Information Sciences 586 (2022) 485–500

Contents lists available at ScienceDirect

Information Sciences

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

Differentially private synthetic medical data generation using convolutional GANs

Amirsina Torfi [a,*], Edward A. Fox [b], Chandan K. Reddy [b]

[a] Instill AI, Fairfax, VA 22031, United States
[b] Department of Computer Science, Virginia Tech, Blacksburg, VA 24061, United States

ARTICLE INFO

Article history:
Received 21 December 2020
Received in revised form 4 December 2021
Accepted 6 December 2021
Available online 11 December 2021

Keywords:
Deep learning
differential privacy
synthetic data generation
generative adversarial networks

ABSTRACT

Deep learning models have demonstrated superior performance in several real-world application problems such as image classification and speech processing. However, creating these models in sensitive domains like healthcare typically requires addressing certain *privacy challenges* that bring unique concerns. One effective way to handle such private data concerns is to *generate realistic synthetic data* that can provide practically acceptable data quality as well as be used to improve model performance. To tackle this challenge, we develop a differentially private framework for synthetic data generation using Rényi differential privacy. Our approach builds on convolutional autoencoders and convolutional generative adversarial networks to preserve critical characteristics of the generated synthetic data. In addition, our model can capture the temporal information and feature correlations present in the original data. We demonstrate that our model outperforms existing state-of-the-art models under the same privacy budget using several publicly available benchmark medical datasets in both supervised and unsupervised settings. The source code of this work is available at https://github.com/astorfi/differentially-private-cgan.

1. Introduction

Deep learning has been successful in a wide range of application domains such as computer vision, information retrieval, and natural language processing due to its superior performance and promising capabilities. However, its success is heavily dependent on the availability of a massive amount of training data. Hence, the progress in deploying such deep learning models can be crippled in certain privacy critical domains such as healthcare where maintaining data privacy is a stringent concern.

Hence, to effectively utilize specific promising data-hungry methods, there is a need to tackle the privacy issues involved in the medical domains. To handle the privacy concerns of dealing with sensitive information, a common method that is often used in practice is the anonymization of personally identifiable information. But, such approaches are susceptible to de-anonymization attacks [1]. That made researchers explore alternative methods. To make the system immune to such attacks, privacy-preserving machine learning approaches have been developed [2]. This particular privacy challenge is usually compounded by some auxiliary ones due to the presence of complex and often noisy data that, in practice, might typically consist of a combination of multiple types of data such as discrete, continuous, and categorical.

* Corresponding author.
E-mail address: amirsina.torfi@gmail.com (A. Torfi).
URL: https://www.sinatorfi.com (A. Torfi).

https://doi.org/10.1016/j.ins.2021.12.018
0020-0255/

1.1. Synthetic Data Generation

One of the most promising privacy-preserving approaches is *Synthetic Data Generation (SDG)*. Synthetically generated data can be shared publicly without privacy concerns and provides many collaborative research opportunities, including for tasks such as building prediction models and finding patterns. As SDG inherently involves a generative process, Generative Adversarial Networks (GANs) [3] attracted much attention in this research area, due to their recent success in other domains.

GANs are not reversible, i.e., one may not use a deterministic function to go from the generated samples to the real samples, as the generated samples are created using an implicit distribution of the real data. However, a naive usage of GANs for SDG does not guarantee the system being privacy-preserving by just relying on the fact that GANs are not reversible, as GANs are already proven to be vulnerable [4].

This problem becomes much more severe when such privacy violations can have serious consequences, such as when dealing with patient sensitive data in the medical domain. Two important questions arise: (1) How private is a system? (2) How much information is leaked during the training process? Thus, *there is a need to measure the privacy of a system* – to be able to judge whether a system is privacy-preserving.

1.2. Differential Privacy

Differential Privacy (DP) [5] provides a mechanism to ensure and quantify the privacy of a system using a solid mathematical formulation. Differential privacy recently became the de facto standard for statistical exploration of databases that contain sensitive private data. The power of differential privacy is in its accurate mathematical representation, that ensures privacy, without restricting the model for accurate statistical reasoning. Furthermore, utilizing differential privacy, one can measure the privacy level of a system.

Differential privacy, as a strong notion of privacy, becomes crucial in machine learning and deep learning as the system by itself often employs sensitive information to augment the model performance for prediction. Although many different attack types can jeopardize a machine learning model's privacy, it is safest to assume the presence of a powerful adversary with complete knowledge of the pipeline, training, and model [6]. Hence, protecting the system against such an adversary, or at the very least, measuring the upper bound of the privacy leakage in this scenario, can give assurance regarding the system's privacy. A fully differentially private system guarantees that the algorithm's training is not dependent on an individual's sensitive data.

One of the most adopted approaches to ensure differential privacy, under a modest *privacy budget*, and model accuracy, is *Differentially Private Stochastic Gradient Descent (DP-SGD)*, proposed in [7], which is the basis for many different research studies [8,9]. At its core, DP-SGD (1) bounds the sensitivity of the algorithm on individuals by clipping the gradients, (2) adds Gaussian noise, and (3) performs the gradient descent optimization step. As of now, SGD under a DP regime is one of the most promising privacy-preserving approaches.

Due to the rapid rise in utilization of differential privacy, many discussions have addressed practical issue of tracking the privacy budget. Despite its great advantages, the proposed notion of differential privacy has some disadvantages due to what is called the *composition theorem* [5,10]. According to the theorem, for some composed mechanisms, the privacy cost simply adds up, leading to discussion of privacy budget restriction. This is problematic, especially in deep learning, as the model training is an iterative process, and each iteration adds to the privacy loss. To tackle the shortcomings of the differential privacy definition, *Rényi Differential Privacy (RDP)* has been proposed as a natural relaxation of differential privacy [11]. As opposed to differential privacy, RDP is a more robust notion of privacy that may lead to a more accurate and numerically stable computation of privacy loss.

1.3. Challenges with Medical Data

There are *four primary challenges regarding synthetic data generation research in the medical domain*: (1) **Preserving the privacy.** The majority of the existing works do not train the model in a privacy-preserving manner. At best, they only try to address privacy with some statistical or machine learning-based measurements. (2) **Handling discrete data**. When discrete data is involved, many of the methods using GANs face some difficulty since GAN models are designed for generating continuous values, i.e., they fail when there is a mixture of continuous and discrete data. (3) **Evaluation of synthetic data quality**. One of the challenging issues regarding the evaluation of GANs and the quality of synthetic data in a real-world domain is: How can we measure the quality of synthetically generated data? However, this becomes particularly critical in the medical domain since the use of low-quality synthetic data for building prediction models can have dire consequences and may even jeopardize human lives. (4) **Temporal information and correlated features**. The temporal and local correlation between the features is often ignored. Incorporating such information is important in the medical domain since disease incidents, and individual patient records, often exhibit meaningful temporal/correlation patterns. The quality of the generated data can be significantly improved by considering such dependencies.

1.4. Our Contributions

To address the problems described before, we propose a privacy-preserving framework that employs **Ré**nyi **D**ifferential **P**rivacy and **C**onvolutional **G**enerative **A**dversarial **N**etworks (RDP-CGAN). Our work makes the following contributions:

- We track and compute privacy loss with stable numerical computations and have *tighter bounds on the privacy loss function, which leads to better performance under the same privacy budget*. We achieve this by employing Rényi differential privacy in our model.
- Our model can effectively *handle discrete data* and capture the information from a *mixture of discrete–continuous data* by creating a compact, unified representation through *Convolutional Autoencoders (CEs)* for unsupervised feature learning.
- In addition to measuring the quality of synthetic data with statistical measurements, we also employ a *labeled synthetic data generation mechanism* for assessing the quality of the synthetic data in a supervised manner.
- To *incorporate temporal and correlation dependencies in the data*, we utilize one-dimensional convolutional neural networks.

We demonstrate that our model generates better quality synthetic data (under the same privacy budget) compared to existing state-of-the-art approaches (see Table 1, which is explained below). Likewise, it can provide a higher level of privacy under the same level of synthetic data quality. We also empirically show that, regardless of the privacy considerations, our proposed architecture generates higher quality synthetic data, since it captures temporal and correlated information.

The rest of this work is organized as follows: Section 2 discusses some prior research efforts related to synthetic medical data generation, distinguishing that from our work. Section 3 describes preliminary concepts. Section 4 provides structural and implementation details of the proposed system. Section 5 compares our results regarding privacy and synthetic data quality, with state-of-the-art alternatives. Finally, Section 6 concludes this paper.

2. Related Works

There are a wide variety of works utilizing Differential Privacy for generating synthetic data. A majority of them use the mechanism proposed in [7] to train a neural network with differential privacy based upon gradient clipping for bounding the gradient norms and adding noise. That follows the general mechanism proposed in [5]. One of the most critical contributions in [7] is introducing the privacy accountant which tracks privacy loss. Motivated by the success of this mechanism, in our approach, we extend our privacy-preserving framework using Rényi Differential Privacy (RDP) [11] as a new notion of DP to calculate the privacy loss.

