ do
 6:         $Calculate\ Loss(\theta, \phi, \mathcal{X}_l)\ using\ Eq.$ 10.
            $g_{\theta,\phi,l} \leftarrow \nabla_{\theta,\phi} Loss(\theta, \phi\ \mathcal{X}_l)$
 8:     end for
        $\widehat{g_{\theta,\phi}} \leftarrow \frac{1}{r} \sum_{l=1}^{r} (g_{\theta,\phi,l} + \mathcal{N}(0,\ \sigma_{\text{vae}}^2))$
10:     $\widehat{\theta, \phi} = Update(\theta, \phi, \eta, \widehat{g_{\theta,\phi}})$
    end for
```

Algorithm 2 GAN Training

```
Require: Real dataset $X = \{x_i\}_{i=1}^{N}$, generator and discriminator weights $\psi$ and
    $\omega$, respectively, learning rate $\eta$, random noise $z$ where each $z_i$ follows a normal
    distribution $z_i \sim \mathcal{N}(0, 1)$, number of epochs $n_{\text{gan}}$, number of training steps for
    discriminator per one step of generator training $n_d$, norm bound $C$ and standard
    deviation of the additive noise $\sigma_{\text{gan}}$.

 2: for $j = 1 \dots n_{\text{gan}}$ do
        for $k = 1 \dots n_d$ do
 4:         $Take\ a\ mini-batch\ from\ real\ data\ \mathcal{X} = \{x_i\}_{i=1}^{n}$
            $Sample\ a\ mini-batch\ \mathcal{Z} = \{z_i\}_{i=1}^{n}$
 6:         $Partition\ real\ data\ mini-batches\ into\ \mathcal{X}_1, \dots, \mathcal{X}_r$
            $Partition\ noise\ data\ mini-batches\ into\ \mathcal{Z}_1, \dots, \mathcal{Z}_r$
 8:         for $l = 1 \dots r$ do
                $x_i \in \mathcal{X}_l$ and $z_i \in \mathcal{Z}_l$
10:             $Loss = \frac{1}{k} \sum_{i=1}^{k} (D(x_i) - D(Dec(G(z_i))))$
                $g_{\omega,l} \leftarrow \nabla_\omega Loss(\omega,\ \mathcal{X}_l)$
12:             $\widehat{g_{\omega,l}} \leftarrow \frac{g_{\omega,l}}{max(1, \frac{\|g_{\omega,l}\|_2}{C})}$
            end for
14:         $\widehat{g_\omega} \leftarrow \frac{1}{r} \sum_{l=1}^{r} (\widehat{g_{\omega,l}} + \mathcal{N}(0,\ \sigma_{\text{gan}}^2 C^2 \mathbb{I}))$
            Update: $\widehat{\omega} = \omega - \eta \widehat{g_\omega}$
16:     end for
        Sample $\{z_i\}_{i=1}^{n}$ from noise prior
18:     $Loss = -\frac{1}{n} \sum_{i}^{n} (D(Dec(G(z_i))))$
        $g_\psi \leftarrow \nabla_\psi Loss(\psi,\ \mathcal{Z})$
20:     $Update:\ \widehat{\psi} \leftarrow \psi - \eta g_\psi$
    end for
```

$\mathbb{R}^d \rightarrow \mathbb{R}^n$ seeks to reconstruct $\hat{x} \in \mathbb{R}^n$ from the latent space, also with its own set of weights and biases ϕ. The end goal is to precisely recreate the initial input data, which means that $x = \hat{x}$. Autoencoders typically utilize Mean Square Error (MSE) for handling continuous inputs and Binary Cross Entropy (BCE) for managing binary inputs. However, in this research, we apply Variational Autoencoders (VAEs) which serve as a generative model offering a method to learn complex data distributions. Unlike conventional autoencoders which can encode and then reconstruct input data, VAEs also learn a model of the data distribution, enabling the generation of new instances. Training a VAE involves managing two main tasks:

– The aim of the Encoder is to approximate the posterior $P(\mathcal{L} \mid x)$ in such a manner that $P(\mathcal{L})$ conforms to a unit Gaussian distribution.
