CorGAN: Correlation-Capturing Convolutional Generative Adversarial Networks for Generating Synthetic Healthcare Records

Amirsina Torfi, Edward A. Fox
Virginia Polytechnic Institute and State University (Virginia Tech)
Department of Computer Science
Blacksburg, Virginia 24061
{atorfi,fox}@vt.edu

Abstract
Deep learning models have demonstrated high-quality performance in 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 are unique to researchers in this domain. This matter focuses attention on generating realistic synthetic data while ensuring privacy. In this paper, we propose a novel framework called correlation-capturing Generative Adversarial Network (CorGAN), to generate synthetic healthcare records. In CorGAN we utilize Convolutional Neural Networks to capture the correlations between adjacent medical features in the data representation space by combining Convolutional Generative Adversarial Networks and Convolutional Autoencoders. To demonstrate the model fidelity, we show that CorGAN generates synthetic data with performance similar to that of real data in various Machine Learning settings such as classification and prediction. We also give a privacy assessment and report on statistical analysis regarding realistic characteristics of the synthetic data. The software of this work is open-source and is available at: https://github.com/astorfi/cor-gan.

Introduction
Adoption of Electronic Health Records (EHRs) by the healthcare community, along with the massive quantity of available data, has led to calls for employing promising data-driven methods inspired by Artificial Intelligence (AI). Data-powered tools alter how clinicians and healthcare bureaus approach and satisfy patients’ needs for care. However, extending 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 (El Emam et al. 2011). Accordingly, more recent attention has focused on Synthetic Data Generation (SDG) which can satisfy reliably the needs for privacy.

We aim to create realistic synthetic EHR data by Generative Adversarial Networks (GANs), which have been successfully employed in applications such as image generation (Reed et al. 2016; Brock, Donahue, and Simonyan 2018; Karras, Laine, and Aila 2018), video generation (Vondrick, Pirsiavash, and Torralba 2016; Tulyakov et al. 2018), and image translation (Isola et al. 2017; Kim et al. 2017a; Dong et al. 2017). Contributions of this work include:

- We propose an efficient architecture to generate synthetic healthcare records using Convolutional GANs and Convolutional Autoencoders (CAs) which we call “CorGAN”.
- We demonstrate that CorGAN can effectively generate both discrete and continuous synthetic records.
- We demonstrate the effectiveness of utilizing Convolutional Neural Networks (CNNs) as opposed to Multilayer Perceptrons to capture inter-correlation between features.
- We show that CorGAN can generate realistic synthetic data that performs similarly to real data on classification tasks, according to our analysis and assessments.
- We report on a privacy assessment of the model and demonstrate that CorGAN provides an acceptable level of privacy, by varying the amount of synthetically generated data and amount of data known to an adversary.

Related Works
Some distinguished efforts were conducted in a variety of domains about synthetic data generation (Walonoski et al. 2017; Buczak, Babin, and Moniz 2010; McLachlan, Dube, and Gallagher 2016; Park, Ghosh, and Shankar 2013). But some of these works are overly disease-specific, unrealistic, or have failed to provide any substantial measurements regarding privacy.

Highly relevant is “medGAN” (Choi et al. 2017), using GANs for synthetic discrete EHR data. But in contrast to medGAN, we consider the temporal nature of the data and local correlation between features. Instead of regular multilayer perceptrons, we base our architecture on CNNs and provide empirical results to demonstrate the superior performance in capturing inter-correlations between data features.
Method

Discrete EHR Data Description

Many discrete variables (e.g., diagnosis, procedure code) are available in the dataset. Let’s assume there are |D| discrete variables and the vector \( V_C \in \mathbb{N}_0^{|D|} \) (where \( \mathbb{N}_0 \) indicates natural numbers including zero) is in a vector space. 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’s visit (encounter event) by a binary vector \( V_B \in \{0, 1\}^{|D|} \), where the \( j^{th} \) dimension shows whether the \( j^{th} \) variable occurred in the patient record. We represent the input space as a matrix in which columns indicate discrete variables in the EHR record. Such representation extracts multiple patients’ records representing different points in time.

Generative Adversarial Networks

A Generative Adversarial Network (GAN), introduced in (Goodfellow et al. 2014), 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. The goal of the generator is to fool the discriminator so it believes the artificial (i.e., generated) samples synthesized by the generator are real.

