Generating Longitudinal Synthetic EHR Data with Recurrent Autoencoders and Generative Adversarial Networks

Siao Sun[1(✉)], Fusheng Wang[2,3], Sina Rashidian[2], Tahsin Kurc[3], Kayley Abell-Hart[3], Janos Hajagos[3], Wei Zhu[1], Mary Saltz[3], and Joel Saltz[3]

[1] Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY, USA
siao.sun@stonybrook.edu
[2] Department of Computer Science, Stony Brook University, Stony Brook, NY, USA
[3] Department of Biomedical Informatics, Renaissance School of Medicine at Stony Brook University, Stony Brook, NY, USA

Abstract. Synthetic electronic health records (EHR) can facilitate effective use of clinical data in software development, medical education, and medical research without the concerns of data privacy. We propose a novel Generative Adversarial Network (GAN) approach, called Longitudinal GAN (LongGAN), that can generate synthetic longitudinal EHR data. LongGAN employs a recurrent autoencoder and the Wasserstein GAN Gradient Penalty (WGAN-GP) architecture with conditional inputs. We evaluate LongGAN with the task of generating training data for machine/deep learning methods. Our experiments show that predictive models trained with synthetic data from LongGAN achieve comparable performance to those trained with real data. Moreover, these models have up to 0.27 higher AUROC and up to 0.21 higher AUPRC values than models trained with synthetic data from RCGAN and TimeGAN, the two most relevant methods for longitudinal data generation. We also demonstrate that LongGAN is able to preserve patient privacy in a given attribute disclosure attack setting.

Keywords: Deep learning · Machine learning · Electronic health records · Generative models · Synthetic data generation

1 Introduction

Electronic health record (EHR) systems capture vast amounts of digital data about patients' health status, their medical and treatment histories, and clinical outcomes. These data provide opportunities to improve healthcare delivery, reduce medical costs and, when integrated with genomic and imaging data, can enable the development of strategies for personalized medicine. However, the use of EHR data in medical research and software development is often impeded by the complexities of regulatory oversight. Because electronic health records contain patients' information, data access and

E. K. Rezig et al. (Eds.): Poly 2021/DMAH 2021, LNCS 12921, pp. 153–165, 2021.
https://doi.org/10.1007/978-3-030-93663-1_12

sharing are strictly controlled by rules and processes to protect patient privacy. Getting approvals for access to de-identified clinical data can be time consuming. Approvals are generally granted for specific subsets of data (new approvals are required, if a study later needs additional data subsets) with limits on how the data can be shared within and among research teams. Higher security requirements on computing and storage infrastructure put additional burden on EHR based medical and informatics research. The process of data de-identification is also time consuming and expensive, especially for large EHR datasets. Moreover, de-identified data can still pose privacy and security risks [1].

Realistic synthetic datasets that maintain the statistical properties of real datasets can mitigate the complexities of clinical data access by eliminating (or significantly reducing) privacy and security risks and can complement de-identified real clinical data in informatics and medical research [2–11]. For instance, synthetic datasets can be used for data analysis3 and cohort identification tasks [2]. They can also replace or augment real data for a more efficient development and evaluation of computerized analysis methods [4, 5]. Realistic synthetic EHR data can, in particular, benefit deep learning analysis workflows, which often require large volumes of data to train accurate and robust models. Large longitudinal EHR datasets, for example, are critical to the development of reliable predictive models, which are generally based on recurrent neural networks (RNN), such as long short-term memory [12] (LSTM) architectures. However, there are challenges in generating realistic synthetic datasets. Data heterogeneity, large numbers of data elements and types, irregularities in data, and missing values make it arduous to implement efficient methods that can produce realistic synthetic data.

We propose a novel deep learning method, coined as Longitudinal GAN (LongGAN), for generating longitudinal synthetic EHR data. A trained LongGAN model generates high-quality clinical data containing continuous laboratory and medication values for given diseases for a time period of 72 h. It can be applied to any continuous-valued longitudinal data for any reasonable time range given.

