GENERATING SYNTHETIC HEALTHCARE RECORDS USING CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS

Amirsina Torfi and Mohammadreza Beyki *
Department of Computer Science
Virginia Tech
Blacksburg, VA 24060, USA
{atorfi, mohibeyki}@vt.edu

ABSTRACT

Deep learning models have demonstrated high-quality performance in several areas such as image classification and speech processing. However, creating a deep learning model using electronic health record (EHR) data requires addressing particular privacy challenges that make this issue unique to researchers in this domain. This matter focuses attention on generating realistic synthetic data to amplify privacy. Existing methods for artificial data generation suffer from different limitations such as being bound to particular use cases. Furthermore, their generalizability to real-world problems is controversial regarding the uncertainties in defining and measuring key realistic characteristics. Henceforth, there is a need to establish insightful metrics (and to measure the validity of synthetic data), as well as quantitative criterion regarding privacy restrictions. We propose the use of Generative Adversarial Networks to help satisfy requirements for realistic characteristics and acceptable values of privacy metrics simultaneously. The present study makes several unique contributions to synthetic data generation in the healthcare domain. First, utilizing 1-D Convolutional Neural Networks (CNNs), we devise a new approach to capturing the correlation between adjacent diagnosis records. Second, we employ convolutional autoencoders to map the discrete-continuous values. Finally, we devise a new approach to measure the similarity between real and synthetic data, and a means to measure the fidelity of the synthetic data and its associated privacy risks.

1 INTRODUCTION

The embracing of Electronic Health Records (EHRs), in addition to the availability of massive amounts of data, has led to calls for employing promising data-driven methods inspired by *Artificial Intelligence* (AI). Data-powered tools alter the way that clinicians and healthcare bureaus approach and satisfy patients' needs by providing care. However, extending this broad EHR adoption to also support data access for research and development purposes, is far from being practical in the healthcare domain, due to privacy restrictions.

De-identification of EHR data is often employed for mitigation of privacy risks. However, questions and doubts have increased about the safety of prolonged use of de-identification methods regarding their vulnerability to information leakage. To date, there has been little agreement on what sort of data protection methods can reliably satisfy concerns for privacy. This issue has grown in importance in light of recent demands for large volumes of healthcare data to support research.

Accordingly, more recent attention has focused on Synthetic Data Generation (SDG), which may satisfy the privacy necessities immediately and reliably. Despite recent advancements in SDG, research on the subject has been mostly restricted to limited use cases. Moreover, in the majority of the existing SDG frameworks, the utilized backend data is small, and there has been no reliable evidence that the synthetic data can reliably replace the real data for research purposes. A possible

*Equally contributed.

solution is to leverage the massive amount of actual data, but most likely that is impractical due to privacy restrictions. Additionally, little is known about the generalizability of research carried out using synthetic data in practical scenarios, and it is not clear what factors determine how realistic are the results of using synthetic data.

One approach is the utilization of deep learning methods, especially generative models, which are expected to capture the inherent characteristics of data. This project seeks to remedy current problems based on analyzing the literature of *Generative Models* for healthcare-related synthetic data generation. Such an approach could potentially help with satisfying privacy concerns.

The main contributions of this work are as follows:

- We propose a novel architecture to generate synthetic healthcare records by utilizing Convolutional GANs which we call *"corGAN"*.
- By employing statistical method, we demonstrate the effectiveness of utilizing Convolutional Neural Networks (CNNs) instead of Multilayer Perceptrons to capture the feature correlation.
- We propose a novel approach to measure the similarity of real and synthetically generated records.
- We also assess the model's privacy using *Membership Inference Attack*.

2 Related Works

Here we address a variety of methods that were utilized for synthetic healthcare data generation, which is the primary focus of this work.

Some studies were conducted in a variety of domains about synthetic data generation (Walonoski et al., 2017; Buczak et al., 2010; McLachlan et al., 2016; Park et al., 2013). Many of these methods are disease-specific, not realistic, or have failed to provide any substantial proof of privacy. For instance, in Buczak et al. (2010) a data-driven method is proposed to generate synthetic EHR data specifically for a single illness (tularemia); further, it may be vulnerable to re-identification attacks. The Synthea framework (Walonoski et al., 2017) provides EHR data for research and educational practices; however, there is no clear roadmap to expand it to different use-cases. In other words, as the backend of the framework cannot be easily modified, it is impractical to broaden its scope to unsupported or new needs. In practice, the generated synthetic EHR data are usually not adequately realistic for predictive analysis using machine learning (Choi et al., 2017). Further, it is important to verify the usability of synthetic data for related applications.