The generator goal is to learn the distribution \( p_g \) over data \( x \). 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 \( \theta_g \). The discriminator, \( D(x; \theta_d) \), decides if its input data is real or fake. \( D \) is trained to distinguish the training samples from \( G \) by minimizing \( \log(1-D(G(z))) \). \( D \) and \( G \) perform the following min-max game with value function \( V(G, D) \):

\[
\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})}[1 - \log D(G(\mathbf{z}))]
\]

Proposed Architecture

We use the architecture in Fig. 1. The discrete input \( \mathbf{X} \) represents the source EHR data; \( \mathbf{z} \) is the random distribution for the generator \( G \); \( G \) is the employed neural network architecture; \( Dec(G(\mathbf{z})) \) refers to the decoding function which is used to transform the generator \( G \) continuous output to their equivalent discrete values. The discriminator \( D \) attempts to distinguish real input \( \mathbf{X} \) from the discrete synthetic output \( Dec(G(\mathbf{z})) \). For the generator and the discriminator, a 1-Dimensional Convolutional GAN architecture is utilized.

Consider the decoding function \( Dec(\cdot) \). GANs are known for generating continuous values and encountering trouble when dealing with discrete variables. Recently, researchers 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). Some approaches use the indirect method such that they create a separate model to transform continuous to discrete data (Choi et al. 2017). Regarding EHR data generation, we are dealing with discrete data. Hence, our generative model should create discrete data directly, or there should be a function to transform the continuous data samples into discrete equivalents. We chose the second approach, and employed autoencoders.

Considering Fig. 1, the autoencoder digests (right part of the figure) discrete values and reconstructs the same discrete values as well. 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 reverses the process. The Binary Cross-Entropy (BCE) loss function is used for training the autoencoder:

\[
BCE_{loss} = -\frac{1}{N} \sum_{i=1}^{N} x_i \log(y_i) + (1 - x_i) \log(1 - y_i)
\]

\[
y_i = Dec(Enc(x_i))
\]

We used denoising autoencoders (Vincent et al. 2010) to create a more robust pretrained model as we do not expect our model to always generate perfect discrete samples. After training the autoencoder, we need to use its decoder to convert continuous values to their associated discrete values. 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.

\[
\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(\mathbf{z})}[1 - \log D(Dec(G(\mathbf{z})))]
\]

As we are dealing with 1D data, we chose the 1-Dimensional Convolutional Autoencoders (1D-CAEs) as a particular form of the regular CAEs. This approach enables us to capture the neighboring feature correlations. We call our proposed architecture CorGAN. It is worth noting that for our experiments with discrete variables, we round the values of \( Dec(G(\mathbf{z})) \) to their nearest integers (the outcome is zero or one) to guarantee that we train and evaluate the discriminator on discrete values.

Training Augmentation

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. This phenomenon is called Mode Collapse. Some approaches have been proposed to handle the mode collapse issue such as minibatch discrimination (Salimans et al. 2016) and unrolled GANs (Metz et al. 2016). We utilized minibatch discrimination due to its better stability. We also utilized batch normalization (Ioffe and Szegedy 2015) to improve the generator’s learning abilities. Furthermore, we used LeakyRelu activation unit (Maas, Hannun, and Ng 2013) as it consistently demonstrated equal or better results over other common activation functions (Xu et al. 2015).
Privacy
We utilize the Membership Inference (MI) attack as an approach to measure the privacy. Membership Inference (MI), proposed in (Shokri et al. 2017), refers to determining whether a given record generated by a known machine learning model was used as part of the training data. If the adversary has complete access to the records of a particular patient and can recognize their employment in the model training, that is an indication of information leakage, as it can jeopardize the whole dataset privacy or at least the particular patient’s private information. Here, we will assume the adversary has the synthetically generated data as well as a portion of the compromised real data.

Experiments
We evaluated CorGAN with two datasets. First, we explain the datasets and baseline models. Then, we provide the results regarding the evaluation of the synthetic data in terms of the realistic characteristics. Finally, we report on a privacy assessment of the model.

Datasets
We used two publicly available datasets in this study. The first is the MIMIC-III dataset (Johnson et al. 2016) consisting of the medical records of almost 46K patients. From MIMIC-III, we extracted ICD-9 codes only. We represent a patient record as a fixed-size vector with 1071 entries for each patient record. This dataset is used for experiments with binary discrete variables.

We conducted our experiments regarding continuous variables with the UCI Epileptic Seizure Recognition dataset (Andrzejak et al. 2001). This dataset characterizes brain activities. The core task is classification, regarding if a sample indicates a seizure activity. The number of features and samples are 179 and 11500, respectively. Almost 20% of the samples are categorized as seizure activity. So, we are dealing with an unbalanced dataset in a binary classification setting. The first 178 features are the values of the Electroencephalogram (EEG) recordings at different time points, and the last feature is the class. There are five values for the class label \(y = 1, ..., 5\). Except for \(y = 1\), the rest of the classes indicate subjects who do not have an epileptic seizure. Dataset statistics are given in Table 1.