– The goal of the Decoder is to estimate $P(x \mid \mathcal{L})$ in such a manner that it permits a highly probable reconstruction of the original input x.

$$P(x) = \int P(x \mid \mathcal{L}) P(\mathcal{L}) d\mathcal{L} \tag{9}$$

The loss function for the VAE is the negative log-likelihood with a regularizer. Since there are no shared global representations for all data points, we can decompose the loss function into only terms that depend on a single datapoint l_i. The total loss is then given by $\sum_{i=1}^{N} l_i$ for N total data points. The loss function l_i for each datapoint x_i is given by:

$$\begin{aligned} l_i(\theta, \phi) = &-\mathbb{E}_{\mathcal{L} \sim Enc(\mathcal{L} \mid x_i, \theta)}[\log Dec(x_i \mid \mathcal{L}, \phi)] \\ &+\mathcal{KL}(Enc(\mathcal{L} \mid x_i, \theta) \mid\mid Dec(\mathcal{L})) \end{aligned} \tag{10}$$

The initial part refers to the reconstruction loss, which is essentially the expected negative log-likelihood of the i-th data point. The subsequent component is a regularization term often referred to as the Kullback-Leibler divergence. This divergence measures the difference between the distribution produced by the encoder $Enc(\mathcal{L} \mid x_i)$ and the distribution generated by the decoder $Dec(\mathcal{L})$.

Algorithm 1 presents the step-by-step guide to the pre-training process of the autoencoder, encapsulating the following:

We pre-train the VAE for n_{vae} steps, where the number of steps is determined based on the desired level of privacy budget ϵ. To process a mini-batch, we split it into several micro-batches. We compute the loss (line 6) and determine the gradients (line 7) for each individual micro-batch with a size of 1. We then introduce Gaussian noise $\mathcal{N}(0, \sigma_{\text{vae}}^2)$ independently to the gradients which are calculated from micro-batch (line 9). Finally, the optimizer updates the parameters (line 10). Encoder mainly approximates mean μ and variance σ^2 from the input. Afterwards, latent space is sampled using μ and σ^2.

3.3 Model Architecture

To maximize effectiveness, we strategically employed four 1 dimensional (D) convolutional layers in both the GAN discriminator and generator, as well as in the encoder of the autoencoder. Notably, the final dense layer, pivotal for decision-making, possesses an output size of 1. Throughout the network, each layer employed the PReLU activation function [14], with the exception of the concluding layer, which operated without any activation. We employed 1-D fractionally-strided convolutions, usually referred to as transposed convolutions in the generator, following the approach described in [24]. Notably, the generator was fed with a noise of size 128 in the input, ultimately yielding an output size of 128 for optimal outcomes.

The encoder's architecture closely resembled that of the GAN discriminator, except for the omission of the last layer. In the decoder, we made use of 1-D transposed convolutional layers, arranging them in the reverse sequence of those utilized in the encoder, similar to the approach used in the generator. The encoder efficiently utilized PReLU activation functions, with the exception of the final layer, which employed Tanh activation to align with the generator's output. The decoder also adopted PReLU activation for all of its layers, with the exception of the final layer. The concluding layer used a Sigmoid activation function to restrict the output data range to be between [0, 1]. This strategic choice facilitated the reconstruction of discrete data that accurately corresponded to the input data. Importantly, the input size of the decoder was meticulously matched with the output dimension of the GAN generator.

To ensure versatility across diverse datasets, we adeptly tailored the input pipeline of both the GAN discriminator and autoencoder to seamlessly accommodate varying input sizes. Notably, these adjustments enabled smooth adaptability and compatibility. However, it is worth highlighting that no modifications were required for the GAN generator, as we consistently employed a fixed noise dimension throughout all experiments. This approach demonstrated the generator's robustness and autonomy, eliminating the need for additional adaptations.

