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-ganhttps://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 $|\boldsymbol{D}|$ discrete variables and the vector $\boldsymbol{V_C} \in \boldsymbol{N_0}^{|\boldsymbol{D}|}$ (where 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 $\boldsymbol{V_B} \in \{\boldsymbol{0}, \boldsymbol{1}\}^{|\boldsymbol{D}|}$, 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 θ_g. The discriminator, $D(\mathbf{x}; \theta_d)$, 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):

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

Proposed Architecture

We use the architecture in Fig. 1. 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; $\boldsymbol{Dec(G(z))}$ 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 1-Dimensional Convolutional GAN architecture is utilized.

Consider the decoding function $\boldsymbol{Dec(.)}$. 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:

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

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

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.

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

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 $Dec(G(z))$ 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).

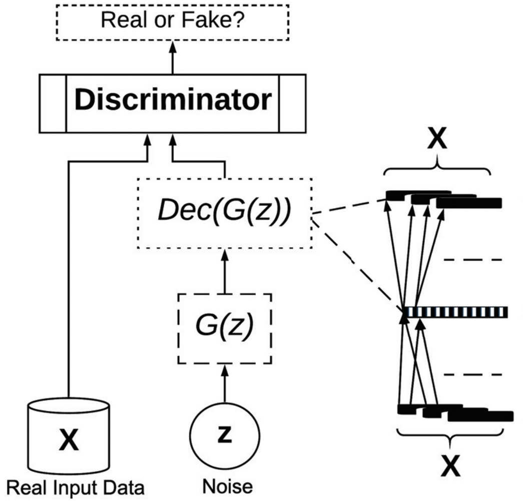

Figure 1: The architecture for generating synthetic data from real samples. The right side of the figures, shows the pre-trained convolutional autoencoder which its decoder part is being used to transform the generated continuous samples to their discrete equivalents.

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.

Table 1: Statistics of the UCI Epileptic Seizure Recognition dataset.

Dataset	**UCI**
# of patients	500
Each patient's data points	4097
Each patient's duration of recording	23.5 seconds
# data points chunks per patient	23
# of data points per chunk	178
Duration per chunk	1 second
Data type	Continuous EEG

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 $\mathcal{S}_{tr} \in \{0,1\}^{R\times|\mathcal{M}|}$ and a test set $\mathcal{S}_{te} \in \{0,1\}^{T\times|\mathcal{M}|}$, where $|\mathcal{M}|$ is the feature size and is consistent for all sets. We use $\mathcal{S}_{tr}$ to train the models, then generate synthetic samples $\mathcal{S}_{syn} \in \{0,1\}^{S\times|\mathcal{M}|}$ using the trained model. Noted that we usually use the same number of samples for $\mathcal{S}_{syn}$ and $\mathcal{S}_{tr}$.

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 $\mathcal{S}_{tr}$ is used to generate $\mathcal{S}_{syn}$. Then, one random fixed dimension (k) from each $\mathcal{S}_{syn}$ and $\mathcal{S}_{tr}$ are selected as $\mathcal{S}_{syn,k} \in \{0,1\}^{N\times 1}$ and $\mathcal{S}_{tr,k} \in \{0,1\}^{N\times 1}$. We call it the testing dimension. The rest of the dimensions ($\mathcal{S}_{syn,\setminus k} \in \{0,1\}^{N\times 1}$ and $\mathcal{S}_{tr,\setminus k} \in \{0,1\}^{N\times 1}$) are used to train a classifier, which aims to predict the value of the testing dimension of the test set $\mathcal{S}_{te,k} \in \{0,1\}^{N\times 1}$.
- **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. As can be seen, the *CorGAN* is superior compared to other methods. An interesting observation is that the VAE is never generating any synthetic data for which the probability of occurrence of a diagnosis code is higher than its counterpart in the real data.

For dimension-wise prediction (Table 2), we use the following classifiers the predictive model types: Logistic Regression, Random Forests (Breiman 2001), Linear SVM (Cortes and Vapnik 1995), and Decision Tree (Quinlan 1986). For our experiments, we conduct E=100 number of runs. In each run, we pick a random testing dimension from test set ($\mathcal{S}_{syn,k} \in \{0,1\}^{N\times 1}$) and will train each predictive model on $\mathcal{S}_{syn,\setminus k}$ and $\mathcal{S}_{tr,\setminus k}$. This results in having two models as $Model_{syn}^{type}$ and $Model_{real}^{type}$. The superscript refers to the kind of model that was used to be trained on both real and synthetic sets. We then report the performance over all predictive models and for all experiments using the F1-score variation. F1-score variation means the difference between the F-1 score obtained from training the model on synthetic and real datasets.