Much of the recent research has addressed challenges regarding synthetic healthcare data generation [16–18]. One of the early systems generating synthetic medical data is MedGAN [12], in which there is no privacy-preserving mechanism being enforced. It only uses GAN models to create synthetic data, though it is known that GANs are easy to attack. Hence, despite its good performance in generating data, it does not provide any kind of privacy guarantee. Nevertheless, *MedGAN* contributes by generating discrete data through denoising autoencoders [12]. On the other hand, utilization of synthetic patient population simulators such as Synthea [19] is minimal as they only rely on standard guidelines and do not model factors that may contribute to predictive analysis [20]. Another work is *CorGAN* [14] that tries to generate longitudinal event sequences by capturing feature correlations and temporal information. The *TableGAN* approach [13] also uses convolutional GANs to synthesize data by leveraging auxiliary classifier GANs. Another work in this area is the *CTGAN* model presented in [21] that addresses tabular data which has a mixture of discrete and continuous features. However, none of these approaches guarantees any sort of privacy during the data generation. Hence, there is a good chance that many of these models can compromise the privacy of the original medical data, making these models vulnerable in practice. Accordingly, we will now discuss various privacy-preserving strategies.

One of the earliest works that addressed the privacy-preserving deep learning in the healthcare domain is [22], that uses *Auxiliary Classifier GANs* [23] to generate synthetic data. It assumes having access to labeled data. However, we aim to develop a model able to create synthetic data from both labeled and unlabeled real data. *PATE-GAN* [15] is one of the successful SDG approaches that use the *Private Aggregation of Teacher Ensembles (PATE)* [24,25] mechanism to ensure Differential Privacy. Another framework related to this topic is *DPGAN* [8], which implements a mechanism similar to the one developed in [7], with the main difference of clipping weights instead of gradients. We compare our work with PATE-GAN and DPGAN due to their success and relevance to the medical domain. One of the research efforts associated with utilizing RDP is [26] in which the authors training a privacy-preserving model on the MNIST dataset build on conditional GANs. The work proposed in [27] uses RDP to enforce differential privacy. However, they do not consider temporal information. Further, we claim that their system is not fully-differentially private as they only partially apply differential privacy. A general comparison for related methods is depicted in Table 1. The rows in Table 1 describe important characteristics. First, we ask if a model *preserves the privacy or not*. Second, it is crucial to assess the model's utility by checking if a model *requires labels*. If a model does not need labels, it can be used for unsupervised synthetic data generation. Third, we identify approaches that can *handle a mixture of data types*. Finally, we consider models that can *capture correlated and temporal information*.

Table 1
Comparison of various methods based on different criteria.

Method	MedGAN [12]	TableGAN [13]	CorGAN [14]	DPGAN [8]	PATE-GAN [15]	RDP-CGAN (Proposed)
Privacy-Preserving	×	×	×	✓	✓	✓
Does Not Require Labels (Annotations)	✓	×	✓	✓	×	✓
Can Handle Mixed Data Types	✓	✓	✓	×	×	✓
Capture Correlated and Temporal Information	×	✓	✓	×	×	✓

To assess the privacy-preserving characteristics of our model, we compare our method to two approaches that guarantee differential privacy in the context of synthetic data generation, namely, DPGAN and PATE-GAN. PATE-GAN uses a modified version of a PATE mechanism. One issue with the original PATE is that its privacy cost depends on the amount of data it requires for a labeling process incorporated in the mechanism. That PATE mechanism uses some public dataset (similar to real private data) for its labeling mechanism. Not only can this increase the privacy loss significantly, but the availability of such a labeled public dataset is a limiting assumption in a practical, real-world scenario. Although PATE-GAN changed the PATE training paradigm so that it does not require public data, its basis is to train a student model by employing generator output that may aggregate the error, and it still needs labeled data. However, the advantage of PATE is that it provides a tighter privacy upper bound as opposed to the mechanism used in DPGAN. Regarding the privacy, the advantage of our method compared to both these methods is that *we use the RDP mechanism which provides a tighter privacy upper bound*. Furthermore, our approach uses *convolutional architectures which typically perform better than standard multilayer perceptron models* by capturing the correlated features and temporal information. Furthermore, we use *convolutional autoencoders that, in an unsupervised manner*, create a feature space that can effectively incorporate both discrete and continuous values.

3. Preliminaries

3.1. Autoencoders

Autoencoders are a kind of neural network architecture that consists of two encode and decode functions. $\mathbf{Enc}(.) : \mathbb{R}^n \rightarrow \mathbb{R}^d$ and $\mathbf{Dec}(.) : \mathbb{R}^d \rightarrow \mathbb{R}^n$ aim to transpose the input $\mathbf{x} \in \mathbb{R}^n$ to the latent space $\mathcal{I} \in \mathbb{R}^d$ and then reconstruct $\hat{\mathbf{x}} \in \mathbb{R}^n$. In an ideal scenario, a perfect reconstruction of the original input data can be achieved, i.e., $x = \hat{x}$. The autoencoders usually employ *Binary Cross Entropy (BCE)* and *Mean Square Error (MSE)* losses for binary and continuous inputs, respectively.

$$BCE(\mathbf{x}, \hat{\mathbf{x}}) = -\frac{1}{m}\sum_{i}^{m}(x_i log(\hat{x}_i) + (1 - x_i) log(1 - \hat{x}_i)) \tag{1}$$

$$MSE(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{m}\sum_{i=1}^{m}|x_i - \hat{x}_i|^2 \tag{2}$$

We utilize autoencoders in our model for capturing low-dimensional representations for both discrete and continuous variables.

3.2. Differential Privacy

Differential Privacy establishes a guarantee of individuals' privacy via measuring the privacy loss (associated with any information release extracted from a database) by a mathematical definition [10,5]. The most widely adopted definition is (ϵ, δ)-differential privacy.

Definition 1 ((ϵ, δ)-*DP*). The randomized algorithm $\mathcal{A} : \mathcal{X} \rightarrow \mathcal{Q}$, is (ϵ, δ)-differentially private for all query outcomes $\mathcal{Q}$ and all neighbor datesets D and $D\prime$ if:

$$Pr[\mathcal{A}(D) \in \mathcal{Q}] \leqslant e^{\epsilon} Pr[\mathcal{A}(D\prime) \in \mathcal{Q}] + \delta \tag{3}$$

Two datasets $D, D\prime$ that only differ by one record (e.g., a patient's record) are called *neighbor datasets*. The notion of neighboring datasets emphasizes the sensitivity of any individual private data. The parameters (ϵ, δ) denote the privacy budget, i.e., being differentially private does not mean we have absolute privacy. It only indicates our confidence level of privacy, given the (ϵ, δ) parameters. The smaller the (ϵ, δ) parameters are, the more confident we become about our algorithm's privacy, as (ϵ, δ) indicates the privacy loss by definition.

The (ϵ, δ)-DP with $\delta = 0$ is called ϵ-DP which is the initially proposed definition [10]. It provides a more substantial promise of privacy as even a tiny amount of δ might be a dangerous privacy violation due to the distribution shift [10]. One of the most important reasons in the usage of (ϵ, δ)-DP is the applicability of advanced composition theorems [5,28].

Theorem 1 (*Strong Composition [5]*). *Let us assume that we execute a k-fold adaptive composition of a mechanism with* (ϵ, δ)*-DP. Then, the composite mechanism is* $(\epsilon\prime, k\delta\prime + \delta)$*-DP for* $\delta\prime$*, in which* $\epsilon\prime = \sqrt{2kln(1/\delta\prime)}\epsilon + k\epsilon(e^{\epsilon} - 1)$ *[5].* ▲

3.3. Rényi Differential Privacy

Strong composition (see Definition 1) enables a tighter upper bound for compositions of (ϵ, δ)-DP steps compared to the basic composition theorem [5]. One particular problem with the advanced composition theorem is that iterating this process leads to the rapid growth of parameters as each application of the theorem (each step of the iterative process) leads to a selection of possible $(\epsilon(\delta), \delta)$ values. To address some of the shortcomings of the (ϵ, δ)-DP definition, the *Rényi Differential Privacy (RDP)* has been proposed in [11] based on Rényi divergence (Eq. 2).

Definition 2 (*Rényi divergence of order* α *[29]*). The Rényi divergence of order α of a distribution P from the distribution $P\prime$ is defined as:

$$D_{\alpha}(P||P\prime) = \frac{1}{\alpha - 1} log\left(E_{P\prime(x)}\left(\frac{P(x)}{P\prime(x)}\right)^{\alpha}\right) \tag{4}$$

The Rényi divergence, as a generalization of the Kullback–Leibler divergence, is precisely equal to the Kullback–Leibler divergence for the order $\alpha = 1$. The special case of $\alpha = \infty$ is:

$$D_{\infty}(P||P\prime) = log\left(sup_{P\prime(x)}\frac{P(x)}{P\prime(x)}\right) \tag{5}$$

which is the log of the maximum ratio of the probabilities over $P\prime(x)$. The order of $\alpha = \infty$ establishes the connection between the Rényi divergence and ϵ-DP. A randomized mechanism $\mathcal{A}$ is ϵ-differentially private if for two neighbor datasets D and $D\prime$, we have:

$$D_{\infty}(\mathcal{A}(D)||\mathcal{A}(D\prime)) \leqslant \epsilon \tag{6}$$

Now, based on these definitions, let us define Rényi differential privacy (RDP) [11] as a new notion of differential privacy.

Definition 3 (*Rényi Differential Privacy (RDP) [11]*). The randomized algorithm $\mathcal{A} : \mathcal{D} \rightarrow \mathcal{U}$ is (α, ϵ)-RDP if, for all neighbor datasets D and $D\prime$, we have:

$$D_{\alpha}(\mathcal{A}(D)||\mathcal{A}(D\prime)) \leqslant \epsilon \tag{7}$$

There are two essential properties of the RDP definition (Definition 3) that are required here.