Models
To show the effectiveness of our proposed architecture, we compare our results with different baseline methods as below:

- **Stacked Deep Boltzmann Machines (DBMs):** We trained a stacked Deep Boltzmann Machine (DBM) (Hinton and Salakhutdinov 2009). After which, we used Gibbs sampling to generate synthetic binary samples. All hidden layers have 256 dimensions. We employed greedy contrastive divergence to create the model. We ran Gibbs sampling for 500 iterations per sample.

- **Variational Autoencoder (VAE):** We used VAEs (Kingma and Welling 2013) as one of our baseline models. For both the encoder and the decoder, we used 1D convolutional neural networks, each having two hidden layers. All hidden layers have the size of 128. We trained VAE with Adam optimizer for 500 epochs and for the batch size of 500.

- **medGan:** The medGan (Choi et al. 2017) architecture consists of the following elements: (1) regular multilayer perceptrons for autoencoder, discriminator, and generator. (2) shortcut connections to improve the power of generator. (3) minibatch-averaging (Choi et al. 2017) to cope with the mode collapse.

Evaluation
In this section, we report our evaluation results regarding the quality of the synthetic data and the privacy risks. Here, we divide the dataset into a training set \(S_{tr} \in \{0, 1\}^{R \times |M|}\) and a test set \(S_{te} \in \{0, 1\}^{T \times |M|}\), where \(|M|\) is the feature size and is consistent for all sets. We use \(S_{tr}\) to train the models, then generate synthetic samples \(S_{syn} \in \{0, 1\}^{S \times |M|}\) using
the trained model. Noted that we usually use the same number of samples for $S_{syn}$ and $S_{tr}$.

**Evaluation of the Synthetic Data Quality**

We use the following two metrics to evaluate our synthetically generated data.

- **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).

- **Dimension-wise prediction**: This approach measures how robust the model catches the inter-dimensional connections of the real data samples. Assume $S_{tr}$ is used to generate $S_{syn}$. Then, one random fixed dimension ($k$) from each $S_{syn}$ and $S_{tr}$ are selected as $S_{syn,k} \in \{0,1\}^{N \times 1}$ and $S_{tr,k} \in \{0,1\}^{N \times 1}$. We call it the testing dimension. The rest of the dimensions ($S_{syn,\setminus k} \in \{0,1\}^{N \times 1}$ and $S_{tr,\setminus k} \in \{0,1\}^{N \times 1}$) are used to train a classifier, which aims to predict the value of the testing dimension of the test set $S_{te,k} \in \{0,1\}^{N \times 1}$.

- **Binary Classification**: We use this metric for our experiments with continuous data. To empirically verify the quality of the synthetic data, we consider two different settings. (A) Train and test the predictive models on the real data. (B) train the predictive model on synthetic data and test it on the real data. If the model evaluated in setting (B), represents competitive results with the same model performed in setting (A), we can conclude the synthetic data has good predictive modeling similar to the real data.

For Dimension-wise probability and Dimension-wise prediction experiments, we used MIMIC-III dataset and for Binary Classification experiments we used UCI Epileptic Seizure Recognition dataset. The results regarding the investigation of dimension-wise probability are depicted in Fig. 2.

In this section, the experiments are conducted on the MIMIC-III dataset regarding the membership inference attack. For privacy assessment, we randomly take $P$ samples from each $S_{te}$ and $S_{tr}$, and call them $S_{te}^{P}$ and $S_{tr}^{P}$. We assume the attacker has the complete knowledge of both $S_{te}^{P}$ and $S_{tr}^{P}$. Clearly, $S_{te}^{P}$ was used to train the generating model, but $S_{tr}^{P}$ wasn’t. So we have $R = 2 \times P$ records. Then, we compared each of these records with the synthetically generated data samples $S_{syn} \in \{0,1\}^{S \times |M|}$.

For the MIMIC dataset experiments, although the temporal information is ignored, the medical health diagnosis are sorted in terms of similarity in the feature vector of the MIMIC data. Therefore, 1D CNNs capture the correlation between features rather than the temporal information.