It is crucial to investigate whether the realism of the synthetic data can lead to a violation of privacy restrictions. In the literature, there are insufficient comprehensive definitions related to privacy. Although a perfectly de-identified dataset may appear to satisfy privacy considerations, however, unlike what is claimed in Baowaly et al. (2018), the disconnectivity of individuals' personal information and their health-related data (such as encounter-level diagnosis and test results) via de-identification cannot guarantee privacy (Choi et al., 2017; El Emam et al., 2011). Consequently, there is a need to define concrete metrics for privacy, especially when synthetic data is generated using real data.

GANs have been successfully employed in various applications such as image generation (Reed et al., 2016; Brock et al., 2018; Karras et al., 2018), video generation (Vondrick et al., 2016; Tulyakov et al., 2018), image translation (Isola et al., 2017; Kim et al., 2017a; Dong et al., 2017), etc. We aim to create realistic synthetic EHR data by GANs. Recently, many GAN models were developed to generate medical records. The most recent attention has focused on natural language processing. Choi et al. (2017) undertook preliminary work on using GANs for synthetic data generation called "medGAN" which generates discrete variables via a mixture of autoencoders and generative adversarial networks. However, they do not consider the temporal nature of the data.

GANs have trouble handling and generating discrete data. In fact, GANs are designed to produce continuous variables. Despite the impressive performance of GANs regarding continuous data, it remains challenging to create discrete data with GANs, which restricts its applicability in domains such as NLP. In particular, this drawback becomes troublesome while dealing with multi-variable discrete EHR data, and new strategies must be devised. Recently, new methods such as (Hjelm et al.,

2017; Kim et al., 2017b) were proposed to handle generating discrete data with GANs. Particularly regarding EHR data generation, we aim for a framework with the ability to handle discrete variables and preserve privacy as well.

3 DATA DESCRIPTION

At first, we have to identify how many discrete variables (e.g., medical codes) are available in the dataset. Let's assume there are $|\boldsymbol{D}|$ discrete variables and a vector space with $\boldsymbol{V_C} \in \boldsymbol{N_0}^{|\boldsymbol{D}|}$ (where N_0 indicates natural numbers including zero). The j^{th} dimension designates the number of incidents of the j^{th} variable in a subject's medical records. We can represent a patient visit (encounter event) by a binary vector $\boldsymbol{V_B} \in \boldsymbol{\{0,1\}}^{|\boldsymbol{D}|}$, where the j^{th} dimension shows if the j^{th} variable occurred in the patient record.

4 PRELIMINARY - GENERATIVE ADVERSARIAL NETWORKS

Generative Adversarial Networks (GANs) were introduced in Goodfellow et al. (2014). A GAN is a combination of two neural networks, a discriminator and a generator. The whole network is trained in an iterative process. First, the generator network produces a fake sample. Then the discriminator network tries to determine whether this sample (ex.: an input image) is real or fake, i.e., whether it came from the real training data or not. The goal of the generator is to fool the discriminator so it believes the artificial (i.e., generated) samples created by the generator are real.

The iterative process continues until the generator produces samples that are indistinguishable by the discriminator. In other words, the probability of classifying a sample as fake or real becomes like flipping a fair coin for the discriminator. The goal of the generative model is to capture the distribution of real data while the discriminator tries to identify the fake data.

The generator wants to learn the distribution p_g over data $\mathbf{x}$. In that regard, $p_z(\mathbf{z})$ represents the input noise variables distribution which generates random data shown by $G(\mathbf{z}; \theta_g)$. The function G is differentiable with parameters θ_g. The discriminator is defined as $D(\mathbf{x}; \theta_d)$ which decides if its input data is real or fake. D is trained to distinguish the training examples and samples from G by minimizing $\log(1 - D(G(z))$. D and G actually perform the following min-max game with the value function V(G, D):

$$\underset{G}{Min}\underset{D}{Max}V(G,D) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[1 - \log D(G(x))] \tag{1}$$

5 ARCHITECTURE

5.1 BASELINES

GANs have been demonstrated to generate realistic samples. GANs, however, have a strict restriction on the kind of variables they can handle because they need the combination of the generator and discriminator to be differentiable. With discrete variables, this is a problem. For example, think about using a step function to generate discrete values. In this instance, back-propagation fails as the derivative of a step function is always zero. Discrete variables are a problem in many different applications. In particular, in the majority of healthcare applications, we have to deal with discrete data. To solve this, we started with *medGAN* (Choi et al., 2017). In this setting, an autoencoder is used to trasnform the continuous variables generated by the GAN, to their associated discrete variables.