Proposition 1 (*Composition of RDP [11]*). If $\mathcal{A} : \mathcal{X} \rightarrow \mathcal{U}_1$ is (α, ϵ_1)-RDP and $\mathcal{B} : \mathcal{U}_1 \times \mathcal{X} \rightarrow \mathcal{U}_2$ is (α, ϵ_2)-RDP, then the mechanism $(\mathcal{M}_1, \mathcal{M}_2)$, where $\mathcal{M}_1 \sim \mathcal{A}(\mathcal{X})$ and $\mathcal{M}_2 \sim \mathcal{B}(\mathcal{M}_1, \mathcal{X})$, satisfies the $(\alpha, \epsilon_1 + \epsilon_2)$-RDP conditions.▲

Proposition 2. If the mechanism $\mathcal{A} : \mathcal{X} \rightarrow \mathcal{U}$ is (α, ϵ)-RDP then it also obeys $(\epsilon + \frac{log(1/\delta)}{(\alpha - 1)}, \delta)$-DP for any $0 < \delta < 1$ [11]. ▲

The above two propositions form the basis for preserving privacy in our approach. Proposition 1 determines the privacy cost as a composition of two structures (autoencoder and convolutional GAN). Proposition 2 is applicable when one wants to calculate the extent to which our system is differentially private based on the traditional (ϵ, δ) notion (see Definition 1).

4. Privacy-Preserving Framework

We build a privacy-preserving GAN model using Rényi differential privacy (see Fig. 1). Since it is well known that GAN models typically have poor performance in generating non-continuous data [30], we utilize autoencoders [31] to aid the GAN model by creating a continuous feature space to represent the input. At the same time, the GAN aims to generate high-fidelity synthetic data in a privacy-preserving manner. The autoencoder will transform the input space into a continuous space, whether we have a discrete, continuous, or a mixture of both types as the input. Thus, the autoencoder acts as a bridge between the GAN model and non-continuous data present in the medical domain.

Furthermore, the majority of existing research efforts in generating synthetic data [12] ignore the features' local correlations or temporal information. They use multilayer perceptrons, which are inconsistent with real-world scenarios such as disease progression. To remedy this drawback, for both autoencoder and GAN architecture (generator & discriminator), we use convolutional neural networks. Specifically, we employ *One Dimensional Convolutional Neural Networks (1D-CNNs)* to capture correlated input features' patterns as well as temporal information and incorporate them within an autoencoder based framework.

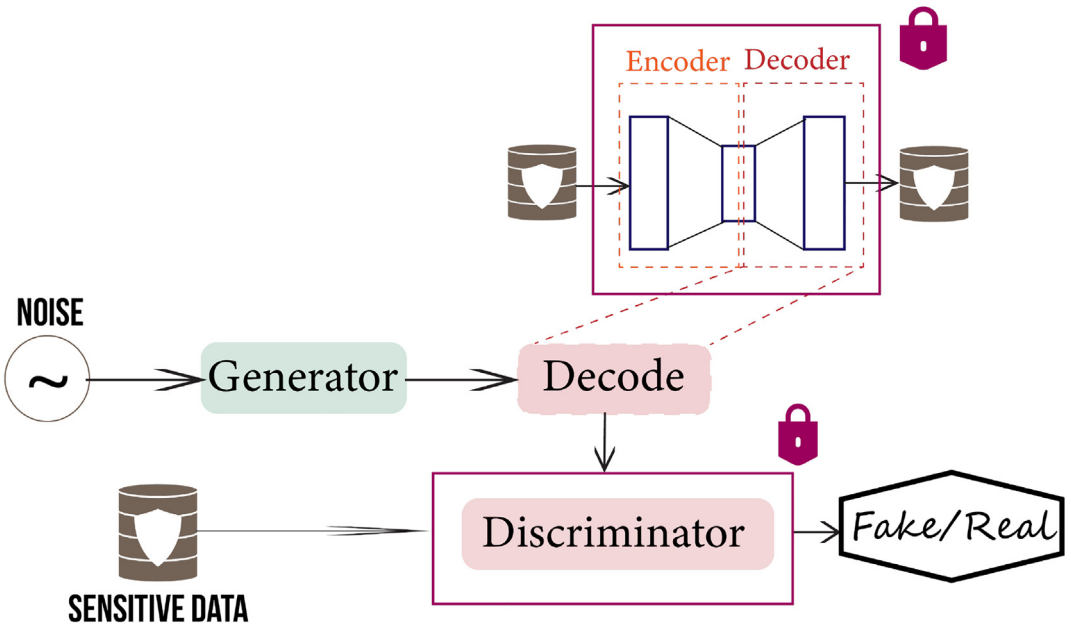


Fig. 1. The proposed RDP-CGAN framework. In the Autoencoder, the Encoder and Decoder are formed with convolutional layers. The Discriminator and Generator have architectures similar to that of the Encoder and Decoder, respectively.

The proposed framework is depicted in Fig. 1. The inputs to the *generator* $\mathbf{G} : \mathbb{R}^r \rightarrow \mathcal{D}_g^d$ and the *discriminator* $\mathbf{D} : \mathbb{R}^n \rightarrow \mathcal{D}_d$ are the random noise $\mathbf{z} \in \mathbb{R}^r$ sampled from $\mathcal{N}(0, 1)$ and the real data $\mathbf{x} \in \mathbb{R}^n$. $\mathcal{D}_g^d$ and $\mathcal{D}_d$ are the generator and discriminator domains which are usually $\{0, 1\}$ and $[-1, 1]^d$, respectively. In RDP-CGAN, as opposed to the regular GAN's training procedure, the generated fake data is *decoded* before being fed to the discriminator. The decoding is done by feeding the fake data to a pre-trained autoencoder.

Algorithm 1: Private Convolutional Autoencoder Pre-Training

Input: Real data $X = \{x_i\}_{i=1}^N$, learning rate η, network weights θ, number of epochs for autoencoder training n_{ae}, norm bound C, and additive noise standard deviation σ_{ae}.
for $i = 1 \ldots n_{ae}$ **do**
  Sample a mini-batch of n examples. $\mathcal{X} = \{x_i\}_{i=1}^n$.
  Partition $\mathcal{X}$ into $\mathcal{X}_1, \ldots, \mathcal{X}_r$ where $r = \lfloor \frac{n}{K} \rfloor$.
  for $l = 1 \ldots r$ **do**
    $\mathcal{L} = BCE(\mathcal{X}_l, \hat{\mathcal{X}}_l)$, $\hat{\mathcal{X}}_l = Dec(Enc(\mathcal{X}_l))$
    $\mathbf{g}_{\theta,\mathbf{l}} \leftarrow \nabla_\theta \mathcal{L}(\theta, \mathcal{X}_l)$
    $\hat{\mathbf{g}_{\theta,\mathbf{l}}} \leftarrow \mathbf{g}_{\theta,\mathbf{l}} / max\left(1, \frac{\|\mathbf{g}_{\theta,\mathbf{l}}()\|_2}{C}\right)$
  end for
  $\hat{\mathbf{g}_\theta} \leftarrow \frac{1}{r} \sum_{l=1}^r \left(\hat{\mathbf{g}_{\theta,\mathbf{l}}} + \mathcal{N}(0, \sigma_{ae}^2 C^2 \mathbb{I})\right)$
  Update: $\hat{\theta} \leftarrow \theta - \eta \hat{\mathbf{g}_\theta}$
end for

Algorithm 2: Private 1-D Convolutional GAN Training

Input: Real data $X = \{x_i\}_{i=1}^{N}$, random noise $\mathbf{z}$ where $z_i \sim \mathcal{N}(0,1)$, learning rate η, discriminator weights ω, generator weights ψ, number of epochs for GAN training n_{gan}, norm bound C, additive noise standard deviation σ_{gan}, and number of discriminator training steps per one step of generator training n_d.
for $j = 1 \ldots n_{gan}$ **do**
  for $k = 1 \ldots n_d$ **do**
    Take a mini-batch from private data $\mathcal{X} = \{x_i\}_{i=1}^{n}$
    Sample a mini-batch $\mathcal{Z} = \{z_i\}_{i=1}^{n}$
    Partition real data mini-batches into $\mathcal{X}_1, \ldots, \mathcal{X}_r$
    Partition noise data mini-batches into $\mathcal{Z}_1, \ldots, \mathcal{Z}_r$
    for $l = 1 \ldots r$ **do**
      $x_i \in \mathcal{X}_l$ and $z_i \in \mathcal{Z}_l$
      $\mathcal{L} \leftarrow \frac{1}{k}\sum_{i=1}^{k}(D(x_i) - D(\mathbf{Dec}(G(z_i))))$
      $\mathbf{g}_{\omega,\mathbf{l}} \leftarrow \nabla_{\omega}\mathcal{L}(\omega, \mathcal{X}_l)$
      $\hat{\mathbf{g}}_{\omega,\mathbf{l}} \leftarrow \mathbf{g}_{\omega,\mathbf{l}} / max\left(1, \frac{\|\mathbf{g}_{\omega,\mathbf{l}}()\|_2}{C}\right)$
    end for
  $\hat{\mathbf{g}}_{\omega} \leftarrow \frac{1}{r}\sum_{l=1}^{r}\left(\hat{\mathbf{g}}_{\omega,\mathbf{l}} + \mathcal{N}(0, \sigma_{gan}^2 C^2 \mathbb{I})\right)$
  Update: $\hat{\omega} \leftarrow \omega - \eta\hat{\mathbf{g}}_{\omega}$
  end for
  Sample $\{z_i\}_{i=1}^{n}$ from noise prior
  $\mathcal{L} \leftarrow -\frac{1}{n}\sum_{i}^{n}(D(Dec(G(z_i))))$
  $\mathbf{g}_{\psi} \leftarrow \nabla_{\psi}\mathcal{L}(\psi, \mathcal{Z})$
  Update: $\hat{\psi} \leftarrow \psi - \eta\mathbf{g}_{\psi}$
end for

4.1. Convolutional Autoencoder

We build a *one-dimensional convolutional autoencoder (1D-CAE)* based architecture. The main goal of such a 1D-CAE is to: **(1)** capture the correlation between neighboring features, **(2)** represent a new compact feature space, **(3)** transform the possible discrete records to a new continuous space, and **(4)** simultaneously model discrete and continuous phenomena (which is a challenging problem [21]).

