SMOOTH-GAN: Towards Sharp and Smooth Synthetic EHR Data Generation

Sina Rashidian[1(✉)], Fusheng Wang[1,2], Richard Moffitt[2], Victor Garcia[2],
Anurag Dutt[1], Wei Chang[1], Vishwam Pandya[1], Janos Hajagos[2], Mary Saltz[2],
and Joel Saltz[2]

[1] Stony Brook University, Stony Brook, NY 11794, USA
{sina.rashidian,fusheng.wang,anurag.dutt,wei.chang,
vishwam.pandya}@stonybrook.edu

[2] Stony Brook Medicine, Stony Brook, NY 11794, USA
{fusheng.wang,richard.moffitt,victor.garcia,janos.hajagos,
mary.saltz,joel.saltz}@stonybrookmedicine.edu

Abstract. Generative adversarial networks (GANs) have been highly successful for generating realistic synthetic data. In healthcare, synthetic data generation can be helpful for producing annotated data and improving data-driven research without worries on data privacy. However, electronic health records (EHRs) are noisy, incomplete and complex, and existing work on EHR data is mainly devoted to generating discrete elements such as diagnosis codes and medications or frequent laboratory values. In this work, we propose SMOOTH-GAN, a novel approach for generating reliable EHR data such as laboratory values and medications given diagnosis codes. SMOOTH-GAN takes advantage of a conditional GAN architecture with WGAN-GP loss, and is able to learn transitions between disease stages with high flexibility over data customization. Our experiments demonstrate the model's effectiveness in terms of both statistical similarity and accuracy on machine learning based prediction. To further demonstrate the usage of our model, we apply counterfactual reasoning and generate data with occurrence of multiple diseases, which can provide unique datasets for artificial intelligence driven healthcare research.

Keywords: Generative adversarial networks · Electronic health records · Synthetic data generation · Counterfactual machine learning

1 Introduction

Electronic health records (EHRs) include rich information to support artificial intelligence (AI) driven healthcare. Analyzing EHR data has many practical applications such as predicting mortality [3], phenotyping diseases [6], detecting missing/missed diagnosis codes [17] and predicting unplanned readmissions [2].

M. Michalowski and R. Moskovitch (Eds.): AIME 2020, LNAI 12299, pp. 37–48, 2020.
https://doi.org/10.1007/978-3-030-59137-3_4

In the meantime, EHR data is difficult to access due to privacy protection. It is also noisy, incomplete and complex, thus difficult for researchers to work with. Generating synthetic EHR datasets can help both AI and medical communities to share datasets for developing new algorithms and comparing results.

Synthetic data generation can provide the opportunity for researchers to share large datasets without privacy concerns and improve the quality of studies with competitive and reproducible experiments. Having a reliable data generator can also be useful for augmentation tasks and building more robust machine learning models that can potentially provide new insights into how models can interpret and capture patterns from EHR data. However, for a various number of reasons including, but not limited to, large dimensions, longitudinal irregularity, missing values, and heterogeneity it is more challenging to provide synthetic data generation for EHR data, compared with other applications such as imaging.

Generative adversarial networks (GANs) are generative models for creating realistic synthetic data based on an adversarial process which are proven to be more effective than their statistical counterparts [10]. GANs have been very successful with image generation, and there are many interesting applications of GANs such as real images augmenting with Invertible Conditional GANs [16]. This success inspired studies to adapt strategies to tabular data [19].

In recent years, the concept of counterfactual reasoning has gained attention within the machine learning community as one of the potential methods for explainable AI and generating never-before-seen patterns [13]. This concept has a lot of potential in AI driven healthcare, where physicians encounter new patterns among diseases and are skeptical about black box models. Such patterns can be potentially uncovered through GAN based methods.

In this paper, we take advantage of GANs for high quality synthetic data generation and data augmentation, and explore how the models can track patients over the course of their disease using EHR data. Instead of a human-based perception of disease progression by a clinical expert, we are interested in understanding how the models can observe and capture these patterns. We believe these observations can help building more robust models and provide essential knowledge for understanding decisions made by neural networks. We will first introduce SMOOTH-GAN (Sharp sMOOTh eHr), a new approach for generating synthetic EHR data, and then, we will provide in-depth analysis of the models generated by defining new metrics and concepts. At the end, we explore an application of counterfactual data generation.