3.4 Privacy Loss

For precise and efficient privacy loss calculations, we leveraged the advanced RDP privacy accountant [1], surpassing the computational accuracy of conventional DP methods. Proposition 2 proved instrumental in seamlessly converting RDP computations to DP, expanding the applicability of our approach. The technique of adding Gaussian noise, commonly known as the Sampled Gaussian Mechanism (SGM) [21], served as a reliable tool for introducing noise. To effectively monitor and track privacy loss, we relied upon the guidance of the following theorem, ensuring a comprehensive understanding of the privacy landscape.

Theorem 3 (Privacy loss of SGM [21]). *Assume that D and D' are two datasets that differ by only one entry, and let $\mathcal{G}$ be a Sampled Gaussian Mechanism applied to a function f with an l_2 sensitivity of one, then if we define the following:*

$$\mathcal{G}(D) \sim \varphi_1 \triangleq \mathcal{N}(0,\ \sigma^2) \tag{11}$$

$$G(D') \sim \varphi_2 \triangleq (1 - q)\mathcal{N}(0,\ \sigma^2) + q\mathcal{N}(1,\ \sigma^2) \tag{12}$$

Assuming φ_1 and φ_2 represent probability density functions, the $(\alpha,\ \epsilon)-RDP$ privacy requirement is met by G, if the following condition holds:

$$\epsilon \leq \frac{1}{\alpha\ -\ 1}\, log(max\ \{A_\alpha,\ B_\alpha\}) \tag{13}$$

Where, $A_\alpha \triangleq \mathbb{E}_{x\sim\varphi_1}[(\frac{\varphi_2}{\varphi_1})^\alpha]$ and $B_\alpha \triangleq \mathbb{E}_{x\sim\varphi_2}[(\frac{\varphi_1}{\varphi_2})^\alpha]$.

Through careful analysis, it has been firmly established that A_α is consistently less than or equal to B_α, effectively simplifying the privacy computations utilizing the RDP framework. This simplified approach strategically focuses on upper bounding A_α, streamlining the privacy calculations. By employing a combination of closed-form bounds and numerical computations, we are able to accurately determine the value of A_α [21]. It should be noted, this method sets a limit on ϵ for each individual step. Nonetheless, to ascertain the cumulative bound of ϵ for the full training procedure, we need to multiply the number of steps by ϵ, in line with what Proposition 1 delineates. For a tighter upper bound assurance, we execute experiments across a range of α values and use the RDP privacy accountant to pinpoint the smallest (ϵ) and its associated α. Finally, Proposition 2 is employed to precisely compute the $(\epsilon, \delta) - DP$, providing a comprehensive measure of privacy.

When combining autoencoder and GAN training, it is important to determine how to calculate the $(\alpha,\ \epsilon) - RDP$. Proposition 1 examines the specific input and output domains, denoted by $\mathcal{X}$, U_1, and U_2, corresponding to the autoencoder and the discriminator in the given context. Here, the autoencoder is represented by the mechanism $\mathcal{A}$, and the discriminator is represented by the mechanism $\mathcal{B}$. U_1 becomes the output space after the decoder processes the fake samples. Subsequently, the discriminator observes U_1, while the real input space is represented by $\mathcal{X}$. With α held constant, Proposition 1 suggests that the entire system maintains $(\alpha,\ \epsilon_{vae} +\ \epsilon_{gan}) - RDP$. Nevertheless, it's not assured that there can be a constant value for α, since the RDP contains a budget constraint that is specified by this variable. To determine a fixed α, we use the following procedure: Consider two systems, S_1 one for the autoencoder with $(\alpha_1,\ \epsilon_1) - RDP$ and S_2 for the GAN with $(\alpha_2,\ \epsilon_2) - RDP$. Assuming $\epsilon_1 \leq \epsilon_2$ without any loss of generality, we can choose $\alpha_{total} = \alpha_2$. This results in S_2 having $(\alpha_{total},\ \epsilon_2) - RDP$. For the system S_1, we choose $\alpha_{total} = \alpha_1$ and compute ϵ' in a way that ensures $\epsilon \leq \epsilon'$. The entire system, comprised of S_1 and S_2, subsequently meets the $(\alpha_{total},\ \epsilon_2 +\ \epsilon') - RDP$ condition.