We enforce privacy by adding noise, and by clipping gradients of both the encoder and decoder (see Algorithm 1, *Autoencoder Pretraining Step*), as in some previous studies [9,32]. One might claim adding noise to the decoder part should suffice since the discriminator only has access to the decoder part of the CAE. We assert that such an action might jeopardize the privacy of the model since we cannot rely on the autoencoder training to be trusted and secure as it has access to the real data. The details of autoencoder pretraining, demonstrated in Algorithm 1, are as follows:

- We pre-train the autoencoder for n_{ae} steps (which will be determined based on the privacy budget ϵ), and hence it varies by changing the desired level of ϵ.
- We divide a mini-batch into several micro-batches.
- For each micro-batch of size 1, we calculate the loss (*line 7*), calculate the gradients (*line 8*), and clip the gradients to bind the model's sensitivity to the individuals (*line 9*). The operation $\|\mathbf{g_i}()\|_2$ indicates the ℓ_2 norm over the micro-batch gradients, and C is an arbitrary upper bound for the gradients' norm.
- The Gaussian noise ($\mathcal{N}(0, \sigma_{ae}^2 C^2 \mathbb{I})$) will then be independently added to the gradients of each micro-batch and will be aggregated (*line 10*).
- The last step is where the optimizer performs the parameter update (*line 11*).

4.2. Convolutional GAN

The pseudocode for the Convolutional GAN is given in Algorithm 2, under the procedure *GAN Training Step*. As can be observed in Algorithm 2, we only enforce differential privacy on the discriminator since it is the only component that has access to the real data. To avoid the issue of mode collapse [33], we train the CGAN model, using Wasserstein GAN [34], since it is an efficient approximation of *Earth Mover's (EM)* distance and is shown to be robust to the mode collapse problem during model training [34].

The *Earth-Mover's (EM)* distance or Wasserstein-1 distance represents the minimum price in transforming the generated data distribution $\mathbf{P_g}$ to the real data distribution $\mathbf{P_r}$:

$$W(\mathbb{P}_x, \mathbb{P}_g) = \inf_{\vartheta \in \Pi(\mathbb{P}_x, \mathbb{P}_g)} E_{(x,y)\sim\vartheta}[\,\|x-y\|\,], \tag{8}$$

$\vartheta(x, y)$ refers to how much "mass" should be moved from x to y to transform $\mathbb{P}_x$ to $\mathbb{P}_g$.

However, as the infimum in (8) is intractable, based on the Kantorovich-Rubinstein duality [35], WGAN employs the Eq. (9) optimization:

$$W(\mathbb{P}_x, \mathbb{P}_g) = \sup_{\|f\|_L \leqslant 1} E_{x\sim\mathbb{P}_x}[f(x)] - E_{x\sim\mathbb{P}_g}[f(x)] \tag{9}$$

For simplicity of the definition, infimum and supremum indicate the greatest lower bound and the least upper bound, respectively.

Definition 4 (*1-Lipschitz functions*). Given two metric spaces (X, d_X) and (Y, d_Y), where d denotes the metric (e.g., distance metric), the function $f : \mathbb{X} \to \mathbb{Y}$ is called K-Lipschitz if:

$$\forall(x, x\prime) \in \mathbb{X}, \exists K \in \mathbb{R} : d_Y(f(x), f(x\prime)) \leqslant K d_X(x, x\prime) \tag{10}$$

Using the distance metric and $K = 1$, Eq. (10) is equivalent to:

$$\forall(x, x\prime) \in \mathbb{X} : |f(x) - f(x\prime)| \leqslant |x - x\prime| \tag{11}$$

It is clear that for *computing the Wasserstein distance, we should find a 1-Lipschitz function (see Definition 4 and Eq. 11)*. The approach is to build a neural model to learn it. The procedure is to construct a discriminator D without the Sigmoid function, and output a scalar instead of the probability of confidence.

Regarding the privacy considerations, the generator has no access to real data directly (although it has access to the gradients from the discriminator); only the discriminator will have access to it. We propose to only train the discriminator under differential privacy for this aim. More intuitively, this approach considers the post-processing theorem in differential privacy as follows.

Theorem 2 (*Post-processing [5]*). *Assume* $\mathbf{D} : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{D}$ *is an algorithm which is* (ϵ, δ)*-differentially private and* $\mathbf{G} : \mathbb{D} \to \mathbb{O}$ *is some arbitrary function operates on. Then* $\mathbf{G} \circ \mathbf{D} : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{O}$ *is* (ϵ, δ)*-differentially private as well [5].*▲

Following the argument above, we consider **D** and **G** as discriminator and generator functions, respectively. Since the generator is an arbitrarily randomized mapping on top of the discriminator, enforcing the differential privacy on the discriminator suffices to guarantee that the overall system ensures differential privacy, and we do not need to train a private generator as well. The generator training procedure is given in Algorithm 2, *lines 17–20*. As can be observed, we do not have a private generator and the loss function is the regular generator loss function of the WGAN method [34].

4.3. Architecture Details

For the GAN discriminator, we used five convolutional layers similar to the autoencoder (encoder), except for the last layer. It is another dense layer, of output size 1, for decision making. For all layers, we used PReLU activation [36], except for the last layer that does not use any activation. For the generator, we used *transposed convolutions* (also called *fractionally-strided convolutions*) similar to the ones in [37]. However, we use 1-D transposed convolutions. The GAN generator has an input noise of size *100*; its output size is set to 128.

In the autoencoder and the encoder, we used an architecture similar to the GAN discriminator, except we discard the last layer of the discriminator. For the decoder, as in the generator, we used *transposed convolutional layers* in the reverse order of the ones used for the encoder. For the encoder, we used *PReLU* activation layers except for the last layer of the encoder where *Tanh* was used to match the generator output. For the decoder, we also used *PReLU* activation except for the last layer where a *Sigmoid* activation function was used to bound the range of output data to $[0, 1]$ to ideally reconstruct the discrete range of $\{0, 1\}$ aligned with the input data. The decoder input size (encoder output size) is equal to the GAN's generator output dimension.

Since our model inputs' sizes may change for different datasets, we modify the input pipeline of our architecture by varying the dimensions – of convolution kernels, stride for the GAN discriminator, and the autoencoder – to match the new dimensionality. The GAN generator *does not require any change* since, for all experiments, we used the same noise dimension as mentioned above.

4.4. Privacy Loss

As tracking the RDP privacy accountant [7] is computationally more precise than the regular DP, *we based our privacy loss calculation on RDP*. Then, the RDP computations can be transformed to DP according to Proposition 2. The Gaussian noise

additive procedure is also called *Sampled Gaussian Mechanism (SGM)* [38]. For tracking privacy loss, we use the following theorem.

Theorem 3 (*SGM privacy loss [38]*). *Let us suppose D and D′ are two neighbor datasets and $\mathcal{G}$ is a SGM applied to a function f with ℓ_2-sensitivity of one. If we set the following:*

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

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

where φ_1 and φ_2 are PDFs, then, the mechanism $\mathcal{G}$ satisfies (α, ϵ)-RDP for:

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

where $A_\alpha \triangleq \mathbb{E}_{x\sim\varphi_1}\left[(\varphi_2/\varphi_1)^\alpha\right]$ and $B_\alpha \triangleq \mathbb{E}_{x\sim\varphi_2}\left[(\varphi_1/\varphi_2)^\alpha\right]$. ▲

It can be proven that $A_\alpha \leqslant B_\alpha$ and the RDP privacy computation can solely be focused on upper bounding A_α which can be calculated with a closed-form bound and numerical calculations [38]. We use the same numerical calculations here. However, that bounds ϵ for each step. The overall bound of ϵ for the entire training process can be calculated by $\epsilon_{total} = num\_steps \times \epsilon$ for any given α (see Proposition 1). To determine a tighter upper bound, we try multiple α values and use the minimum (ϵ) and its associated α (obtained by the RDP privacy accountant) to compute the (ϵ, δ)-DP employing Proposition 2.

System privacy budget: One important question is: How do we calculate the (α, ϵ)-RDP for the *combination of an autoencoder and GAN training?* Consider Proposition 1, in which $\mathcal{X}, \mathcal{U}_1$, and $\mathcal{U}_2$ are the input and output spaces of the autoencoder and the output space of the discriminator. The mechanisms $\mathcal{A}$ and $\mathcal{B}$ are the autoencoder and discriminator, respectively. Consider $\mathcal{U}_1$ as what the discriminator observes after decoding of the fake samples, and $\mathcal{X}$ as the space of real inputs. Then, Proposition 1 directly results in having the whole system with $(\alpha, \epsilon_{ae} + \epsilon_{gan})$-RDP by fixing the α. But we cannot guarantee that we can have a fixed α, and the RDP has a budget curve parameterized by it. The procedure for fixing α is as follows.