2 Related Work

Recently, generating synthetic EHR data using GANs has become an active research area. However, there is limited work due to several challenges associated with EHR data. One notable project is MedGAN which focuses on generating discrete data elements -medications and diagnosis codes- by adding an additional encoder decoder inside the GAN architecture [7]. Another inspiring work is RCGAN which provides a framework for generating frequent sequences using conditional recurrent GANs designed for medical time series data [8]. Moreover, the SSL-GAN augments medications and diagnosis codes for improving classification tasks with a semi-supervised learning approach [5].

In this study, we design a conditional GAN which generates both medications and laboratory values for given diseases. Our work has the following salient features. First, the generator generates both continuous and binary values and there is no need to have separate generators. Secondly, we created new methodology to have more control over conditions, which can help with generating patients with different stages of disease. Furthermore, conditions in SMOOTH-GAN can be combined together, creating more realistic and diverse encounters.

3 Data

We extracted inpatient encounter data for adults ($\geq$18) from the Cerner Health-Facts database, a large multi-institutional de-identified database derived from EHRs and administrative systems. From the 10 highest volume inpatient facilities, we randomly chose one acute-care facility (143) and extracted encounters with at least one diagnosis code, laboratory value, and medication from 1/1/2016 to 12/31/2017. We used 47,412 encounters that were broken into 80% for the training set and the rest for the test set.

As multiple values for each laboratory test exist for an encounter, we take the median of each test for each encounter. For medications, we consider them binary whether they were ordered or not. After filtering out features with less than 5% occurrence rate to reduce sparsity and noise, 166 features remained. Diagnosis codes for 5 major chronic conditions, hypertension, congestive heart failure (CHF), diabetes mellitus, cardiac arrhythmias, and chronic kidney disease (CKD) were defined according to [18] and used in this study.

4 Methods

We first briefly review the GAN concept and the architecture we are adapting, and then discuss the details of our algorithm and methods.

4.1 Generative Adversarial Networks Concept

A GAN is normally comprised of two neural networks, which compete with each other in a minimax game: a discriminator and generator. The generator's $G(z;\theta_g)$ goal is to generate samples intended to come from the same distribution as the training set, where z is random noise usually from the normal distribution. The discriminator $D(x,\theta_d)$ tries to detect whether the samples generated by the generator are real or fake. Ideally, the data distribution by G (p_g) should be the same as the real data distribution (p_{data}) [9,10]. Conditional GANs are extensions where generators generate data based on some extra information as conditions or labels [14]. The formal optimization formula is:

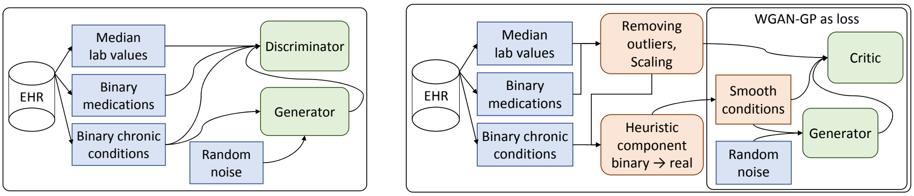
$$\min_G \max_D V(\theta_g,\theta_d) = E_{x\sim p_{data}}[log(D(x|y)] + E_{z\sim p_z}[log(1-D(G(z|y))]$$

where θ_g, θ_d are parameters for the generator and discriminator, and p_z is the normal distribution. The ideal generator would create authentic samples similar to the training set that force the discriminator to guess randomly. y is the vector condition, which is given to both the generator and the discriminator in *cgan* architecture. We adapted Wasserstein-GAN with gradient penalty (WGAN-GP) as the loss function in this work. It has several advantages including not suffering from the gradient diminishing problem during training and producing more robust results [1,12]. The discriminator becomes critic in this method which assigns a real value score instead of a binary value.