4 Experimental Evaluation

In this section, we outline the specifics of the experimental design, convey the results procured from multiple experiments, and perform a comparative analysis with multiple approaches documented in the existing research literature.

The dataset is partitioned into two segments.: D_{tr} for training and D_{te} for testing. We employ D_{tr} for model training, subsequently using these trained models to generate synthesized samples, denoted as D_{syn}. As elaborated in Section 3, although the structures may change depending on the size of the dataset, the dimensions of the latent space (also serving as the input space for the decoder) sampled from encoder's output (mean μ and variance σ^2) and the output space of the generator remain unchanged. Likewise, the dimensions of the input space for the encoder, the output space for the decoder, and the input space for the discriminator also remain the same. It's essential to highlight that all stated ϵ values correspond to the definition of $(\epsilon, \delta) - DP$ where $\delta = 10^{-5}$, except when explicitly stated otherwise.

We trained both the Convolutional GAN and the Convolutional Variational Autoencoder with a mini-batch size of 64, using the Adam optimizer [17], and applying a learning rate of 0.005. To enhance the training process, we employed Batch Normalization (BN) [16] for both the discriminator and the generator. A single GeForce RTX 3080 NVIDIA GPU was utilized for our experimental work.

4.1 Datasets

In our research, we utilized two datasets to assess the effectiveness of our model.

- **Early Prediction of Sepsis from Clinical Data (Sepsis) [26]**: The first dataset used in this study was obtained from the publicly available 2019 PhysioNet Challenge. In accordance with the Sepsis III guidelines, the diagnosis of sepsis relies on the identification of a two-point increase in the patient's Sequential Organ Failure Assessment (SOFA) score, along with the presence of clinical signs and symptoms indicating a potential infection. The dataset consists of approximately 40,000 electronic health records of patients admitted to the ICU with suspected or confirmed sepsis. The data was collected from Beth Israel Deaconess Medical Center and Emory University Hospital. Each patient record comprises a time series of clinical measurements taken at 1-hour intervals with a binary label indicating the presence or absence of sepsis at each time point, thereby represented by a single row. The clinical measurements are a set of 40 numerical and categorical attributes that encompass patient demographics, vital signs and laboratory values. Notably, the dataset contains a total of 2,932 septic and 37,404 non-septic patients.
- **Cardiovascular Diseases Dataset [28]**: The second dataset utilized in this study was retrieved from Kaggle. The dataset includes approximately 70,000 records of patients' data with 12 attributes encompassing both numerical and categorical types. The dataset is primarily composed of three types of input features, namely objective features representing patient demographics, results of medical examination conducted at that time and subjective information provided by the patients themselves. The target feature is denoted by a binary value that signifies the presence or absence of cardiovascular disease. Additionally, the dataset demonstrates a well-balanced distribution of male and female records. Similarly, the target feature exhibits a balanced spread of binary values, enhancing the dataset's reliability.

4.2 Comparison

Our model is juxtaposed with two benchmark methods for comparison. The choice of benchmark models hinges on the specific characteristics inherent to the experiments being performed.

DPGAN [31]: This model is capable of generating superior quality data points while maintaining adequate differential privacy safeguards. This is accomplished by injecting carefully engineered noise into the gradients during the learning phase. For this research, we employed an open-source edition of DPGAN.

MedGAN [8]: Structured to generate discrete entries, such as those required for unsupervised synthetic data creation, MedGAN's architecture incorporates an autoencoder and a traditional GAN. However, it does not ensure data privacy. For our experimentation, we used an open-source version of MedGAN.