- Let us assume we have two systems S_1 and S_2 such that one is the autoencoder, and the other is the GAN. Hence we have two systems which (α_1, ϵ_1)-RDP and (α_2, ϵ_2)-RDP.
- Without loss of generality, we assume $\epsilon_1 \leqslant \epsilon_2$. We pick $\alpha_{total} = \alpha_2$. Now, we have system S_2 which $(\alpha_{total}, \epsilon_2)$-RDP.
- For S_1 we pick $\alpha_{total} = \alpha_1$ and calculate $\epsilon\prime$ such that $\epsilon \leqslant \epsilon\prime$.
- Now, the total system (S_1, S_2) satisfies $(\alpha_{total}, \epsilon_2 + \epsilon\prime)$-RDP.

5. Experimental Results

In this section, we present the details of the experimental setup, report results obtained in various experiments, and compare with different methods available in the literature.

5.1. Experimental Setup

We split the dataset to train with $\mathcal{D}_{tr}$ and test with sets $\mathcal{D}_{te}$. We utilize $\mathcal{D}_{tr}$ to train the models, and then generate synthesized samples $\mathcal{D}_{syn}$ using the trained model. We set $|\mathcal{D}_{syn}| = |\mathcal{D}_{tr}|$. Recall from Section 4 that although for different datasets we alter the architectures associated with the input size, the encoder output space (decoder input space) and the generator output space will always have the same dimensionality. Furthermore, the encoder input space, the decoder output space, and the discriminator input space will also have the same dimensionality. It is worth noting that all of the reported ϵ values are associated with the (ϵ, δ)-DP definition with $\delta = 10^{-5}$, unless otherwise stated.

Both the convolutional autoencoder and convolutional GAN were trained with the *Adam optimizer* [39] with learning rate 0.005, with a mini-batch size of **64**. For both generator and discriminator we used Batch Normalization (BN) [40] to improve the training. We used one GeForce RTX 2080 NVIDIA GPU for our experiments.

We compare our framework with various different methods. Depending on the experiment, some of those methods may or may not be used for comparison, according to their characteristics. For further details, see Table 1.

5.2. Datasets

To evaluate performance, we used the following datasets:

1. **MIMIC-III [41]:** This dataset consists of the medical records from around 46 K patients.

2. **Kaggle Cervical Cancer [42]:** This dataset covers patient records and is used to classify cervical cancer. There are multiple attributes in the dataset of discrete and continuous types, and one attribute associated with the class label (e.g., Biopsy as the target for classification).
3. **UCI Epileptic Seizure Recognition [43]:** Here the task is to classify seizure activity. The features are the values of Electroencephalogram (EEG) records at various time points.
4. **PTB Diagnostic ECG [44]:** The electrocardiograms (ECGs) are used to classify the heart activity as normal or abnormal. This dataset contains 14552 samples, out of which 4046 are classified as normal, while 10506 are abnormal activities.
5. **Kaggle Cardiovascular Disease [45]:** This dataset is used to determine if a patient has cardiovascular disease or not, from a variety of features such as age, systolic blood pressure, and diastolic blood pressure – with continuous as well as these discrete features.
6. **MIT-BIH Arrhythmia [46]:** This database includes ambulatory ECG recordings, created from 47 subjects. The records were obtained from a mixed group of inpatients (requiring overnight hospitalization) and outpatients (usually not requiring overnight hospitalization) at Boston's Beth Israel Hospital. Unlike other datasets in this set of experiments, this database is used in a multicategory classification task. The ECG signals correspond to normal (negative) and abnormal (positive) cases. There are a total of five classes. Class zero is associated with normal cases. The rest of the classes are associated with different arrhythmias and myocardial infarction, which we call abnormal.

5.3. Baseline Comparison Methods

We compare our model with different benchmark methods (see Table 1). Depending on the nature of the experiments, various models are used as benchmarks. Thus, if the experiments are associated with different privacy settings, models that do not preserve privacy will not be used for comparison.

1. **MedGAN [12]:** MedGAN's architecture is designed to generate discrete records, e.g., for unsupervised synthetic data generation. It has an autoencoder and a vanilla GAN. MedGAN does not preserve privacy. We used an open-source implementation[1].
2. **TableGAN [13]:** TableGAN consists of three components: a generator, a discriminator, and an auxiliary classifier that aims to augment the semantic integrity of synthesized data. Similar to our method, TableGAN uses Convolutional Neural Networks. TableGAN requires labeled training data and will be used in experiments with a supervised setting. We used an open-source implementation[2].
3. **PATE-GAN [15]:** PATE-GAN proposes a differentially private model, based on a modification of the PATE mechanism. We utilized our own implementation of PATE-GAN as its code was not available publicly.

5.4. Unsupervised Synthetic Data Generation

We chose electronic health records after converting them to a high-dimensional binary discrete dataset to demonstrate the privacy-preserving potential of the proposed model. Here, we have $\mathcal{D}_{tr} \in \{0,1\}^{R\times|\mathcal{T}|}$, $\mathcal{D}_{te} \in \{0,1\}^{T\times|\mathcal{T}|}$, and $\mathcal{D}_{syn} \in \{0,1\}^{S\times|\mathcal{T}|}$, where $|\mathcal{T}|$ represents the feature size. The goal is to capture the information from a real private dataset, and use that to synthesize a dataset with similar distribution in a privacy-sensitive manner. In this section, we compare our method with models that do not require labeled data (for unsupervised synthetic data generation) and are also privacy-preserving (see Table 1). In the experiments of this section, to pretrain the autoencoder, we used Eq. (1) as the loss function.

Dataset Construction: We used the MIMIC-III dataset [41] as an example *high-dimensional* and *discrete* dataset. From MIMIC-III, we extracted and used 1071 unique ICD-9 codes. The dataset has different discrete variables (e.g., diagnosis codes, procedure, etc.). Assuming there are $|\mathbf{V}|$ discrete variables, we can represent the patient's record using a binary vector $\mathbf{v} \in \{0,1\}^{|V|}$. The i^{th} variable is set to one if it is available in the patient's record. In our experiments, $|\mathbf{V}| = 1071$.

Evaluation Metrics: To assess the quality of synthetically generated data, we used two evaluation metrics:

1. **Maximum Mean Discrepancy (MMD):** This indicates the extent to which the model captures the statistical distribution of the real data. In a recent study [47], MMD demonstrated most of the desired features of an evaluation metric, especially for GANs. MMD is used in an *unsupervised setting* since there are no labeled data for statistical measurements. To report MMD, we compared two samples of real and synthetic data, each of size 10000.
2. **Dimension-wise prediction:** This setting aims to illustrate the inter-connection between features, i.e., if we can predict missing features using the features available in the dataset. We select top-10 and top-50 most frequent features (ICD-9 codes in the patients' history) from training, test, and synthetic sets. In each run, one testing dimension (k) from $\mathcal{D}_{syn}$ and $\mathcal{D}_{tr}$ will be selected as $\mathcal{D}_{syn,k} \in \{0,1\}^{N\times1}$ and $\mathcal{D}_{tr,k} \in \{0,1\}^{N\times1}$. The remaining dimensions ($\mathcal{D}_{syn,\setminus k} \in \{0,1\}^{N\times1}$ and $\mathcal{D}_{tr,\setminus k} \in \{0,1\}^{N\times1}$) will be utilized to train a classifier that aims to predict $\mathcal{S}_{te,k} \in \{0,1\}^{N\times1}$ from the test set. We employed

[1] https://github.com/mp2893/medgan
[2] https://github.com/mahmoodm2/tableGAN

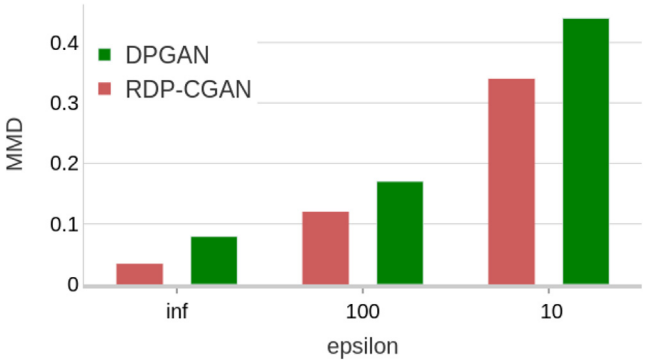


Fig. 2. The comparison of generated and real data distributions. Lower MMD score indicates a higher distribution similarity and hence, a better model.

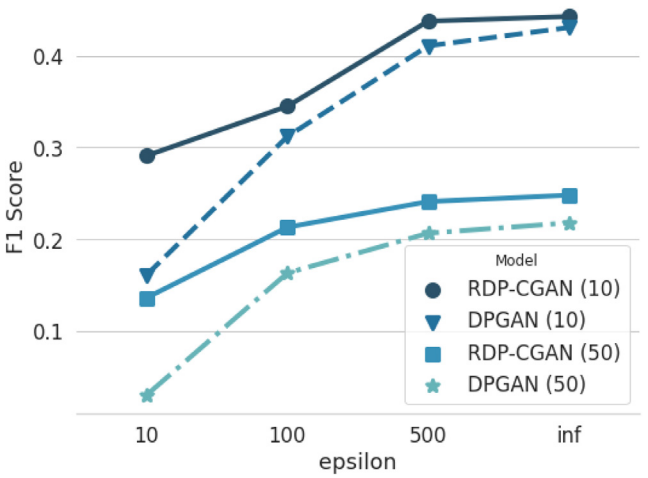


Fig. 3. The prediction accuracy of models trained on synthetic data, with varying privacy budget. The number of top features used for training are '10' and '50'. The averaged F-1 score of all models to train on the *real data* is *0.44* and *0.35* for picking top-10 and top-50 features, respectively.