Deep learning has in recent years become the preferred method for data analysis in a wide range of applications including analysis of clinical data for identification of disease risk, outcome prediction, and the extraction and classification of clinical information. For example, deep learning methods have been used to analyze EHR data to identify the risk of opioid use disorder [13] and opioid overdose [14] in population studies and to detect miscoded diabetes diagnosis codes for quality improvement [15]. Deep learning methods have also been successfully applied to synthetic data generation in many application domains, such as text-to-image synthesis [16], video generation [17], and music generation18. Most synthetic data methods employ the Generative Adversarial Network [19] (GAN) architecture, which consists of a generator component and a discriminator (or a critic) component. The generator produces synthetic data, whereas the discriminator distinguishes between real and synthetic data. The adversarial relationship between the generator and the discriminator forces the generator to learn to produce realistic synthetic data. Several recent projects have employed GANs

for synthetic EHR data generation [20–27]. Medical Generative Adversarial Network (MedGAN) [20] implements a method for generating discrete data elements (medication codes and diagnosis codes). SMOOTH-GAN [21] demonstrated GANs would generate more realistic synthetic data when binary labels are converted to continuous values by using imperfect machine learning models as heuristic functions for generating laboratory values and medications as a snapshot of patients' records. However, most of the previous efforts have focused on producing non-longitudinal synthetic data that represent a snapshot of a patient's medical history. Applications of GANs for synthetic time-series clinical data remain scarce, owing mainly to the fact that generating sequences requires the generated data to have not only similar overall distribution of attributes, but also similar temporal dynamics to the real sequences. Some recent efforts have resulted in methods for generating longitudinal synthetic data. Recurrent Conditional Generative Adversarial Networks (RCGAN) [22] used a RNN architecture for both the generator and the discriminator and took conditional input at each time step. The authors evaluated the performance on the eICU Collaborative Research Database with four selected regularly sampled features. Time-series Generative Adversarial Networks (TimeGAN) [23] introduced supervised loss to enforce temporal dynamic preservation and trained the generator and the discriminator in embedded space. The authors of TimeGAN measured its success on a discrete-valued lung cancer dataset. Dual Adversarial Autoencoder (DAAE) [24] made use of an inner GAN and an outer GAN to learn set-valued sequences of medical entities such as diagnosis codes.

LongGAN takes advantage of recurrent autoencoders and the Wasserstein Generative Adversarial Network with Gradient Penalty [28] (WGAN-GP) architecture. Recurrent autoencoders have been successfully applied to multivariate time series analysis such as forecasting [29] and anomaly detection [30]. They can learn useful representations of sequences while preserving temporal dynamics during the reconstruction. LongGAN leverages this property of recurrent autoencoders and adapts it to train an autoencoder model to generate realistic sequences. Our work differs from the previous work as follows: 1) Unlike regularly sampled bedside data, our data are irregularly sampled with many missing values. 2) Our data contains many features, rather than only a few handcrafted features. 3) Conditions are combined to generate realistic longitudinal data.

We evaluated the performance of LongGAN by training a logistic regression model, a random forest model, and a two-layer long short-term memory (LSTM) network model to predict acute kidney injury (AKI). These models represent examples of linear models, nonlinear models, and deep learning models, respectively. The experimental results show that predictive models trained with synthetic data from LongGAN achieve comparable Area Under the Receiver Operating Characteristics (AUROC) and Area Under the Precision Recall Curve (AUPRC) values to models trained with real data. In addition, synthetic datasets from LongGAN lead to much better models, with up to 0.27 higher AUROC and up to 0.21 higher AUPRC values, compared with synthetic data from RCGAN and TimeGAN, the two most relevant GAN-based methods for synthetic longitudinal data generation.