4.2 SMOOTH-GAN

The SMOOTH-GAN is a conditional GAN adapting WGAN-GP for healthcare data. Its main objective is to generate high quality EHRs, including laboratory values and medications, given diagnosis codes as conditions. We refer to diagnosis codes as set C, where $c \in \{0, 1\}^{|C|}$ is a random set of conditions, and the i^{th} dimension c_i shows presence or absence of i^{th} disease in a patient's encounter record. In EHR data, diagnosis codes are recorded as binary values indicating which diseases patients have. Although having a disease is a binary status, reaching the certain threshold to have the disease is in a probabilistic continuous space for most chronic diseases. For instance, patients with a "hemoglobin A1C" of 6.0 and 4.5 are both below the threshold of diabetes, but the first patient is closer to being a positive case and has a higher risk of getting diabetes. However, in EHR data both of these patients are labeled as 0.

A generative model needs to be reliable and adjustable to have practical usage. We observed by generating a GAN model directly with those binary values as conditions, the generated data in many cases was borderline and did not pass the cutoff for that disease. The GAN was learning broad patterns and the control of the output was limited. The outcome was not deterministic by input conditions and was highly dependent on random variables.



General cGAN

SMOOTH-GAN

Fig. 1. Illustrating how the heuristic function is added to the model.

Based on the issued discussed above, we added a unit which would change the GAN input conditions to smooth labels. Note that assigning an exact probability is a very difficult task, especially when the definition of what is the probability is debatable. Therefore, we are looking for a heuristic function that given binary conditions and input data, can estimate the condition in a continuous space, $H(c, x) = \tilde{c}$ where $\tilde{c} \in [0, 1]$. Although finding the perfect function for assigning probabilities/risk scores to encounters is an active field of research in healthcare, finding a heuristic function simplifies the task and provides a fast solution. The architecture is shown in Fig. 1.

There are different ideas and models to use as the heuristic function. We use random forest (RF) models as the core part of this heuristic function in this work. These models can be trained on the training set and assigns probabilities for each disease accordingly. When the estimated probability is in contrary with the original label, we adjust it to the center (0.5). It is necessary that the model can label the majority of each class correctly. We demonstrate how the model is capable of generating more diverse synthetic data with traceable disease progress by using $\tilde{c}$ instead of c in training the GAN in Sect. 5.3.

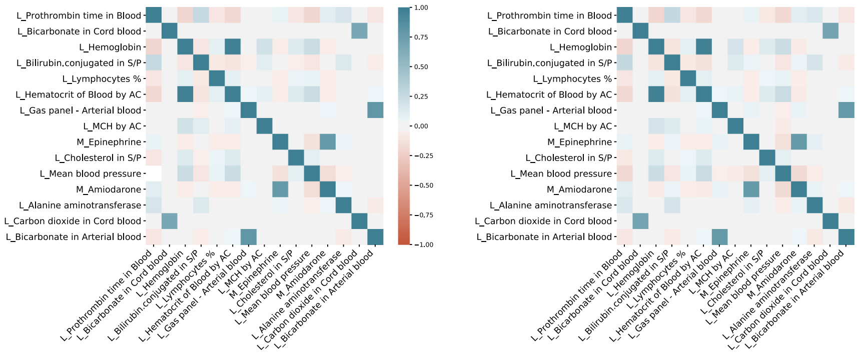
Training Details. The generator has two *leakyRelu* hidden layers with $\alpha = 0.2$ each one is followed by a batch normalization layer and *tanh* output layer. The critic has two *leakyRelu* hidden layers with $\alpha = 0.2$ and *linear* output layer. The critic is trained 5 times more than the generator in each epoch. Moreover, the heuristic function is pre-trained in advance (RF models). The model was trained for 600 epochs. Data is scaled to $[-1, 1]$ and outliers with Z-score more than 4 are removed for non-binary features before the median imputation.

5 Results

In this section, we provide in-depth analysis of our GAN method and innovative applications. We used random forest as the prediction model since we needed to know the important features and output probability of inputs for most experiments. To have a reasonable comparison, the synthetic dataset is generated given a set of conditions similar to the training set.