Random Forests [48], XGBoost [49], and Decision Tree [50]. We report the averaged performance over all predictive models (trained on synthetic data) and for all features using the *F1-score* (Fig. 3). *We compare models by considering different privacy budgets by fixing* $\delta = 10^{-5}$ *and varying* ϵ.

As can be observed in Fig. 2 and Fig. 3, without enforcing privacy (ϵ), our model performs better compared to DPGAN. *This is due to the fact that by using 1-D convolutional architecture, our model captures the correlated information among adjacent ICD-9 codes in the MIMIC-III dataset. Top-features indicate the most frequent features in the database.*

To further investigate the effect of CGANs and CAEs in capturing correlated features, we conduct some experiments without enforcing any privacy and compare our method with existing methods. As can be observed in Table 2, our method outperforms other methods. Note that when the number of top features is increased, our method shows even better performance compared to other methods since capturing correlation for higher number of features becomes more feasible with convolutional layers.

5.5. Supervised Synthetic Data Generation

In this part, we consider a supervised setting and generate labeled synthetic data. For the experiments of this part, we need labeled data. The supervised setting here includes various classification tasks on different datasets. In the experiments of this section, to pretrain the autoencoder, we used Eq. (2) as the loss function.

Data Processing: For all the datasets except for MIMIC, we follow the same data processing steps that are described in Section 5.2. For the MIMIC-III dataset, we used one of the top three codes across hospital admissions [41], which is 414.01 (associated with *coronary atherosclerosis*) and extracted patients diagnosed with that specific medical code. The classification task is mortality prediction. We used 12,127 unique admissions and a total of 31 variables such as demographic information (marital status, ethnicity, and insurance status), admission information (e.g., days of admission), treatment information (cardiac defibrillator implant with/without cardiac catheterization), diagnostic information (respiratory disorder, cancer, etc.),

Table 2
Comparison of different methods using *F-1 score* for the dimension-wise prediction setting. Except for the *Real Data* column, the classifiers are trained on the synthetic data. The closer the results are to the real data experiments, the higher is the quality of synthetic data; hence the model is better.

Top Features	Real Data	MedGAN	DPGAN	RDP-CGAN
10	0.51 ± 0.043	0.41 ±0.013	0.36 ± 0.010	**0.44 ±0.017**
50	0.45 ± 0.037	0.24 ± 0.017	0.21 ± 0.017	**0.35 ±0.056**
100	0.36 ± 0.051	0.15 ± 0.016	0.12 ± 0.009	**0.25 ±0.096**

and lab results (kidney function tests, creatine kinase, etc.). Each admission is considered as one data sample. Since multiple admissions might be associated with one patient, we will have multiple samples per patient. This comes from the fact that, in a 1-year interval, a patient may survive, but within the next few years and with new medical conditions, the same patient may not survive in the 1-year observation window.

Evaluation: Let us consider the following two settings: *(A)* Train and test the model on the real data as the baseline (results shown in Table 3). *(B)* Train on the generated data and test on the real data (this setting is used to quantify the quality of synthetic data). *Note that the class distribution of the generated synthetic data must be identical to the real data*. Setting (B) can demonstrate how well the model has performed the task of synthetic data generation. *The closer the performance of setting (B) is to setting (A), the better the model is in terms of the quality of the generated synthetic data*. We conducted different sets of experiments. We conduct ten runs (E = 10), for each experiment, and report the averaged AUROC (Area Under the ROC curve) and *AUPRC* (Area Under the Precision-Recall Curve) for the models' evaluations. The AUPRC metric is being utilized here since it operates better than AUROC for an imbalanced classification setting.

5.5.1. The effect of architecture

In order to investigate the effect of convolutional GANs and convolutional autoencoders in our model, we perform some experiments with *no privacy enforcement*. The case $\epsilon = \infty$ corresponds to the setup where we do not enforce any privacy. The AUROC and AUPRC results are reported in Table 4 and Table 5, respectively. In the aforementioned tables, the best and second best results are indicated with **bold** and underline text, respectively. The classifiers are trained on the synthetic data. The closer the results are to the real data experiments, the higher is the quality of synthetic data; hence, we have a better model. As can be observed from those tables, for the challenging datasets where there is a mixture of continuous and discrete variables, our model outperforms others since it not only captures feature correlations but also utilizes convolutional autoencoders which allow our approach to learn robust representations for the mixture of data types. For other datasets, TableGAN closely competes with our model. This is because TableGAN also uses convolutional GANs as well as auxiliary classifiers [23] for improving the performance of the GANs. We could incorporate auxiliary classifiers to improve our model, however, it would narrow down the scope of our work since using such auxiliary models would make it infeasible for unsupervised synthetic data generation.

5.5.2. The effect of differential privacy

We next describe how well different methods perform in the differential privacy setting. *The base model is when we train using setting (A), which will have the highest accuracy as expected*. So the question we investigate here is: *How much will the accuracy drop for different models at the same level of privacy budget* (ϵ, δ)*?*We compare our model with privacy-preserving models (see Table 1). Table 6 shows these results for the supervised setting described above. In the majority of the experiments, the synthetic data generated by our model demonstrates higher quality for classification tasks compared to other models *under the same privacy budget*.

5.5.3. The effect of privacy budget

We also investigate how models will perform under different privacy budgets. Fig. 4 demonstrates the trade-off between the privacy budget and the synthetic data quality. RDP-CGAN consistently shows better performance compared to other benchmarks in all of the datasets. Our approach demonstrates it is particularly effective in lower privacy budgets. As an example, for the Kaggle Cervical Cancer dataset with $\epsilon = 100, 10, 1, 0.1$, our model achieves significantly higher AUPRC compared to the PATE-GAN.

5.5.4. Ablation study

We also describe an ablation study to investigate the effect of each component of our model. In particular, we are interested in assessing the importance and impact of utilizing autoencoders and convolutional architectures. The results of our ablation study are reported in Tables 7,8. The various scenarios reported are described below:

- *W/O CAE:* without convolutional architecture in autoencoder.
- *W/O AE:* without autoencoder.
- *W/O CG:* without convolutional architecture in GAN's generator.
- *W/O CD:* without convolutional architecture in GAN's discriminator.

Table 3
Summary statistics of the datasets used in this paper along with the performance of the base model.

Dataset	Samples	Features	Positive Labels	AUROC	AUPRC
MIMIC-III	12,127	31	3272 (27%)	0.81 ± 0.005	0.74 ±0.007
Kaggle Cervical Cancer	858	36	52 (6%)	0.94 ± 0.011	0.69 ±0.003
UCI Epileptic Seizure	11,500	178	2300 (20%)	0.97 ±0.008	0.94 ±0.014
PTB Diagnostic	14,552	118	10506 (62%)	0.97 ±0.006	0.96 ±0.003
Kaggle Cardiovascular Disease	70,000	14	35000 (50%)	0.80 ±0.017	0.78 ±0.011
MIT-BIHArrhythmia	109,444	188	18606 (17%)	0.94 ±0.006	0.89 ±0.014

Table 4
Performance comparison of different methods using AUROC metric under no privacy constraints.

Dataset	MedGAN	TableGAN	DPGAN	PATE-GAN	RDP-CGAN
MIMIC-III	0.71 ± 0.011	0.72 ± 0.017	0.71 ±0.014	0.70 ±0.007	**0.74 ±0.012**
Kaggle Cervical Cancer	0.89 ± 0.010	0.90 ± 0.012	0.90 ±0.016	0.89 ± 0.009	**0.92 ±0.009**
UCI Epileptic Seizure	0.87 ± 0.013	**0.91 ±0.019**	0.88 ±0.024	0.85 ± 0.014	0.90 ± 0.011
PTB Diagnostic	0.86 ± 0.018	**0.94 ±0.009**	0.88 ± 0.018	0.91 ± 0.006	0.93 ± 0.016
Kaggle Cardiovascular Disease	0.71 ± 0.031	0.73 ± 0.016	0.71 ± 0.019	0.72 ± 0.008	**0.76 ±0.014**
MIT-BIHArrhythmia	0.90 ± 0.021	0.89 ± 0.019	0.88 ± 0.012	0.86 ± 0.014	**0.90 ±0.009**

Table 5
Performance comparison of different methods using AUPRC metric under no privacy constraints.