Beyond the realisticness of synthetic data, a key concern is protecting patient privacy (i.e., an attacker should not be able to discover the identities of patients from a synthetic dataset). We examined this aspect of LongGAN in the context of attribute disclosure attacks. The experimental results show that an attacker, who has a subset of attributes from the real dataset, could achieve a mean accuracy of 20% in predicting missing attributes with k-nearest neighbors (KNN) estimation using the synthetic dataset generated by LongGAN. This value is lower than the mean accuracy of 26% that the attacker could achieve without access to the synthetic dataset using a population median method.

2 Methods

2.1 Architecture of LongGAN

The proposed method consists of a recurrent autoencoder network and a GAN network as is shown in Fig. 1. A recurrent autoencoder is a neural network trained to copy its input sequence to its output sequence31. More specifically, it can be viewed as having two parts: the encoder Enc takes sequential data X and maps it to a dense representation h, then the decoder Dec takes h and tries to reconstruct the input from it. Here, $X = (s_1, s_2, \ldots, s_T)$ is a time-ordered sequence of vectors. Each vector $s_i = \left(s_i^1, s_i^2, \ldots, s_i^C\right), 1 \leq i \leq T$ represents C features at the time point*i*. In our implementation, the encoder and decoder both have three LSTM layers. We aim to minimize the reconstruction loss, which is:

$$\sum_{X \in D} ||X - X'||^2$$

where D is the dataset, and $X' = Dec(h) = Dec(Enc)(X)$ is the reconstruction of X.

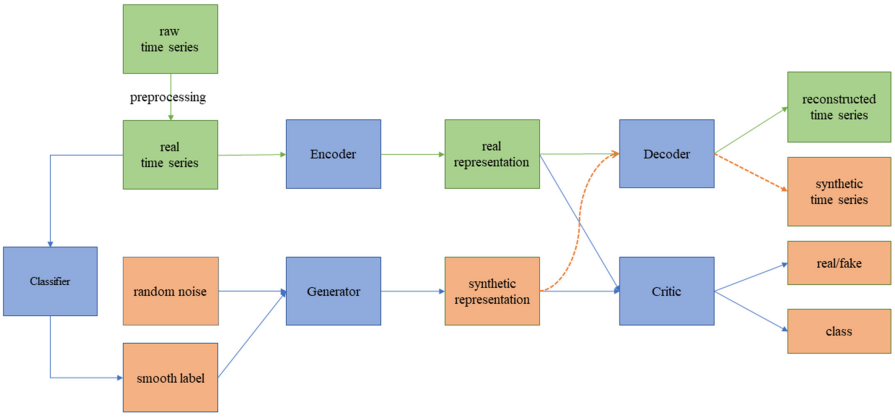

Fig. 1. Architecture of the proposed LongGAN model.

The GAN network is based on the WGAN-GP architecture with conditional inputs. A GAN consists of two components: a generator $G(z;\theta_g)$ and a discriminator $D(h;\theta_d)$. The generator takes random noise and tries to generate samples that follow the same distribution of the real data. Meanwhile, the discriminator receives both real and generated data, and tries to detect whether a sample is real or fake. Ideally, the optimal generator G^* would generate samples that are indistinguishable from real samples, and the discriminator would be forced to make a random guess. Conditional GANs [32] (cGANs) are extensions of GANs where the generator takes not only random noise but also some auxiliary information such as labels, to help with the generation. The objective of a conditional GAN is:

$$min_G max_D V(\theta_g, \theta_d) = E_{h\sim P_h}[\log(D(h|y))] + E_{z\sim P_z}[\log(1 - D(G(z|y))]$$

Here h is the output of the pre-trained encoder, i.e. representation of real longitudinal data, P_h represents the distribution of real representation, P_z is the distribution of random noise (here we used Gaussian distribution), and y is the conditional input. WGAN-GP is an extension of the basic GAN architecture that improves the stability when training the model. Compared to the original GAN, it uses the Wasserstein distance instead of the Jenson-Shannon (JS) divergence, replaces the discriminator with a critic that scores the realness or fakeness of a given sample, and adds gradient penalty to enforce Lipschitz constraints on the critic. The objective function of WGAN-GP is:

$$L = E_{\tilde{h}\sim P_g}\left[D\left(\tilde{h}\right)\right] - E_{h\sim P_h}[D(h)] + \lambda E_{\widehat{h}\sim P_{\widehat{h}}}[(\|\nabla_{\widehat{h}} D(\widehat{h})\|_2 - 1)^2]$$