The architecture of the baseline models (Fig. 1) includes regular *Multilayer Perceptrons* (MLPs) which are used for both discriminator and generator. Such a baseline architecture is similar to *medGAN*, by Choi et al. (2017).

The discrete input $\boldsymbol{X}$ represents the source EHR data; $\boldsymbol{z}$ is the random distribution for the generator $\boldsymbol{G}$; $\boldsymbol{G}$ is the employed neural network architecture; and $\boldsymbol{Dec(G(z))}$ refers to the decoding function, which is used to transform the generator $\boldsymbol{G}$ continuous output to their equivalent discrete

values. The discriminator $\boldsymbol{D}$ attempts to distinguish real input $\boldsymbol{X}$ from the discrete synthetic output $\boldsymbol{Dec(G(z))}$. For the generator and the discriminator, a 1D Convolutional GAN architecture is utilized.

Now, let's discuss the decoding function $\boldsymbol{Dec(G(z))}$. GANs are known for generating continuous values and encountering trouble when dealing with discrete variables. Recently, some research efforts proposed solutions to the problem of generating discrete variables (Hjelm et al., 2017; Wang et al., 2017; Kim et al., 2017b; Yu et al., 2017), such as for EHR data generation. The indirect method creates a separate model to transform continuous to equivalent discrete data (Choi et al., 2017). Alternatively, our generative model could generate discrete data directly. Here, we choose the transform approach and propose the use of autoencoders.

Considering Fig. 1, the autoencoder digests (right part of the figure) discrete values as the input and reconstructs the same input in the output. The autoencoder structure consists of two main elements: encoder and decoder. While encoding, the autoencoder transforms the discrete space into a corresponding (we call it equivalent as well) continuous space (the output of the hidden layer), and the decoder transforms the continuous values to their equivalent discrete values all over again. The *Binary Cross-Entropy (BCE)* loss function is used for training the autoencoder.

$$BCE_{loss} = -\frac{1}{N}\sum_{i=1}^{N} x_i \log(y_i) + (1 - x_i)\log(1 - y_i) \tag{2}$$

$$y_i = \boldsymbol{Dec}(\boldsymbol{Enc}(x_i)) \tag{3}$$

After training the autoencoder, all we need is to use its decoder to convert continuous values to their associated discrete values. Henceforth, the cost function to train our proposed architecture is similar to Eq. 1 with the exception of operating the decoder on top of the generator's output.

$$\underset{G}{Min}\underset{D}{Max}V(G,D) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[1 - \log D(\boldsymbol{Dec}(G(x)))] \tag{4}$$

5.2 1D Convolutional GANs

We used the architecture of Fig. 1, substituting CNNs for MLPs.

As we are dealing with 1D data, *we chose the 1D CAEs as a particular case of the regular AEs*. This approach enables us to capture the neighboring feature correlations without disregarding the locality characteristics of the features. We call our proposed architecture *corGAN*.

6 Experiments

6.1 Setup

For experiments with binary discrete variables, we used a publicly available dataset: *MIMIC-III* dataset (Johnson et al., 2016), consisting of the medical records of almost 46K patients, from which we extracted ICD-9 codes only. We aggregate a patient record into a vector of size 1071, since there are 1071 unique ICD-9 codes in this dataset.

The implemented models are summarized in Table 1, while the architecture is depicted in Fig 1. MD, MA, and BN are abbreviations of Minibatch Averaging (Choi et al., 2017), Minibatch Discrimination (Salimans et al., 2016), and Batch Normalization (Ioffe & Szegedy, 2015), respectively. The MLP architecture consists of two fully connected layers with sizes of 256 and 128, respectively. We have chosen this layer size to have a consistent architecture size among all baseline models. We add shortcut connections for preventing the vanishing gradient phenomena (He et al., 2016).