5.1 Statistical Analysis

The first step is to measure how the synthetic data distribution fits to the real training set. We measured the mean absolute error (MAE) for means and standard deviations of columns and element-wise Pearson correlations as shown in Table 1. For medications which are binary values, we calculated MAE for dimension-wise probability. The Loss functions based on Wasserstein distance have robust progress even when data is partially binary. Figure 2 shows heatmaps of Pearson correlations for real and synthetic data.



Training data

Synthetic data

Fig. 2. Heatmap for 15 features with highest correlation. MC: Manual Count, AC: Automated Count, S/P: Serum or Plasma, L: lab value, M: medication

Table 1. Mean absolute error for statistics between real and synthetic data

Method name	Laboratory mean	Laboratory std	Medications prob	Correlation
cGAN	248.10	22.92	0.382	NaN
AC-GAN	79.49	14.53	0.068	0.196
WGAN	1.33	5.39	0.007	0.058
WGAN-GP	0.80	1.77	0.003	0.039
SMOOTH-GAN	0.68	2.29	0.003	0.039

5.2 Synthetic Data Prediction Models

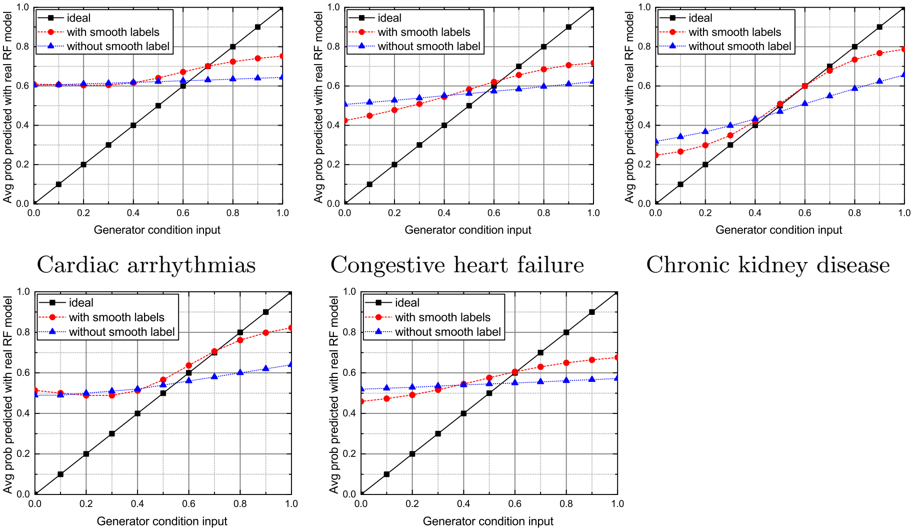
One major goal of generating synthetic data is to use it in place of real data when training machine learning models. We are comparing RF models trained on real training data and generated data with the same **real test set** which was untouched in Table 2. This experiment has become the main metric to measure a GAN's success in related publications [7,8,19]. Moreover, it is critical that the synthetic model is making predictions based on similar factors to the real training set. Otherwise, GANs might have altered other features correlated with the input conditions and generated new patterns which is undesired. The last rows of Table 2 show the number of overlapping features in the top 15 most important features of both the synthetic and real trained models. Note that AC-GAN is designed for mutually exclusive input conditions [15].

Table 2. Performance of trained RF models on synthetic and real data measured on **real test set**. "#/15" represents number of common features with top 15 most important features identified by real RF model.