The synthetic representation $\tilde{h}$ was fed to the pre-trained encoder to generate synthetic longitudinal data:

$$\tilde{X} = Dec\left(\tilde{h}\right) = Dec(G(z|y))$$

In our method, the generator has two leakyRelu hidden layers with $\alpha = 0.2$, each followed by a batch normalization layer, and the output layer is tanh. The critic has two leakyRelu hidden layers with $\alpha = 0.2$. The output layer is linear.

2.2 Training LongGAN

We extracted inpatient encounter data for adults (18+) from the Cerner Health Facts database [33–35], a large multi-institutional de-identified database derived from EHRs and administrative systems. The extracted data were mapped to the OHDSI Common Data Model (version 5.3) and vocabulary release (2/10/2018) [36]. We randomly chose two facilities (131 and 143) from the 10 highest volume inpatient facilities and extracted encounters from 1/1/2016 to 12/31/2017 for the experimental evaluation of the proposed method.

Medications and laboratory tests with no less than 5% appearance rate by encounter in both facilities were extracted, and the raw values were converted to quantiles. We further extracted encounters with length of stay no less than 72 h and sampled one medication/laboratory test per hour. If there were more than one measurement in an hour, medians were computed. Diagnosis codes were mapped from the International Classification of Diseases (ICD) codes to Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT). SNOMED codes and the descendant codes for Acute Kidney Injury (AKI) were combined and used as labels for our study.

The extracted datasets were extremely sparse because not all patients have measurements of all medications/laboratory tests every hour, and thus imputation was necessary. There are many approaches to impute time-series data. Here we used the interpolation part of the Interpolation-Prediction Network [37]. The Interpolation-Prediction Network is a semi-parametric network designed for irregularly sampled multivariate time series, taking into account correlations across all time series from different dimensions.

After the preprocessing, we obtained multidimensional longitudinal data for every patient, where each dimension represents the trajectory of a specific medication/laboratory test measurement from the first 72 h of hospitalization. We then trained a classifier to get smooth labels [21, 38] of AKI. More specifically, we trained a random forest model on the training set of real longitudinal EHR data with AKI as labels to assign probabilities of patients' developing AKI, and then adjusted these probabilities to obtain smooth labels. The adjustment is done as follows:

$$\text{SmoothLabel}\left(\text{X}_{\text{prob}}, \text{X}_{\text{label}}\right) = \begin{cases} 0.49, & \textit{if } \text{X}_{\text{prob}} > 0.5 \textit{ and } \text{X}_{\text{label}} == 0 \\ 0.51, & \textit{if } \text{X}_{\text{prob}} < 0.5 \textit{ and } \text{X}_{\text{label}} == 1 \\ \text{X}_{\text{prob}}, & \textit{otherwise} \end{cases}$$

Here X_prob is the probability of getting AKI assigned by the trained classifier, and X_label is the original binary valued label for AKI.

To train a synthetic data generation model, we first pre-trained the encoder and decoder with the real EHR data with reconstruction loss. We then took the output of the encoder, i.e., the representation of the input data, and trained the WGAN-GP to produce synthetic representations. The smooth labels of AKI were used as conditional input for both the generator and the discriminator. Finally, the generated representations were input into the trained decoder to obtain synthetic longitudinal data.

The method was implemented in Python v3.6. The random forest and logistic regression method were implemented using the scikit-learn package [39]. The recurrent encoder network and the GAN network were developed using Tensorflow [40]. Other libraries used include Python Numpy41, Python Pandas [42], and Python Scipy [43]. Training was performed on an NVIDIA Tesla V100 (16 GB RAM).