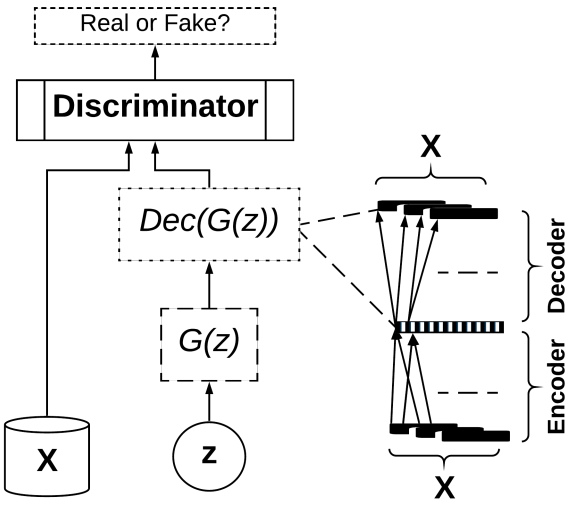

Figure 1: The architecture for generating synthetic data from real samples. The right side of the figure shows the pretrained convolutional autoencoder. Its decoder part is being used to transform the generated continuous samples to their discrete equivalents.

Table 1: Comparison of different baseline architectures.

Name	Decoder (Pretrained)	Generator	Technique
GAN	Autoencoder (NO)	MLP	Regular Training
GAN_{pre}	Autoencoder (YES)	MLP	Regular Training
GAN_{pre}	Autoencoder (YES)	MLP	MD
$medGAN$ (Choi et al., 2017)	Autoencoder (YES)	MLP	MA + BN
$\boldsymbol{corGan[Ours]}$	Autoencoder (YES)	1-D CNN	MD + BN

6.2 Implementation detail

We implemented our model with PyTorch 1.3 (Paszke et al., 2017). We used the Adam optimizer (Kingma & Ba, 2014) with a fixed learning rate of 0.001, and a mini-batch size of 1,000. Both the encoder and the decoder of the autoencoder are 1D convolutional networks, with the input compressed to a 256-dimensional vector. It is worth noting that, for the decoder, we used deconvolution operations (Noh et al., 2015) to rebuild the features. The generator and discriminator are both formed with 1D convolutional layers as well, with dimensions 256 and 128.

6.3 Mode Collapse

One of the primary crash forms for GAN is for the generator to collapse to a set of parameters and always generate the same sample. As the discriminator considers each sample independently, it cannot tell the output part of the generator to create outputs that are not identical. Instead, all generated outputs are a single point that the discriminator believes is the best. So, the generator only produces one or a few samples over and over.

Minibatch discrimination (Salimans et al., 2016), in general, is referring to the process of the discriminator model investigating multiple examples in succession, instead of individually, to help to avoid the collapse of the generator.

6.4 IMPROVEMENTS

We implemented several techniques to improve the training of our GAN architecture. To improve the power of our generative model, we shifted our focus to use Wasserstein GAN (Arjovsky et al., 2017). The *Earth-Mover (EM)* distance or Wasserstein-1 distance represents the minimum price in transforming the generated data distribution $\boldsymbol{P_g}$ to the real data distribution $\boldsymbol{P_r}$:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} E_{(x,y)\sim\gamma}\big[\, \|x - y\| \,\big] \,, \tag{5}$$

Here $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ is the set of all possible joint distributions $\gamma(x, y)$ whose marginals are $\mathbb{P}_r$ and $\mathbb{P}_g$. $\gamma(x, y)$ refers to how much "mass" should be moved from x to y to transform $\mathbb{P}_r$ to $\mathbb{P}_g$.

However, as the infimum in equation 5 is intractable, based on the the Kantorovich-Rubinstein duality (Villani, 2008), the authors of WGAN suggested:

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} E_{x\sim\mathbb{P}_r}[f(x)] - E_{x\sim\mathbb{P}_\theta}[f(x)] \tag{6}$$

Here the supremum is over all the 1-Lipschitz functions (Villani, 2008) $f : \mathcal{X} \to \mathbb{R}$. For simplicity of definition, infimum and supremum indicate the greatest lower bound and the least upper bound, respectively.

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:

$$\forall(x_1, x_2) \in \mathbb{X}, \exists K \in \mathbb{R} : d_Y(f(x_1), f(x_2)) \leq K d_X(x_1, x_2) \tag{7}$$

Using the distance metric and $K = 1$, Eq. 7 is equivalent to:

$$\forall(x_1, x_2) \in \mathbb{X} : |f(x_1) - f(x_2)| \leq |x_1 - x_2| \tag{8}$$

It is clear that for *computing the Wasserstein distance, we should find a 1-Lipschitz function*. 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 probability. To enforce the restriction, WGAN authors suggested applying an effortless clipping to limit the maximum weight value in f.

6.5 EVALUATION

We use the following two metrics to evaluate our synthetically generated data. We divided the real data into two sets of $\mathbb{S}_{tr}$ and $\mathbb{S}_{te}$. We use the set $\mathbb{S}_{tr}$ to train the system. After training the system, we generate the synthetic set $\mathbb{S}_{syn}$ as $|\mathbb{S}_{syn}| = |\mathbb{S}_{tr}|$.