Table 2: Comparison of different baseline architectures. The reported metric demonstrate the mean and standard deviation of the F-1 score differences. A better model has a closer score to zero.

Generative Model	F1-Score
DBM (Hinton and Salakhutdinov 2009)	0.12 ± 0.052
VAE (Kingma and Welling 2013)	0.069 ± 0.043
medGAN (Choi et al. 2017)	0.043 ± 0.049
CorGAN [ours]	**0.021 ± 0.045**

Table 3: Comparison of different generative models for binary classification. The averaged AUROC and AUROC for utilizing the predictive models on the real data are **0.95** and **0.46**.

Generative Model	AUROC	AUPRC
DBM	0.81 ± 0.017	0.27 ± 0.013
VAE	0.84 ± 0.021	0.31 ± 0.022
medGAN	0.89 ± 0.023	0.35 ± 0.014
CorGAN [ours]	**0.92 ± 0.012**	**0.41 ± 0.015**

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.

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 termporal 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 $\mathcal{P}$ samples from each $\mathcal{S}_{tr}$ and $\mathcal{S}_{te}$ and call them $\mathcal{S}_{tr}^{\mathcal{P}}$ and $\mathcal{S}_{tr}^{\mathcal{P}}$. We assume the attacker has the complete knowledge of both $\mathcal{S}_{tr}^{\mathcal{P}}$ and $\mathcal{S}_{te}^{\mathcal{P}}$. Clearly, $\mathcal{S}_{tr}^{\mathcal{P}}$ was used to train the generating model, but $\mathcal{S}_{te}^{\mathcal{P}}$ wasn't. So we have $\mathcal{R} = 2 \times \mathcal{P}$ records. Then, we compared each of these records with the synthetically generated data samples $\mathcal{S}_{syn} \in \{0,1\}^{S\times|\mathcal{M}|}$.

We compared each of the samples in the set of $\mathcal{S}_{te}^{\mathcal{P}} + \mathcal{S}_{tr}^{\mathcal{P}}$ with each samples in the set of $\mathcal{S}_{syn}$ 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,

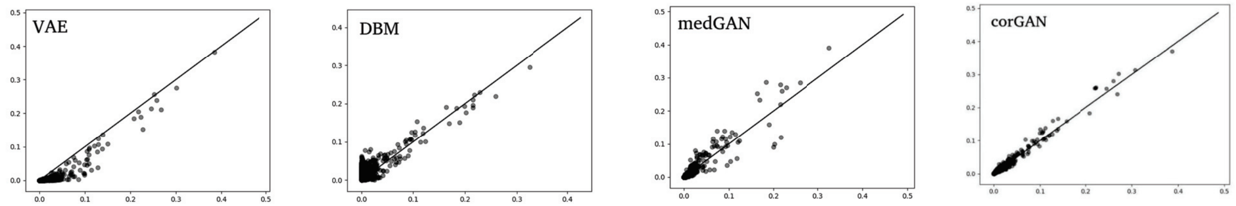

Figure 2: he scatter plots of dimension-wise probability. Each point depicts one of 1071 unique diagnosis codes. The x-axis and y-axis represent the Bernoulli success probability for real and synthetic datasets, respectively. The diagonal line shows the ideal case.

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.

$\mathcal{U}$	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

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 the number of synthetic records, with a fixed number or revealed patient's records to the attacker, does not necessarily raise privacy risk.

Conclusion

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.

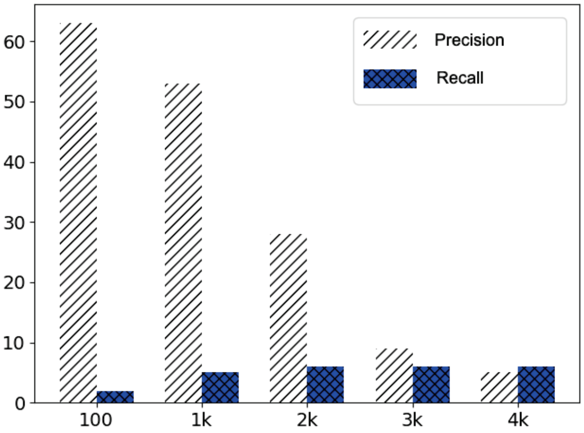

Figure 3: The recall/precision as a function of the number of generated synthetic records. The number of records known to the adversary is considered fixed and is equal to 100.