3 Results

3.1 Evaluation of Realism

We have evaluated the performance of LongGAN by training traditional machine learning models and RNNs to predict whether or not a patient will develop AKI based on the medication and laboratory results from the first 72 h of hospitalization. In our experiments we used logistic regression, random forest, and a two-layer LSTM network as examples of linear models, nonlinear models, and neural networks, respectively. In each case we trained two models, one using the real training dataset and the other using the synthetic dataset, and then evaluated both models on a real test dataset. This approach, called Train on Synthetic and Test on Real (TSTR), is a common mechanism with which to evaluate the realism of synthetic data [21, 22]. Since logistic regression and random forest are not designed for time-series data, we flattened the sequence along the time dimension as input for these two algorithms. We measured the performances of the models with AUROC and AUPRC as they are commonly used metrics for TSTR [21, 22].

We compared our method with RCGAN and TimeGAN. Since TimeGAN is not designed for conditional generation, we trained two TimeGAN models on positive cases and negative cases separately to generate synthetic data with both cases. Table 1 shows the experimental evaluation results. Our results demonstrate that the models trained on synthetic datasets generated by LongGAN have performances closer to those trained on real datasets than other synthetic datasets generated by RCGAN and TimeGAN. Models trained with synthetic data from LongGAN achieved up to 0.27 higher AUROC and up to 0.21 higher AUPRC values than models trained with data from RCGAN and TimeGAN.

Table 1. Performance of trained predictive models on real and synthetic datasets.

Predictive model	Metric	Real	RCGAN	TimeGAN	LongGAN
Logistic regression	AUROC	0.80	0.57	0.61	0.74
	AUPRC	0.57	0.34	0.36	0.51
Random forest	AUROC	0.86	0.50	0.71	0.77
	AUPRC	0.70	0.29	0.50	0.51
LSTM network	AUROC	0.83	0.63	0.67	0.77
	AUPRC	0.67	0.39	0.45	0.52

In the next set of experiments, we examined whether models trained with the synthetic dataset selected a similar set of features for prediction compared with models trained with the real dataset. To this end, we extracted the top 15 most important features of the random forest models trained with the real and synthetic datasets. Table 2 shows the list of features from each random forest model. Our experiments show that 10 features overlap between the two models.

Table 2. Top 15 most important features of random forest model trained on real/synthetic datasets.

Top 15 features from random forest model trained on the real dataset	Top 15 features from random forest model trained on the synthetic dataset
Creatinine [Mass/volume] in Serum or Plasma	Creatinine [Mass/volume] in Serum or Plasma
Creatinine [Mass/volume] in Urine	Neutrophils/100 leukocytes in Blood by Automated count
Chloride [Moles/volume] in Serum or Plasma	Aspartate aminotransferase [Enzymatic activity/volume] in Serum or Plasma
Ferritin [Mass/volume] in Serum or Plasma	Respiratory rate
Phosphate [Mass/volume] in Serum or Plasma	Phosphate [Mass/volume] in Serum or Plasma
Respiratory rate	Eosinophils/100 leukocytes in Blood by Automated count
Sodium [Moles/volume] in Serum or Plasma	Chloride [Moles/volume] in Serum or Plasma
Iron [Mass/volume] in Serum or Plasma	Ferritin [Mass/volume] in Serum or Plasma
Glasgow coma scale	Protein [Mass/volume] in Serum or Plasma
Aspartate aminotransferase [Enzymatic activity/volume] in Serum or Plasma	Diastolic blood pressure
Basophils/100 leukocytes in Blood by Automated count	paracetamol
Glucose [Mass/volume] in Serum or Plasma	Creatinine [Mass/volume] in Urine
Potassium [Moles/volume] in Serum or Plasma	Iron [Mass/volume] in Serum or Plasma
Mean blood pressure	Glucose [Mass/volume] in Serum or Plasma
Cholesterol [Mass/volume] in Serum or Plasma	Basophils/100 leukocytes in Blood by Automated count

3.2 Evaluation of Privacy Preservation

A critical requirement for a synthetic EHR data generator is that it must preserve patient privacy. In this section we evaluate this aspect of our method with respect to attribute disclosure attacks. Attribute disclosure occurs when attackers can derive target attributes about a patient based on key attributes that they already know about the patient [8, 44]. This is a prominent issue for synthetic datasets as attackers might gain sensitive knowledge of real patients based on similar records in a given synthetic dataset.