- **Dimension-wise probability:** As a basic sanity check to see if our proposed models learned the distribution of the real data (for each dimension), we report the dimension-wise probability. This measurement refers to the Bernoulli success probability of each dimension (each dimension is a unique ICD-9 code).
- **Maximum Mean Discrepancy:** Overall, Maximum Mean Discrepancy (MMD) (Borgwardt et al., 2006) represents the notion of expressing the similarity between two distributions as the distance between mean feature embeddings. Assume the distributions P_R and P_G are defined over a set $\mathbb{X}$. We represent the MMD as a feature map $\Phi : \mathbb{X} \to \mathbb{H}$,where $\mathbb{H}$ designates the reproducing kernel Hilbert space. The MMD is as follows:

$$MMD(P_R, P_G) = \left\| E_{\mathbb{X}_\mathbb{R}\sim P_R[\Phi(\mathbb{X}_\mathbb{R})]} - E_{\mathbb{X}_\mathbb{G}\sim P_G[\Phi(\mathbb{X}_\mathbb{G})]} \right\|_\mathbb{H} \tag{9}$$

  The subscripts R and G are related to the ***Real*** and ***Generated*** data. As Φ maps to a generic reproducing kernel Hilbert space, the kernel trick can be applied to measure the MMD, which returns a more accurate comparison rather than merely sticking to the mean discrepancy. This approach is called Kernel Maximum Mean Discrepancy. The Parzen window estimate (Gretton et al., 2007) represents a specific case of the Kernel MMD.

6.5.1 MAXIMUM MEAN DISCREPANCY

Table 2 demonstrates the results. The results are reported based on the mean and standard deviation of five independent experiments. In each experiment a new set of synthetic data is generated and utilized. The baseline compares *real-real* samples and ideally should give *zero* as the score. However, as you can see in Table 2, even the baseline does not give zero score. This is due to the finite-sample nature of our measurements.

Table 2: Distinguishing between real and synthesized samples by employing Maximum Mean Discrepancy metric.

Name	Score
GAN	0.0064 ± 0.00035
GAN_{pre}	0.0048 ± 0.00022
GAN_{pre+mb}	0.0043 ± 0.00018
$medGAN$ (Choi et al., 2017)	0.0032 ± 0.00021
$WGAN$ (Arjovsky et al., 2017)	0.0018 ± 0.00024
$\boldsymbol{corGan}$ $[\boldsymbol{Ours}]$	$\mathbf{0.0008 \pm 0.00015}$

6.5.2 DIMENSION-WISE PROBABILITY

The results are provided in Fig. 2. The points in the scatter plot are close to the $\boldsymbol{y} = \boldsymbol{x}$ line. This demonstrates the similarity of the distribution of both real and synthetic data. We only reported the three best models; in terms of dimension-wise probability, they perform similarly. The WGAN is similar to the corGAN. The difference is: in WGAN we used another loss function as described in Section 6.4.

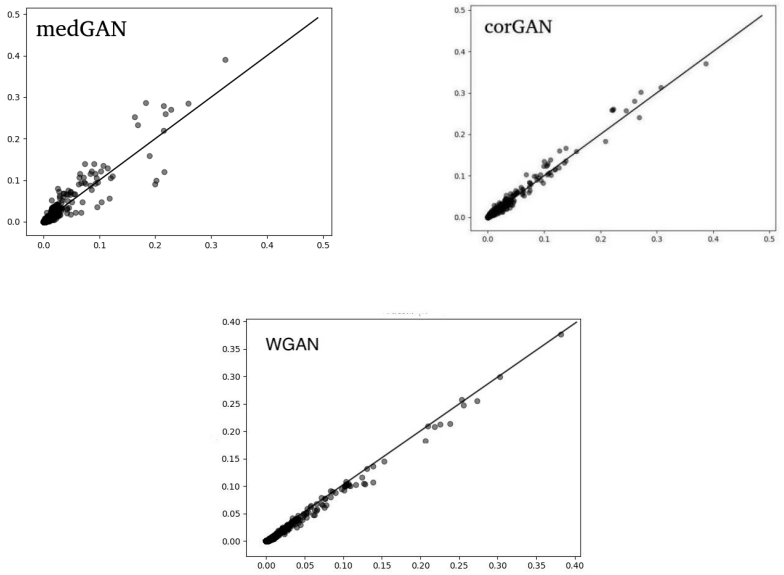

Figure 2: The scatter plots of dimension-wise probability. Each point depicts one of 1071 unique diagnosis codes. The x-axis and y-axis represent the Bernoulli success probability for real and synthetic datasets, respectively. The diagonal line shows the ideal case.