References

Andrzejak, R. G.; Lehnertz, K.; Mormann, F.; Rieke, C.; David, P.; and Elger, C. E. 2001. 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):061907.

Breiman, L. 2001. Random forests. *Machine learning* 45(1):5–32.

Brock, A.; Donahue, J.; and Simonyan, K. 2018. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

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

Choi, E.; Biswal, S.; Malin, B.; Duke, J.; Stewart, W. F.; and Sun, J. 2017. Generating multi-label discrete patient records using generative adversarial networks. *arXiv preprint arXiv:1703.06490*.

Cortes, C., and Vapnik, V. 1995. Support-vector networks. *Machine learning* 20(3):273–297.

Dong, H.; Neekhara, P.; Wu, C.; and Guo, Y. 2017. Unsupervised image-to-image translation with generative adversarial networks. *arXiv preprint arXiv:1701.02676*.

El Emam, K.; Buckeridge, D.; Tamblyn, R.; Neisa, A.; Jonker, E.; and Verma, A. 2011. The re-identification risk of Canadians from longitudinal demographics. *BMC medical informatics and decision making* 11(1):46.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Hinton, G. E., and Salakhutdinov, R. R. 2009. Replicated softmax: an undirected topic model. In *Advances in neural information processing systems*, 1607–1614.

Hjelm, R. D.; Jacob, A. P.; Che, T.; Trischler, A.; Cho, K.; and Bengio, Y. 2017. Boundary-seeking generative adversarial networks. *arXiv preprint arXiv:1702.08431*.

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

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1125–1134.

Johnson, A. E.; Pollard, T. J.; Shen, L.; Li-wei, H. L.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Celi, L. A.; and Mark, R. G. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3:160035.

Karras, T.; Laine, S.; and Aila, T. 2018. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*.

Kim, T.; Cha, M.; Kim, H.; Lee, J. K.; and Kim, J. 2017a. Learning to discover cross-domain relations with generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1857–1865. JMLR. org.

Kim, Y.; Zhang, K.; Rush, A. M.; LeCun, Y.; et al. 2017b. Adversarially regularized autoencoders. *arXiv preprint arXiv:1706.04223*.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*.

Maas, A. L.; Hannun, A. Y.; and Ng, A. Y. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, 3.

Mateo-Sanz, J. M.; Sebé, F.; and Domingo-Ferrer, J. 2004. Outlier protection in continuous microdata masking. In *International Workshop on Privacy in Statistical Databases*, 201–215. Springer.

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

Metz, L.; Poole, B.; Pfau, D.; and Sohl-Dickstein, J. 2016. Unrolled generative adversarial networks.

Park, Y.; Ghosh, J.; and Shankar, M. 2013. Perturbed Gibbs samplers for generating large-scale privacy-safe synthetic health data. In *2013 IEEE International Conference on Healthcare Informatics*, 493–498. IEEE.

Quinlan, J. R. 1986. Induction of decision trees. *Machine learning* 1(1):81–106.

Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; and Lee, H. 2016. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training GANs. In *Advances in neural information processing systems*, 2234–2242.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, 3–18. IEEE.

Tulyakov, S.; Liu, M.-Y.; Yang, X.; and Kautz, J. 2018. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1526–1535.

Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; and Manzagol, P.-A. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research* 11(Dec):3371–3408.

Vondrick, C.; Pirsiavash, H.; and Torralba, A. 2016. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, 613–621.

Walonoski, J.; Kramer, M.; Nichols, J.; Quina, A.; Moesel, C.; Hall, D.; Duffett, C.; Dube, K.; Gallagher, T.; and McLachlan, S. 2017. 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.

Wang, J.; Yu, L.; Zhang, W.; Gong, Y.; Xu, Y.; Wang, B.; Zhang, P.; and Zhang, D. 2017. 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*, 515–524. ACM.

Xu, B.; Wang, N.; Chen, T.; and Li, M. 2015. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*.