We assume the attacker has full access to the synthetic dataset and partial access to the real dataset. This is a commonly adopted setting for evaluating the attribute disclosure risk [20, 45]. More specifically, we randomly sampled 1% of patients from the real training set as the compromised records, flattened them along the time dimension, and randomly masked 10% of the attributes as the set of target attributes that are unknown to the attacker.

While there are different potential attack methods for synthetic dataset [20, 46, 47], in this paper, due to space limitation, we focused on KNN estimation, a common method considered for privacy preserving evaluation. For each compromised record, we retrieved its k-nearest neighbors in the synthetic dataset based on the key attributes and estimated the target attribute using the median of corresponding attributes of these k neighbors. We call an estimation accurate, if the relative error of the estimation is below 5%. We used a dummy baseline where the attacker simply guesses the median value in the population. Here it is 0.5 since our data are in quantile. This simulates the attacker's behavior when they have no knowledge of the original dataset or the synthetic dataset and have to make estimations uniformly at random [46].

The idea is that a privacy-preserving synthetic dataset should avoid providing the attacker with additional knowledge for better estimation of target attributes, in order to minimize the risk of attribute disclosure. We repeated the experiment for 30 times, with different records of patients randomly selected and different attributes randomly masked and mean accuracy computed for all masked attributes. The experiments showed that with the KNN estimation the attacker on average achieved a mean accuracy of 20%, while with the estimation of the population median the mean accuracy was 26%. The paired samples t-test of the mean accuracies from different experiments resulted in a p-value of 7.12e−23. This indicates that the mean accuracy from the KNN estimation was significantly smaller than that of random guess, suggesting that in the given scenario, an attacker using KNN estimation cannot do better than random guess.

4 Discussion

Generating synthetic clinical data has great potential for researchers to conduct competitive and reproducible research with electronic health records without privacy concerns. However, very few works have tackled the problems of generating continuous time-series clinical data. We have proposed a model that combines a recurrent autoencoder and WGAN-GP to generate realistic time-series data containing continuous laboratory and medication values for given diseases. While we focused on a specific disease (AKI) in our experimental evaluation, the methodology is universal and can be applied in the context of other diseases.

4.1 Comparison with Previous Work

In Esteban's work on RCGAN, they used RNNs (LSTM) as the generator and the discriminator, and the labels were fed to the generator and the discriminator at every time step [22]. In Yoon's work on TimeGAN, they used RNNs as embedding and recovery functions to provide mappings between feature space and latent space, and then trained the GAN within the latent space [23]. The GAN aspect of TimeGAN also utilized the RNNs as both the generator and the discriminator. In addition, another RNN (called supervisor in the paper) was added to enforce the generated longitudinal data having similar temporal relationships to the real longitudinal data.

GANs and RNNs can both be hard to train [28, 48], and using the RNN structures in GANs would intuitively introduce instability in training. Compared with previous studies, the key difference of our study is that we managed to bypass the RNN structure in the GAN. We accomplished this by taking advantage of a pre-trained recurrent autoencoder and transformed the problem of generating sequences to the problem of generating dense representations of sequences. Since the generated representations are input to the decoder, which was trained on real longitudinal data, the generated longitudinal data would maintain similar temporal dynamics to the real dataset. Our model also differs from previous work in that we took advantage of WGAN-GP and smooth labels, which made the training more stable. Moreover, our model requires minimal domain knowledge to make hand-crafted features, rendering it more generalizable.