4.3 Synthetic Data Generation

We opted for electronic health records, which were transformed into a high-dimensional dataset, to illustrate the privacy-protection abilities of our suggested model. In this situation, the data features are comprised of two types: numerical and multi-label. Multi-label features (such as gender categories) are encoded using one-hot encoding. Our aim is to harness the insight relating to a private dataset and utilize it to generate a synthetic dataset that have a comparable distribution while maintaining a commitment to privacy. At first, we aligned our method with various models that neither require labeled data nor rely on it, and are tailored for creating synthetic data in an unsupervised manner, all while continuing to adhere to the principles of privacy preservation. We employed Eq. 10 as the loss function for pretraining the variational autoencoder.

In order to gauge the quality of the artificially generated data in an unsupervised setting, we employed two evaluative metrics:

- Maximum Mean Discrepancy (MMD): This metric shows the magnitude to which the model replicates the statistical distribution acquired from the actual data. Recently, a work [33] underscored that MMD encapsulates many desirable characteristics of an evaluative metric, especially pertinent to GANs. MMD is typically employed in an unsupervised environment due to the absence of labelled data for statistical quantifications. To record MMD, we drew comparisons between two sample sets of real and synthetic data, each comprising 800 entries. Figure 2 shows the comparison of MMD scores obtain from generated and real data distributions from different datasets. A lower MMD score suggests a greater similarity between the synthetic and real data distributions, suggesting a more effective model.
- Dimension-wise Prediction: This evaluation method elucidates the interdependencies among features, in other words, it assesses the model's capability to predict absent features by using the features already present in the dataset. Let's suppose that from D_{tr}, we derive D_{syn}. We then randomly pick one specific dimension (denoted as k) from both D_{syn} and D_{tr}, labeling them as $D_{syn,k}$ and $D_{tr,k}$ respectively. This chosen dimension is what

we refer to as the testing dimension. All the remaining dimensions, namely $D_{syn,\setminus k}$ and $D_{tr,\setminus k}$, serve to train a classifier. The classifier's objective is to anticipate the test set's testing dimension value, represented by $D_{te,k}$. For prediction tasks, the Random Forests algorithm [6] was used. The efficiency of the holistic predictive models (those trained on synthetic data) and across all attributes are presented in terms of the F1-score in Table 1.

From Table 1, it's evident that our proposed model (RDPVAEGAN) has a superior performance in capturing correlated features compared to DPGAN and MedGAN across both datasets. The nearer the outcomes align with real data experiments, the superior the quality of the synthetic data, indicating a more effective model. However, it's worth noting that only RDPVAEGAN and DPGAN provide privacy guarantees as previously mentioned.

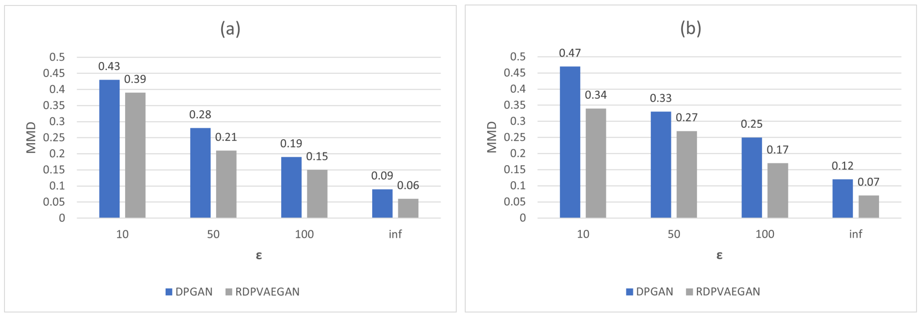

Fig. 2. The analysis of the differences and similarities between the real and the generated distributions. A lower MMD score means a greater similarity between the data distributions. (a) for the Sepsis dataset and (b) for the Cardiovascular Diseases dataset.