Disease name	Metric	Real	cGAN	WGAN	AC-GAN	WGAN-GP	**SMOOTH-GAN**
Hypertension	AUROC	.8822	.5896	.8434	.5165	.8515	**.8625**
	AUPRC	.7965	.3929	.7474	.3324	.7562	**.7688**
	#/15	-	2	8	2	8	**9**
Diabetes	AUROC	.9357	.5849	.8412	.5759	.8641	**.8708**
	AUPRC	.8905	.3821	.7702	.3872	.8061	**.8089**
	#/15	-	2	9	2	**11**	**11**
Congestive heart failure	AUROC	.9000	.5663	.8239	.5795	.8619	**.8633**
	AUPRC	.7471	.2483	.5885	.2708	.6551	**.6577**
	#/15	-	3	8	2	9	**12**
Chronic kidney disease	AUROC	.9544	.6331	.9386	.4240	**.9404**	.9380
	AUPRC	.8705	.3654	.8380	.1740	**.8384**	.8321
	#/15	-	3	9	1	10	**12**
Cardiac arrhythmias	AUROC	.8110	.5191	.7065	.4791	**.7564**	.7512
	AUPRC	.7037	.3609	.5825	.3353	**.6352**	.6144
	#/15	-	3	5	4	7	**8**

5.3 Smooth Conditions, Sharp Synthetic Data

The ideal conditional generator should be capable of generating high quality data according to given conditions. Here we define two terms, sharpness and smoothness. The generator must be *sharp* as generated data reflects attributes of given conditions clearly when expected. For instance, a patient with 100% chance of diabetes must have obvious observations/or medications. Second, it must be *smooth*, which means that it has control over what is generated with a realistic continuous distribution of the data. In other words, it should learn to transit between disease stages, which is natural for chronic diseases. Sharpness is more obvious at the boundaries, aka disease chances are closer to 0 or 1, while smoothness is a characteristic for transitions between stages.



Cardiac arrhythmias

Congestive heart failure

Chronic kidney disease

Diabetes

Hypertension

Fig. 3. Gradually increasing input conditions to measure average probability of generated data according to the random forest model trained on real data.

Having a set of conditions $\tilde{C}$, for the i^{th} disease, we changed $\tilde{c}_i$ increasing from 0 to 1 by 0.1 steps gradually while all other $\tilde{c}_j$ (where $j \neq i$) remained the same as conditions passed for training the GAN. This process lead to creation of 11 data groups. For each group, we measured the average probability assigned by the random forest model trained on real data. Results are shown in Fig. 3. The ideal result in this model would be the solid diagonal line, where for given input condition generated data would get similar probability by the model trained on real data. Considering g_i as the average for the i^{th} group and n as the number of steps (here $n = 10$), we define $\mathit{sharpness} = (g_0 - 0) + (1 - g_n)$ and $\mathit{smoothness} = (\sum_{i=0}^{n} |\frac{i}{n} - g_i|))/(n+1)$. In both metrics lower magnitude is better. As a baseline, when probabilities of all groups are 0.50 (horizontal line in middle), the sharpness and the smoothness are 1 and 0.27 respectively.

Training with smooth labels decreased the sharpness and the smoothness from 0.86 and 0.23 to 0.69 and 0.18 average among all diseases. Of note, the other conditions in the input also affect the output of the generator. For this reason, reaching absolute 0 or 1 probability over a reasonable set of conditions is unrealistic. For instance, a passed vector condition with high diabetes and hypertension with exactly 0 % chance of CKD is not possible. This can explain why the curves are bent when they get closer to 0 or 1.

Table 3. Sample synthetic CKD cases generated.

#	Lab name	Initial	CKD GAN input probability										
		State	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
1	BUN	54.03	37.82	38.12	40.31	43.99	48.98	53.58	58.59	63.60	67.62	70.79	73.21
	Creatinine	1.53	0.70	0.76	0.88	1.05	1.27	1.50	1.81	2.31	2.92	3.63	4.35
	GFR	27.32	85.49	76.52	64.23	50.61	38.45	28.22	19.47	13.14	8.67	5.82	4.18
2	BUN	33.90	29.60	30.09	31.09	32.90	35.29	38.60	42.07	45.19	47.49	48.97	50.73
	Creatinine	0.58	0.41	0.42	0.46	0.53	0.64	0.78	0.98	1.23	1.52	1.87	2.29
	GFR	84.76	100.85	97.51	93.70	87.63	79.58	69.58	56.57	43.04	31.38	22.59	15.78

In Table 3, we show how samples made by the SMOOTH-GAN change over given CKD conditions for three important features: blood urea nitrogen (BUN), glomerular filtration rate (GFR) and creatinine in serum/plasma. The initial state is what is generated by passing a set of random conditions and random noise to the generator. We set the CKD condition from 0 to 1 to get a spectrum of potential states for this encounter.