Dataset	MedGAN	TableGAN	DPGAN	PATE-GAN	RDP-CGAN
MIMIC-III	0.68 ± 0.011	0.71 ± 0.007	0.69 ± 0.006	0.70 ± 0.019	**0.72 ±0.009**
Kaggle Cervical Cancer	0.55 ± 0.012	0.60 ± 0.019	0.59 ± 0.008	0.58 ± 0.020	**0.62 ±0.017**
UCI Epileptic Seizure	0.79 ± 0.013	**0.86 ±0.016**	0.82 ± 0.015	0.81 ± 0.012	0.84 ± 0.020
PTB Diagnostic	0.88 ± 0.006	**0.93 ±0.008**	0.90 ± 0.007	0.88 ± 0.014	0.92 ± 0.012
Kaggle Cardiovascular Disease	0.69 ± 0.021	0.73 ± 0.011	0.73 ± 0.021	0.72 ± 0.017	**0.75 ±0.009**
MIT-BIHArrhythmia	0.77 ± 0.016	0.85 ± 0.013	0.82 ± 0.008	0.81 ±0.005	**0.86 ±0.013**

Table 6
The comparison of different models under the $(1, 10^{-5})$-DP setting. For $\epsilon = \infty$, we used our RDP-CGAN model without enforcing privacy. We used synthetic data for training all models. The best and second best results are indicated with **bold** and underline text, respectively.

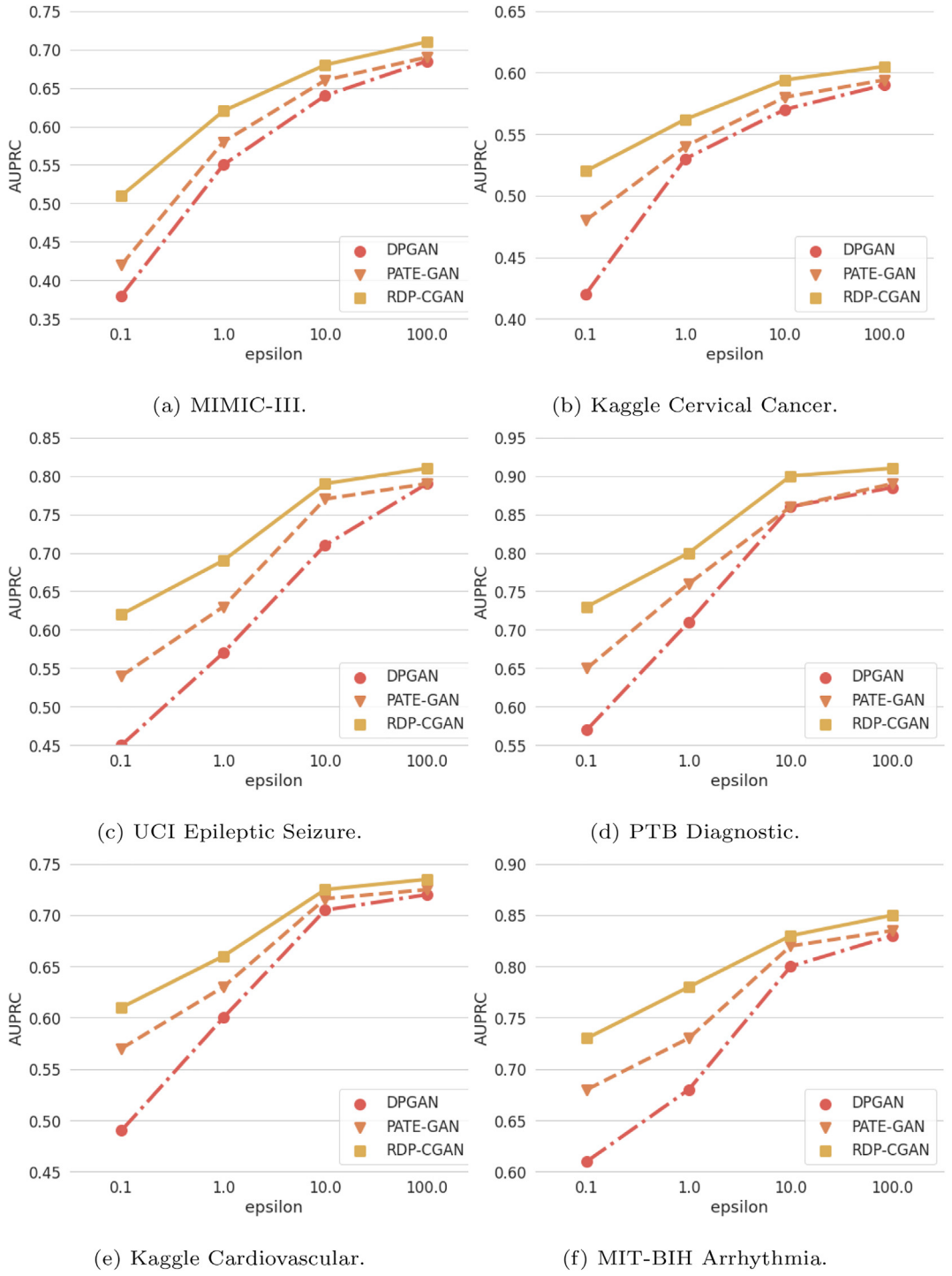
	AUROC				AUPRC			
Dataset	$\epsilon = \infty$	DPGAN	PATE-GAN	RDP-CGAN	$\epsilon = \infty$	DPGAN	PATE-GAN	RDP-CGAN
MIMIC-III	0.74 ± 0.012	0.59 ± 0.009	0.62 ± 0.011	**0.67 ±0.010**	0.72 ± 0.009	0.55 ± 0.013	0.58 ± 0.017	**0.62 ±0.015**
Kaggle Cervical Cancer	0.92 ± 0.009	0.86 ± 0.009	**0.91 ±0.005**	0.89 ± 0.005	0.62 ± 0.017	0.53 ± 0.008	0.54 ± 0.014	**0.57 ±0.007**
UCI Epileptic Seizure	0.90 ± 0.011	0.72 ± 0.009	0.74 ± 0.014	**0.84 ±0.008**	0.84 ± 0.020	0.57 ± 0.003	0.63 ± 0.016	**0.69 ±0.019**
PTB Diagnostic ECG	0.93 ± 0.016	0.71 ± 0.012	0.75 ± 0.012	**0.79 ±0.009**	0.92 ± 0.012	0.71 ± 0.018	0.76 ± 0.011	**0.80 ±0.008**
Kaggle Cardiovascular	0.76 ± 0.014	0.61 ± 0.019	0.66 ± 0.006	**0.69 ±0.013**	0.75 ± 0.009	0.60 ± 0.001	0.63 ± 0.007	**0.66 ±0.016**
MIT-BIHArrhythmia	0.90 ± 0.009	0.69 ± 0.004	0.73 ± 0.006	**0.77 ±0.003**	0.86 ± 0.013	0.68 ± 0.023	0.73 ± 0.016	**0.78 ±0.008**

- *W/O CDCG:* a standard GAN architecture with MLP.

The results shown in Table 7 and Table 8 demonstrate the importance of convolutional architectures. The interesting finding here is that, for the datasets with only continuous values like UCI Epileptic Seizure, PTB Diagnostic ECG, and MIT-BIHArrhythmia (as opposed to a mixture of continuous-discrete variables), the use of autoencoders actually downgraded the system performance. Hence, we can conclude that autoencoders are very useful in the presence of discrete data or a mixture of both continuous and discrete data types.

6. Conclusion

In this work, we proposed and developed a differentially private framework for synthetic data generation using Rényi Differential Privacy. The model aimed to capture the temporal information and feature correlation using convolutional neural networks. We empirically demonstrated that by employing convolutional autoencoders we can effectively handle variables that are continuous, discrete, or a mixture of both. We argue that we will need to secure both encoder and decoder parts of



(a) MIMIC-III. (b) Kaggle Cervical Cancer.

(c) UCI Epileptic Seizure. (d) PTB Diagnostic.

(e) Kaggle Cardiovascular. (f) MIT-BIH Arrhythmia.

Fig. 4. The effect of privacy budget on the synthetic data quality measured by AUPRC. The baseline is associated with $\epsilon = \infty$ for which we trained each model with no privacy constraint (see Table 5). Higher AUPRC is associated with higher quality of synthetic data and hence represents. a better model.

the autoencoder since there is no guarantee that the autoencoder's training is being done by a trusted third-party. We show that our model outperforms other models under the same privacy budget. This phenomenon may come in part from reporting a tighter bound and, in part, from utilizing convolutional networks. We reported the performance of different models by replacing real data with synthetic data for training our machine learning models.

Table 7
Ablation study results. For the baseline $\epsilon = \infty$, we used our RDP-CGAN model without enforcing privacy. Each column corresponds to a change in the core RDP-CGAN architecture. The reported results are AUROC.

Dataset	$\epsilon = \infty$	W/O CAE	W/O AE	W/O CG	W/O CD	W/O CDCG
MIMIC-III	0.74 ± 0.012	0.73 ± 0.017	0.69 ± 0.031	0.72 ± 0.023	0.71 ± 0.027	0.70 ± 0.016
Kaggle Cervical Cancer	0.92 ± 0.009	0.90 ± 0.019	0.86 ± 0.027	0.92 ± 0.031	0.90 ± 0.014	0.88 ± 0.021
UCI Epileptic Seizure	0.90 ± 0.011	0.89 ± 0.016	0.91 ± 0.021	0.88 ± 0.022	0.86 ± 0.008	0.85 ± 0.026
PTB Diagnostic ECG	0.93 ± 0.016	0.91 ± 0.011	0.94 ± 0.015	0.92 ± 0.029	0.91 ± 0.027	0.89 ± 0.014
Kaggle Cardiovascular	0.76 ± 0.014	0.74 ± 0.012	0.71 ± 0.041	0.43 ± 0.015	0.72 ± 0.021	0.71 ± 0.026
MIT-BIHArrhythmia	0.90 ± 0.009	0.88 ± 0.013	0.90 ± 0.014	0.88 ± 0.012	0.86 ± 0.017	0.84 ± 0.041

Table 8
Ablation study results. For the baseline $\epsilon = \infty$, we used our RDP-CGAN model without enforcing privacy. Each column corresponds to a change in the core RDP-CGAN architecture. The reported results are AUPRC.