Table 1. The table illustrates a comparison of different techniques utilizing the F-1 score within the dimension-wise prediction setting. Except for the column labelled "Real Data", classifiers are trained using synthetic data.

Dataset	Real Data	DPGAN	MedGAN	RDPVAEGAN
Sepsis	0.53	0.39	0.43	0.45
Cardiovascular Disease	0.49	0.27	0.31	0.38

We've extended our approach to accommodate a supervised environment and create labelled synthetic data as well. All the experiments were carried out, with each one repeated ten times. To assess the models, we provide the average AUROC (Area Under the Receiver Operating Characteristic Curve) and AUPRC (Area Under the Precision-Recall Curve) scores. Table 2 provides the AUROC results, while Table 3 details the AUPRC outcomes. When $\epsilon = \infty$, privacy constraints are not applied. Based on the information in these tables, our model surpasses DPGAN in a supervised setting. Even though MedGAN displays superior results compared to both DPGAN and RDPVAEGAN in terms of AUROC and AUPRC, it does not ensure the privacy of the generated data.

Table 2. This table compares various models in terms of AUROC within the context of the $(1, 10^{-5}) - DP$ setting. For $\epsilon = \infty$, we implemented our RDPVAEGAN model without the privacy enforcement component. MedGAN does not have any privacy enforcement as well. All models were trained utilizing synthetic data.

Dataset	$\epsilon = \infty$	DPGAN	MedGAN	RDPVAEGAN
Sepsis	0.86	0.69	0.81	0.75
Cardiovascular Disease	0.77	0.62	0.73	0.70

Table 3. This table compares various models in terms of AUPRC within the context of the $(1, 10^{-5}) - DP$ setting. For $\epsilon = \infty$, we implemented our RDPVAEGAN model without the privacy enforcement component. MedGAN does not have any privacy enforcement as well. All models were trained utilizing synthetic data.

Dataset	$\epsilon = \infty$	DPGAN	MedGAN	RDPVAEGAN
Sepsis	0.83	0.67	0.80	0.76
Cardiovascular Disease	0.76	0.60	0.72	0.68

This discussion outlines the performance of different methods within the framework of DP. The base model is anticipated to yield the highest accuracy. Thus, the primary investigation is: To what extent does the accuracy decrease across different models when keeping the privacy budget (ϵ, δ) at the same level? Tables 2 and 3 illustrate the results for this specific condition. In most experiments, the artificial data generated by our system demonstrates superior quality in classification tasks when contrasted with other models, all while operating under the equal privacy budget.

5 Conclusion

In this research, we formulated and implemented a method for generating synthetic data that maintains differential privacy, making use of Rényi Differential Privacy. The objective of our model was to extract temporal data and feature correlations through the use of convolutional neural networks. The empirical evidence showed that utilizing variational autoencoders allows for effective management of variables, whether they are discrete, continuous, or a blend of the two. We found that our model surpasses other models in performance while operating within the same privacy constraints. This superior performance can be partially attributed to the reporting of a tighter bound, the use of convolutional networks, and the variational autoencoder. We demonstrated the performance of several models in both supervised and unsupervised approaches, utilizing various metrics across two distinct datasets.

Acknowledgements

This project has been partially funded by the BMBF and the European Union (NextGenerationEU) under project number 16KISA001K (PrivacyUmbrella).