7 DISCRIMINATIVE MODEL

The question is: *How can we train a discriminative model to distinguish between real and fake data?* The main point of this approach is that the *trained discriminator of the GAN architecture cannot be used for evaluation purposes* as it is supposed to be fooled completely. We propose to ***train another independent discriminator using the real and synthesized data***.

7.1 SIAMESE ARCHITECTURE

The discriminative model uses a Siamese architecture, Lecun et al. (2005), which consists of two identical neural networks. The goal is to create a target feature subspace for discrimination between similar and dissimilar pairs based on a simple distance metric. The model is depicted in Fig. 3. The general idea is that when two samples belong to a genuine pair, their distance in the target feature subspace should be as close as possible, while the impostor images should be as far apart as possible. Let X_{p_1} and X_{p_2} be a pair of samples as the inputs of the system whether in training or testing mode. The distance between a pair of samples in the target subspace is defined as $D_W(X_{p_1}, X_{p_2})$ (i.e., the $\ell_2 - norm$ between two vectors) in which W is the parameters of the whole network (weights). Stated more simply, $D_W(X_{p_1}, X_{p_2})$ should be low for genuine pairs and should be high for impostor pairs; this defines the contrastive loss function. Consider Y as the label which is considered to be 1 if both samples are genuine and 0 otherwise. F is the network function which maps the input to the target feature subspace.

The outputs of the Siamese CNNs are denoted by $F_W(X_{p_1})$ and $F_W(X_{p_2})$, where W is the same because both CNNs share the same weights. The distance is computed as:

$$D_W(X_{p_1}, X_{p_1}) = ||F_W(X_{p_1}) - F_W(X_{p_2})||_2. \tag{10}$$

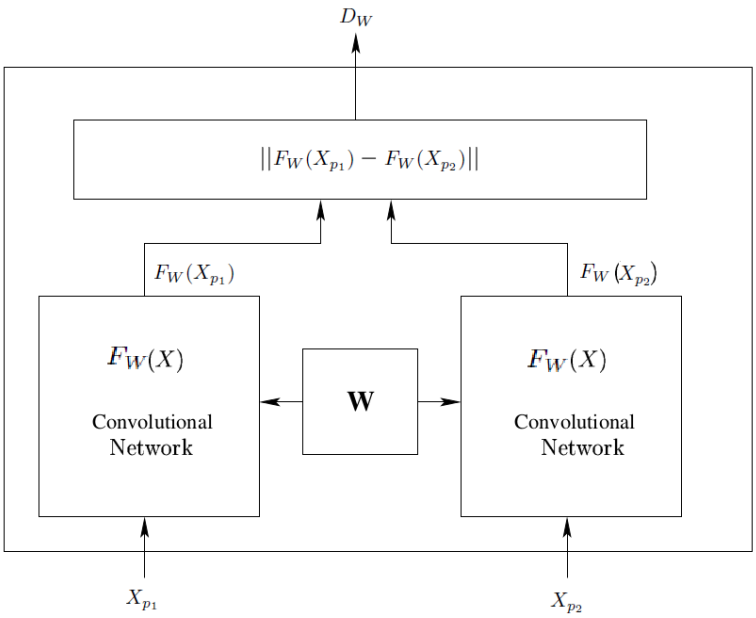

Figure 3: Siamese Model Framework

7.2 CONTRASTIVE COST

The goal of the loss function $L_W(X, Y)$ is to minimize the loss in both scenarios of encountering genuine and impostor pairs, so the definition should satisfy both conditions as given by:

$$L_W(X, Y) = \frac{1}{N} \sum_{i=1}^{N} L_W(Y_i, (X_{p_1}, X_{p_2})_i), \tag{11}$$

where N is the number of training samples, i is the index of each sample, and $L_W(Y_i, (X_{p_1}, X_{p_2})_i)$ is defined:

$$\begin{aligned} L_W(Y_i, (X_{p_1}, X_{p_2})_i) = Y * L_{gen}(D_W(X_{p_1}, X_{p_2})_i) \\ + (1 - Y) * L_{imp}(D_W(X_{p_1}, X_{p_2})_i) + \lambda ||W||_2 \end{aligned} \tag{12}$$

in which the last term is for regularization and λ is the regularization parameter. Finally, L_{gen} and L_{imp} are defined as the functions of $D_W(X_{p_1}, X_{p_2})$ by the following equations:

$$\begin{cases} L_{gen}(D_W(X_{p_1}, X_{p_2})) = \frac{1}{2} D_W(X_{p_1}, X_{p_2})^2 \\ L_{imp}(D_W(X_{p_1}, X_{p_2})) = \frac{1}{2} max\{0, (M - D_W(X_{p_1}, X_{p_2}))\}^2 \end{cases} \tag{13}$$

where M is a margin which is obtained by cross-validation. Moreover the max argument declares that in case of an impostor pair, if the distance in the target feature subspace is larger than the threshold M, there would be no loss.