Dataset	$\epsilon = \infty$	W/O CAE	W/O AE	W/O CG	W/O CD	W/O CDCG
MIMIC-III	0.72 ±0.009	0.70 ±0.032	0.67 ± 0.054	0.70 ± 0.076	0.69 ± 0.023	0.66 ± 0.065
Kaggle Cervical Cancer	0.62 ± 0.017	0.61 ± 0.031	0.58 ± 0.017	0.59 ± 0.036	0.59 ± 0.041	0.55 ± 0.023
UCI Epileptic Seizure	0.84 ± 0.020	0.82 ± 0.034	0.86 ± 0.041	0.79 ± 0.024	0.81 ± 0.012	0.80 ± 0.018
PTB Diagnostic ECG	0.92 ± 0.012	0.90 ± 0.028	0.94 ± 0.031	0.89 ± 0.022	0.88 ± 0.022	0.88 ± 0.029
Kaggle Cardiovascular	0.75 ± 0.009	0.73 ± 0.052	0.71 ± 0.017	0.72 ± 0.033	0.71 ± 0.041	0.70 ± 0.014
MIT-BIHArrhythmia	0.86 ± 0.013	0.83 ± 0.019	0.87 ± 0.026	0.85 ± 0.038	0.84 ± 0.023	0.82 ± 0.032

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported in part by the US National Science Foundation grants IIS-1619028, IIS-1707498, and IIS-1838730, and by NVIDIA Corp. We also acknowledge the support provided by NewWave Telecom Technologies during the early stages of this project through a fellowship award.

References

[1] A. Narayanan, V. Shmatikov, Robust de-anonymization of large sparse datasets, in: 2008 IEEE Symposium on Security and Privacy (SP2008), IEEE, 2008, pp. 111–125..
[2] M. Al-Rubaie, J.M. Chang, Privacy-preserving machine learning: Threats and solutions, IEEE Security & Privacy 17 (2) (2019) 49–58.
[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Nets, Advances in Neural Information Processing Systems (2014) 2672–2680.
[4] J. Hayes, L. Melis, G. Danezis, E. De Cristofaro, LOGAN: Membership inference attacks against generative models, Proceedings on Privacy Enhancing Technologies 2019 (1) (2019) 133–152.
[5] C. Dwork, A. Roth, et al, The algorithmic foundations of differential privacy, Foundations and Trends in Theoretical Computer Science 9 (3–4) (2014) 211–407.
[6] R. Shokri, V. Shmatikov, Privacy-preserving deep learning, in: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, ACM, 2015, pp. 1310–1321.
[7] M. Abadi, A. Chu, I. Goodfellow, H.B. McMahan, I. Mironov, K. Talwar, L. Zhang, Deep learning with differential privacy, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2016, pp. 308–318.
[8] L. Xie, K. Lin, S. Wang, F. Wang, J. Zhou, Differentially private generative adversarial network, arXiv preprint arXiv:1802.06739..
[9] G. Acs, L. Melis, C. Castelluccia, E. De Cristofaro, Differentially private mixture of generative neural networks, IEEE Transactions on Knowledge and Data Engineering 31 (6) (2018) 1109–1121.
[10] C. Dwork, F. McSherry, K. Nissim, A. Smith, Calibrating noise to sensitivity in private data analysis, Theory of Cryptography Conference, Springer (2006) 265–284.
[11] I. Mironov, Rényi Differential Privacy, in: 2017 IEEE 30th Computer Security Foundations Symposium (CSF), IEEE, 2017, pp. 263–275..
[12] E. Choi, S. Biswal, B. Malin, J. Duke, W.F. Stewart, J. Sun, Generating multi-label discrete patient records using generative adversarial networks, arXiv preprint arXiv:1703.06490..
[13] N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, Y. Kim, Data synthesis based on generative adversarial networks, Proceedings of the VLDB Endowment 11 (10) (2018) 1071–1083.
[14] A. Torfi, E.A. Fox, CorGAN: Correlation-capturing convolutional generative adversarial networks for generating synthetic healthcare records, in: The Thirty-Third International FLAIRS Conference, 2020..
[15] J. Jordon, J. Yoon, M. van der Schaar, PATE-GAN: Generating synthetic data with differential privacy guarantees, in: International Conference on Learning Representations, 2018..
[16] M.K. Baowaly, C.-C. Lin, C.-L. Liu, K.-T. Chen, Synthesizing electronic health records using improved generative adversarial networks, Journal of the American Medical Informatics Association 26 (3) (2018) 228–241.
[17] A.H. Pollack, T.D. Simon, J. Snyder, W. Pratt, Creating synthetic patient data to support the design and evaluation of novel health information technology, Journal of Biomedical Informatics 95 (2019) 103201.
[18] J. Guan, R. Li, S. Yu, X. Zhang, Generation of synthetic electronic medical record text, in: in: 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2018, pp. 374–380.

[19] J. Walonoski, M. Kramer, J. Nichols, A. Quina, C. Moesel, D. Hall, C. Duffett, K. Dube, T. Gallagher, S. McLachlan, 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) (2017) 230–238.
[20] J. Chen, D. Chun, M. Patel, E. Chiang, J. James, 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) (2019) 44.
[21] L. Xu, M. Skoularidou, A. Cuesta-Infante, K. Veeramachaneni, Modeling tabular data using conditional GAN, Advances in Neural Information Processing Systems (2019) 7333–7343.
[22] B.K. Beaulieu-Jones, Z.S. Wu, C. Williams, R. Lee, S.P. Bhavnani, J.B. Byrd, C.S. Greene, Privacy-preserving generative deep neural networks support clinical data sharing, Circulation: Cardiovascular Quality and Outcomes 12 (7) (2019) e005122.
[23] A. Odena, C. Olah, J. Shlens, Conditional image synthesis with auxiliary classifier GANs, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, pp. 2642–2651..
[24] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, K. Talwar, Semi-supervised knowledge transfer for deep learning from private training data, arXiv preprint arXiv:1610.05755..
[25] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, Ú. Erlingsson, Scalable Private Learning with PATE, arXiv preprint arXiv:1802.08908..
[26] R. Torkzadehmahani, P. Kairouz, B. Paten, Dp-cgan: Differentially private synthetic data and label generation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019, pp. 0–0..
[27] U. Tantipongpipat, C. Waites, D. Boob, A.A. Siva, R. Cummings, Differentially private mixed-type data generation for unsupervised learning, arXiv preprint arXiv:1912.03250..
[28] C. Dwork, G.N. Rothblum, S. Vadhan, Boosting and differential privacy, in: 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, IEEE, 2010, pp. 51–60..
[29] A. Rényi, et al., 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, The Regents of the University of California, 1961..
[30] R.D. Hjelm, A.P. Jacob, T. Che, A. Trischler, K. Cho, Y. Bengio, Boundary-seeking generative adversarial networks, arXiv preprint arXiv:1702.08431..
[31] D.P. Kingma, M. Welling, Auto-encoding Variational Bayes, arXiv preprint arXiv:1312.6114..
[32] N.C. Abay, Y. Zhou, M. Kantarcioglu, B. Thuraisingham, L. Sweeney, Privacy preserving synthetic data release using deep learning, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2018, pp. 510–526..
[33] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A.C. Courville, Improved training of Wasserstein GANs, Advances in Neural Information Processing Systems (2017) 5767–5777.
[34] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein GAN, arXiv preprint arXiv:1701.07875..
[35] C. Villani, Optimal transport: old and new, Vol. 338, Springer Science & Business Media, 2008..
[36] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification, in: in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.
[37] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434..
[38] I. Mironov, K. Talwar, L. Zhang, Rényi differential privacy of the sampled Gaussian mechanism, arXiv preprint arXiv:1908.10530..
[39] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980..
[40] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167..
[41] A.E. Johnson, T.J. Pollard, L. Shen, H.L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L.A. Celi, R.G. Mark, MIMIC-III, a freely accessible critical care database, Scientific Data 3 (2016) 160035.
[42] K. Fernandes, J.S. Cardoso, J. Fernandes, Transfer learning with partial observability applied to cervical cancer screening, in: Iberian Conference on Pattern Recognition and Image Analysis, Springer, 2017, pp. 243–250.
[43] R.G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, C.E. Elger, Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state, Physical Review E 64 (6) (2001) 061907.
[44] R. Bousseljot, D. Kreiseler, A. Schnabel, Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet, Biomedizinische Technik/Biomedical Engineering 40 (s1) (1995) 317–318.
[45] S. Ulianova, Cardiovascular disease dataset, data retrieved from the Kaggle dataset, url:https://www.kaggle.com/sulianova/cardiovascular-disease-dataset (2018)..
[46] G.B. Moody, R.G. Mark, The impact of the MIT-BIH arrhythmia database, IEEE Engineering in Medicine and Biology Magazine 20 (3) (2001) 45–50.
[47] Q. Xu, G. Huang, Y. Yuan, C. Guo, Y. Sun, F. Wu, K. Weinberger, An empirical study on evaluation metrics of generative adversarial networks, arXiv preprint arXiv:1806.07755..
[48] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32.
[49] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[50] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1) (1986) 81–106.