<table>
<thead>
<tr>
<th>Generative Model</th>
<th>AUROC</th>
<th>AUPRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBM (Hinton and Salakhutdinov 2009)</td>
<td>0.81 ± 0.017</td>
<td>0.27 ± 0.013</td>
</tr>
<tr>
<td>VAE (Kingma and Welling 2013)</td>
<td>0.84 ± 0.021</td>
<td>0.31 ± 0.022</td>
</tr>
<tr>
<td>medGAN (Choi et al. 2017)</td>
<td>0.89 ± 0.023</td>
<td>0.35 ± 0.014</td>
</tr>
<tr>
<td>CorGAN [ours]</td>
<td>0.92 ± 0.012</td>
<td>0.41 ± 0.015</td>
</tr>
</tbody>
</table>

For binary classification experiments, we used the same predictive models as for the dimension-wise predictions. We reported the averaged AUROC (averaged area under the ROC for all models) and AUPRC (averaged area under the PR curve for all models) for the models’ evaluations. The difference with our experiments here is that we are not dealing with binary variables. Hence, for the medGAN and CorGAN methods we eliminate the autoencoder as depicted in Fig. 1. As can be observed in Table 3, our proposed method outperforms the other methods. As the UCI Epileptic Seizure Recognition dataset features contain temporal information, our method is able to capture temporal data information more effectively due to the usage of 1D CNNs.

**Privacy Assessment**

In this section, the experiments are conducted on the MIMIC-III dataset regarding the membership inference attack. For privacy assessment, we randomly take $P$ samples from each $S_{te}$ and $S_{tr}$, and call them $S_{te}^{P}$ and $S_{tr}^{P}$. We assume the attacker has the complete knowledge of both $S_{te}^{P}$ and $S_{tr}^{P}$. Clearly, $S_{te}^{P}$ was used to train the generating model, but $S_{tr}^{P}$ wasn’t. So we have $R = 2 \times P$ records. Then, we compared each of these records with the synthetically generated data samples $S_{syn} \in \{0,1\}^{S \times |M|}$.

We compared each of the samples in the set of $S_{te}^{P} + S_{tr}^{P}$ with each samples in the set of $S_{syn}$ and we calculate cosine similarity score. Cosine similarity is used since it provides a more meaningful correlation metric as opposed to distance metrics (Mateo-Sanz, Sebé, and Domingo-Ferrer 2004) used in previous research efforts (Choi et al. 2017). If the score is higher than a threshold, then it flags the match, otherwise,
we call it a mismatch. For threshold, we randomly select 100 threshold values from a Gaussian distribution with a mean of 0.5 and a standard deviation of 0.01 (ignoring possible negative values), and we report the results which demonstrate the best adversary attack.

For evaluation, we use precision and recall metrics. We conduct two sets of experiments here: (1) investigating the effect of the number of records known by the attacker (Table. 4) and (2) examining the effect of synthetic data volume on the privacy risk (Fig. 3).

As can be seen in Table. 4, by increasing the number of the real patient records known to the adversary, the attack will be even less accurate. It also demonstrates the fact that higher precision is possible at lower recall rates when the number of known records is not high. However, as is evident, a higher amount of revealed data increases the privacy risk significantly.

Regarding the effect of number of generated synthetic data on the privacy risk, as can be observed in Fig. 3, the increasing number of synthetic records does not have a significant effect on the recall, but it causes a dramatic decrease in precision. Henceforth, by having a fixed amount of known records, a higher number of synthetic patient’s records can be very misleading for the adversary. This empirical observation indicates that the increasing number of synthetic records, with a fixed number or revealed patient’s records to the attacker, does not necessarily raise privacy risk.

Table 4: 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.

<table>
<thead>
<tr>
<th>$\mathcal{U}$</th>
<th>100</th>
<th>1k</th>
<th>2k</th>
<th>3k</th>
<th>4k</th>
<th>5k</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>0.60</td>
<td>0.51</td>
<td>0.41</td>
<td>0.40</td>
<td>0.40</td>
<td>0.39</td>
</tr>
<tr>
<td>Recall</td>
<td>0.05</td>
<td>0.10</td>
<td>0.19</td>
<td>0.28</td>
<td>0.27</td>
<td>0.28</td>
</tr>
</tbody>
</table>

Concluding Remarks

In this work, we proposed CorGAN, which utilizes the convolutional generative adversarial networks to learn the distribution of real patient records. Through precise evaluation using real and synthetic datasets, CorGAN demonstrated decent results for both discrete and continuous records. We empirically proved the superiority of CNNs over MLPs to capture the correlated features. We believe our method can be effectively extended and employed to longitudinal records as well for which the goal is to capture the temporal characteristics of the data.

References