5.4 Counterfactual Disease Generation

Generally, counterfactuals are hypothetical "what would happen/have happened if" questions. We designed a very specific experiment to show GANs can also be used for generating special combinations of diseases in healthcare. We removed all cases with both hypertension and diabetes from the training set, and we call this new set the "pruned training set". Then we trained our GAN on this new training set to measure whether the model can produce acceptable encounters having both conditions. We chose these two diseases to have a reasonable amount of data for validating the results as this combination happens often in EHR data. Similarly, we measure machine learning efficacy as the ultimate test. In Table 4, we measure RF performance when trained on 1) real data 2) synthetic data from a GAN model trained on the original training set 3) synthetic data from a GAN model trained on the pruned training set. We observe that while the pruned model does not outperform other models in detecting positive cases, it has captured a significant amount of the existing patterns.

Table 4. Performance for counterfactual disease generation

Disease name	Metric	Real	Original training set	Pruned training set
Hypertension & Diabetes	AUROC	.9106	.8720	.8317
	AUPRC	.7122	.6223	.5252
	# /15	–	10	8

There are several challenges for this type of experiment. First, for disease pairs that usually occur together, there might be very few examples of either disease alone. Thus, the pruned dataset would be inefficient. For instance, 88% of patients with CKD also had hypertension. Secondly, the combination of two diseases might be rare when diseases are less relevant to each other, leaving the validation set very small. Last, it is time consuming to train a GAN model for all permutations. We believe that this approach has high potential and can lead to the discovery of novel patterns, which we will further study with larger datasets in our future work.

6 Conclusion

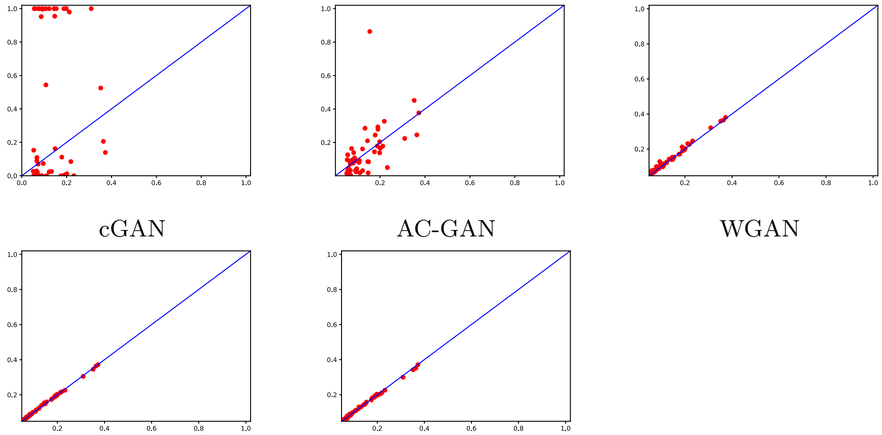
In this paper, we propose SMOOTH-GAN, a new approach for generating synthetic EHR data based on recent advances in generative adversarial networks. We show it is possible to produce high quality synthetic data that maintains important relations and factors in the original data and can be useful for training competitive machine learning models. We define sharpness and smoothness as vital concepts which are applicable in other domains as well. Furthermore, we demonstrate how to create synthetic EHR data with meaningful clinical implications. By combining this approach and Invertible cGANs it is possible to augment existing patient data, as well as helping to produce more accurate machine learning models. Our approach opens doors to new research opportunities and has high potential for generating unseen combinations to support novel research projects such as counterfactual use cases.

Acknowledgments. Authors wish to thank Aryan Arbabi for his constructive comments.