7.3 SETTING AND RESULTS

We have previously mentioned that we divided our dataset into $\mathbb{S}_{tr}$ and $\mathbb{S}_{te}$. We then used our trained GAN model to generate the synthetic data $\mathbb{S}_{syn}$. To train and test the discriminative model, we should create sample pairs (X_{p_1}, X_{p_2}). In the **training setting**, to create ***genuine pairs***, we randomly pick X_{p_1} and X_{p_2} from $\mathbb{S}_{tr}$ and $\mathbb{S}_{tr}$, respectively. To create ***impostor pairs***, we randomly pick X_{p_1} and X_{p_2} from $\mathbb{S}_{tr}$ and $\mathbb{S}_{te}$, respectively. In the **testing setting**, to create ***genuine pairs***, we randomly pick X_{p_1} and X_{p_2} from $\mathbb{S}_{syn}$ and $\mathbb{S}_{tr}$, respectively. To create ***impostor pairs***, we randomly pick X_{p_1} and X_{p_2} from $\mathbb{S}_{syn}$ and $\mathbb{S}_{te}$, respectively.

We evaluate experimental results using the Receiver Operating Characteristic (ROC). The ROC curve consists of the True Positive Rate (TPR) and False Acceptance Rate (FAR). The results are shown in Fig. 4. As can be seen, the discriminator model shows excellent performance, matching the synthetic data with its real counterpart.

8 PRIVACY ASSESSMENT

The Membership Inference Attack, Shokri et al. (2017), is one of the known methods that aims to investigate the disclosure of the individual's identity in a database. We focus on the privacy concerns regarding the individuals whose private data have been used for training purposes. For our experiments we randomly select some patient records ($\mathbb{Z}$), ($\mathbb{X}$), and ($\mathbb{R}$) from the real training set $\mathbb{S}_{tr}$, $\mathbb{S}_{te}$[1], and the synthetic data set $\mathbb{S}_{syn}$, respectively.

We compared each of the samples in the set of $\mathbb{X} + \mathbb{Z}$ with each sample in the set of $\mathbb{R}$ by using cosine similarity. If the score is higher then a threshold, then it flags the match, otherwise, we call it a mismatch. The threshold is determined by taking random samples from a normal distribution with mean 0.5 and standard deviation of 1. We only report the results associated with the most effective attack (best results for the attacker). For evaluation, we use *precision* and *recall* metrics. We investigated the effect of the number of records known by the attacker.

To assess ***the effect of the number of records known by the attacker***, we assume $|\mathbb{R}| = |\mathbb{X}| = |\mathbb{Z}|$. This is a practical assumption since accessing similar data and an abundant amount of synthetic

[1]The remaining portion of the data that has not been used for training.

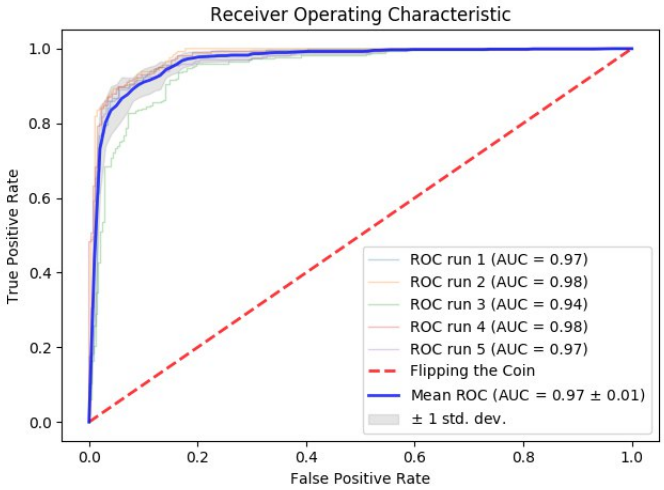

Figure 4: The results using the Discriminative Model. The results demonstrate the mean and standard deviation for 5 runs of experiments.

data (from the source) is not usually a limitation for the adversary. The *precision percentage* indicates that for the predictions that the adversary claims the patients' records have been used for training, only a portion of them (precision value) were actually used. On the other hand, the recall means the adversary has successfully determined that a portion of her known records were used in training. As can be seen in Table 3, by increasing the number of the real patient records known to the adversary, the attack will be even less accurate. Table 3 demonstrates the fact that higher precision is possible at lower recall rates when the number of known records is not high. However, a higher amount of revealed data increases the risk significantly.