References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. pp. 308–318 (2016)
2. Acs, G., Melis, L., Castelluccia, C., De Cristofaro, E.: Differentially private mixture of generative neural networks. IEEE Transactions on Knowledge and Data Engineering **31**(6), 1109–1121 (2018)
3. Al-Rubaie, M., Chang, J.M.: Privacy-preserving machine learning: Threats and solutions. IEEE Security & Privacy **17**(2), 49–58 (2019)
4. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International conference on machine learning. pp. 214–223. PMLR (2017)
5. Baowaly, M.K., Lin, C.C., Liu, C.L., Chen, K.T.: Synthesizing electronic health records using improved generative adversarial networks. Journal of the American Medical Informatics Association **26**(3), 228–241 (2019)
6. Breiman, L.: Random forests. Machine learning **45**, 5–32 (2001)
7. Chen, J., Chun, D., Patel, M., Chiang, E., James, J.: The validity of synthetic clinical data: a validation study of a leading synthetic data generator (synthea) using clinical quality measures. BMC medical informatics and decision making **19**(1), 1–9 (2019)
8. Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W.F., Sun, J.: Generating multi-label discrete patient records using generative adversarial networks. In: Machine learning for healthcare conference. pp. 286–305. PMLR (2017)
9. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science **9**(3–4), 211–407 (2014)
10. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)
11. Guan, J., Li, R., Yu, S., Zhang, X.: Generation of synthetic electronic medical record text. In: 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). pp. 374–380. IEEE (2018)

12. Han, H., Li, Y., Zhu, X.: Convolutional neural network learning for generic data classification. Information Sciences **477**, 448–465 (2019)
13. Hayes, J., Melis, L., Danezis, G., De Cristofaro, E.: Logan: Membership inference attacks against generative models. arXiv preprint arXiv:1705.07663 (2017)
14. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015)
15. Hjelm, R.D., Jacob, A.P., Che, T., Trischler, A., Cho, K., Bengio, Y.: Boundary-seeking generative adversarial networks. arXiv preprint arXiv:1702.08431 (2017)
16. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. pmlr (2015)
17. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
18. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
19. Kroes, S., Leeuwen, M.: an, groenwold, rhh, & anssen, mp (2022). Generating s nthetic mixed discrete-continuous health records with mixed sum-product networks.(1) pp. 16–25
20. Mironov, I.: Rényi differential privacy. In: 2017 IEEE 30th computer security foundations symposium (CSF). pp. 263–275. IEEE (2017)
21. Mironov, I., Talwar, K., Zhang, L.: R\'enyi differential privacy of the sampled gaussian mechanism. arXiv preprint arXiv:1908.10530 (2019)
22. Narayanan, A., Shmatikov, V.: Robust de-anonymization of large sparse datasets. In: 2008 IEEE Symposium on Security and Privacy (sp 2008). pp. 111–125. IEEE (2008)
23. Park, N., Mohammadi, M., Gorde, K., Jajodia, S., Park, H., Kim, Y.: Data synthesis based on generative adversarial networks. arXiv preprint arXiv:1806.03384 (2018)
24. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
25. Rényi, A.: On measures of entropy and information. In: Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics. vol. 4, pp. 547–562. University of California Press (1961)
26. Reyna, M.A., Josef, C., Seyedi, S., Jeter, R., Shashikumar, S.P., Westover, M.B., Sharma, A., Nemati, S., Clifford, G.D.: Early prediction of sepsis from clinical data: the physionet/computing in cardiology challenge 2019. In: 2019 Computing in Cardiology (CinC). pp. Page–1. IEEE (2019)
27. Shokri, R., Shmatikov, V.: Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. pp. 1310–1321 (2015)
28. Ulianova, S.: Cardiovascular disease dataset (Jan 2019), https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset
29. Villani, C.: Grundlehren der mathematischen wissenschaften (2008)
30. Walonoski, J., Kramer, M., Nichols, J., Quina, A., Moesel, C., Hall, D., Duffett, C., Dube, K., Gallagher, T., McLachlan, S.: Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. Journal of the American Medical Informatics Association **25**(3), 230–238 (2018)

31. Xie, L., Lin, K., Wang, S., Wang, F., Zhou, J.: Differentially private generative adversarial network. arXiv preprint arXiv:1802.06739 (2018)
32. Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling tabular data using conditional gan. Advances in Neural Information Processing Systems **32** (2019)
33. Xu, Q., Huang, G., Yuan, Y., Guo, C., Sun, Y., Wu, F., Weinberger, K.: An empirical study on evaluation metrics of generative adversarial networks. arXiv preprint arXiv:1806.07755 (2018)