A Appendix

A.1 Binary Data Distribution

As GANs were known to struggle with generating binary values, we added Fig. 4 to illustrate dimension-wise probability for medications comparing real versus synthetic data.



cGAN

AC-GAN

WGAN

WGAN-GP

SMOOTH-GAN

Fig. 4. Dimension-wise probability performance for binary values.

A.2 Is Training Data Memorized by the GAN?

For ensuring privacy and discovering whether the GAN is generating new cases or memorizing the training set, we followed the footsteps of [8] by measuring maximum mean discrepancy (MMD) and applying the three-sample test [4,11]. MMD can answer whether two sets of samples were generated from the same distribution. If the synthetic data is memorized then MMD(synthetic, training) would be *significantly* lower than MMD(synthetic, test). For this reason, we state the null hypothesis as GAN has not memorized the training set, and consequently MMD(synthetic, test) $\leq$ MMD(synthetic, training). We sampled from these three datasets 35 times and calculated MMDs and p-values for the hypothesis. The mean p-value with its standard deviation is 0.26 ± 0.15 which means we cannot reject the null hypothesis and we can establish that GAN did not memorize from the training set.

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International Conference on Machine Learning. pp. 214–223 (2017)
2. Ashfaq, A., Sant'Anna, A., Lingman, M., Nowaczyk, S.: Readmission prediction using deep learning on electronic health records. Journal of biomedical informatics 97, 103256 (2019)
3. Avati, A., Jung, K., Harman, S., Downing, L., Ng, A., Shah, N.H.: Improving palliative care with deep learning. BMC medical informatics and decision making **18**(4), 122 (2018)
4. Bounliphone, W., Belilovsky, E., Blaschko, M.B., Antonoglou, I., Gretton, A.: A test of relative similarity for model selection in generative models. arXiv preprint arXiv:1511.04581 (2015)

5. Che, Z., Cheng, Y., Zhai, S., Sun, Z., Liu, Y.: Boosting deep learning risk prediction with generative adversarial networks for electronic health records. In: 2017 IEEE International Conference on Data Mining (ICDM). pp. 787–792. IEEE (2017)
6. Choi, E., Bahadori, M.T., Schuetz, A., Stewart, W.F., Sun, J.: Doctor AI: predicting clinical events via recurrent neural networks. In: Machine Learning for Healthcare Conference, pp. 301–318 (2016)
7. Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W.F., Sun, J.: Generating multi-label discrete patient records using generative adversarial networks. In: Machine Learning for Healthcare Conference, pp. 286–305 (2017)
8. Esteban, C., Hyland, S.L., Rätsch, G.: Real-valued (medical) time series generation with recurrent conditional gans. arXiv preprint arXiv:1706.02633 (2017)
9. Goodfellow, I.: Nips 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160 (2016)
10. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
11. Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., Smola, A.J.: A kernel method for the two-sample-problem. In: Advances in neural information processing systems, pp. 513–520 (2007)
12. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems 30, pp. 5767–5777. Curran Associates, Inc. (2017), http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf
13. Liu, S., Kailkhura, B., Loveland, D., Han, Y.: Generative counterfactual introspection for explainable deep learning. arXiv preprint arXiv:1907.03077 (2019)
14. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
15. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. In: Proceedings of the 34th International Conference on Machine Learning. **70**, pp. 2642–2651. JMLR. org (2017)
16. Perarnau, G., Van De Weijer, J., Raducanu, B., Álvarez, J.M.: Invertible conditional gans for image editing. arXiv preprint arXiv:1611.06355 (2016)
17. Rashidian, S., et al.: Deep learning on electronic health records to improve disease coding accuracy. In: AMIA Summits on Translational Science Proceedings. vol. 2019, p. 620 (2019)
18. Steiner, C.A., Barrett, M.L., Weiss, A.J., Andrews, R.M.: Trends and projections in hospital stays for adults with multiple chronic conditions, 2003–2014: Statistical brief# 183. Healthcare Cost and Utilization Project (HCUP) Statistical Briefs. Rockville: Agency for Health Care Policy and Research (US) (2006)
19. Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling tabular data using conditional gan. In: Advances in Neural Information Processing Systems, pp. 7333–7343 (2019)