Table 3: The precision and recall demonstrated as a function of the number of patients whose data is revealed to the attacker. $\mathcal{U}$ = # of Known Records to the attacker.

$\mathcal{U}$	100	1k	2k	3k	4k	5k
Precision	0.60	0.51	0.41	0.40	0.40	0.39
Recall	0.05	0.10	0.19	0.28	0.27	0.28

9 CONCLUSION

In this work, we proposed corGAN, which utilizes the one-dimensional convolutional neural networks to capture the correlated information between the patients' records. Through meticulous evaluation using real and synthetic datasets, corGAN demonstrated better results compared to the baseline methods in terms of the generative aspects. We showed the advantage of CNNs over MLPs to capture the correlated features. We also proposed a novel data-driven approach to measure the similarity between real and synthetic data by training a dedicated discriminative model. The empirical results prove the effectiveness of the discriminative model to connect the synthetic data to its real counterpart.

ACKNOWLEDGMENTS

Foremost, we would like to express our sincere gratitude to our advisor Prof. Edward A. Fox for the endless support of this research, for his patience, encouragement, enthusiasm, and immense knowledge. I gratefully acknowledge the fellowship award received towards my Ph.D. research from NewWave Telecom & Technologies.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

Mrinal Kanti Baowaly, Chia-Ching Lin, Chao-Lin Liu, and Kuan-Ta Chen. Synthesizing electronic health records using improved generative adversarial networks. *Journal of the American Medical Informatics Association*, 26(3):228–241, 2018.

Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Anna L Buczak, Steven Babin, and Linda Moniz. Data-driven approach for creating synthetic electronic medical records. *BMC medical informatics and decision making*, 10(1):59, 2010.

Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. *arXiv preprint arXiv:1703.06490*, 2017.

Hao Dong, Paarth Neekhara, Chao Wu, and Yike Guo. Unsupervised image-to-image translation with generative adversarial networks. *arXiv preprint arXiv:1701.02676*, 2017.

Khaled El Emam, David Buckeridge, Robyn Tamblyn, Angelica Neisa, Elizabeth Jonker, and Aman Verma. The re-identification risk of Canadians from longitudinal demographics. *BMC medical informatics and decision making*, 11(1):46, 2011.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pp. 513–520, 2007.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

R Devon Hjelm, Athul Paul Jacob, Tong Che, Adam Trischler, Kyunghyun Cho, and Yoshua Bengio. Boundary-seeking generative adversarial networks. *arXiv preprint arXiv:1702.08431*, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2018.

Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1857–1865. JMLR. org, 2017a.

Yoon Kim, Kelly Zhang, Alexander M Rush, Yann LeCun, et al. Adversarially regularized autoencoders. *arXiv preprint arXiv:1706.04223*, 2017b.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Yann Lecun, Fu Jie, and Jhuangfu. Loss functions for discriminative training of energy-based models. In *Proc. of the 10-th International Workshop on Artificial Intelligence and Statistics*, 2005.

Scott McLachlan, Kudakwashe Dube, and Thomas Gallagher. Using the caremap with health incidents statistics for generating the realistic synthetic electronic healthcare record. In *2016 IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 439–448. IEEE, 2016.

Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pp. 1520–1528, 2015.

Yubin Park, Joydeep Ghosh, and Mallikarjun Shankar. Perturbed Gibbs samplers for generating large-scale privacy-safe synthetic health data. In *2013 IEEE International Conference on Healthcare Informatics*, pp. 493–498. IEEE, 2013.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in neural information processing systems*, pp. 2234–2242, 2016.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18. IEEE, 2017.

Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1526–1535, 2018.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pp. 613–621, 2016.

Jason Walonoski, Mark Kramer, Joseph Nichols, Andre Quina, Chris Moesel, Dylan Hall, Carlton Duffett, Kudakwashe Dube, Thomas Gallagher, and Scott 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):230–238, 2017.

Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. Irgan: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 515–524. ACM, 2017.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.