Our model achieved much better AUROC and AUPRC values than the baseline models in predictive modeling tasks. The significant overlap of top features between models trained with synthetic data and those trained with real data suggests LongGAN can generate realistic synthetic data which can in turn be used to complement or replace real data for training machine learning models. The experiments on attribute disclosure demonstrated that an attacker cannot reliably obtain additional information about real patients with help of our generated dataset, which minimizes the concerns for privacy issues.

4.2 Limitations

The datasets extracted from the Health Facts database contain many missing values, because not all patients have measurements of all medications/laboratory tests every hour. We performed imputation to obtain fixed-length longitudinal data to fit the model. However, in this process we also eliminated any patterns of the missing data itself, which could contain useful information about patients [49]. While our method generates synthetic data that are similar to the imputed data, it does not have patterns of missing data like the original datasets do.

5 Conclusion and Future Work

LongGAN is a new approach to generating synthetic longitudinal EHR data. It can produce synthetic datasets that enable training of machine/deep learning models with comparable predictive performances to those of models trained with real data. For future work, we shall investigate how to combine the transformer [50] architecture with GAN and implement extensions to produce synthetic data on demographics and preserve patterns of missing data. Transformer networks have achieved great success in natural language processing tasks [51, 52] and have been shown to be powerful tools for extracting useful features of sequences [53, 54]. We will also explore other aspects of privacy attack and preservation and differentially private training methods [55, 56], in order to further minimize or eliminate the risk of information leakage.

References

1. Rothstein, M.A.: Is deidentification sufficient to protect health privacy in research? Am J Bioeth. **10**(9), 3–11 (2010)
2. Foraker, R.E., Yu, S.C., Gupta, A., Michelson, A.P., Pineda Soto, J.A., Colvin, R., et al.: Spot the difference: Comparing results of analyses from real patient data and synthetic derivatives. JAMIA Open. **3**(4), 557–566 (2020)
3. Benaim, A.R., et al.: Analyzing medical research results based on synthetic data and their relation to real data results: Systematic comparison from five observational studies. JMIR Med. Inform. **8**(2), e16492 (2020)
4. Guo, A., Foraker, R.E., MacGregor, R.M., Masood, F.M., Cupps, B.P., Pasque, M.K.: The use of synthetic electronic health record data and deep learning to improve timing of high-risk heart failure surgical intervention by predicting proximity to catastrophic decompensation. Front. Digit. Health **44** (2020)
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. 787–92 (2017)
6. Walonoski, J.A., Kramer, M., Nichols, J., Quina, A., Moesel, C., Hall, D., et al.: Synthea: an approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. J. Am. Med. Inf. Assoc. JAMIA. **25**, 230–238 (2018)
7. Dube, K., Gallagher, T.: Approach and Method for Generating Realistic Synthetic Electronic Healthcare Records for Secondary Use. In: Gibbons, J., MacCaull, W. (eds.) FHIES 2013. LNCS, vol. 8315, pp. 69–86. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-53956-5_6
8. Goncalves, A., Ray, P., Soper, B., Stevens, J., Coyle, L., Sales, A.P.: Generation and evaluation of synthetic patient data. BMC Med. Res. Method. **20**(1), 1–40 (2020)
9. McLachlan, S., Dube, K., Gallagher, T., Simmonds, J.A., Fenton, N.: Realistic Synthetic Data Generation: The ATEN Framework. In: Cliquet Jr., A., et al. (eds.) BIOSTEC 2018. CCIS, vol. 1024, pp. 497–523. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29196-9_25
10. Pollack, A.H., Simon, T.D., Snyder, J., Pratt, W.: Creating synthetic patient data to support the design and evaluation of novel health information technology. J. Biomed. Inf. **95**, 103201 (2019)
11. Walonoski, J., et al.: Synthe, novel coronavirus (covid-19) model and synthetic data set. Intell. Based Med. **1**, 100007 (2020)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–1780 (1997)
13. Dong X, et al.: Identifying risk of opioid use disorder for patients taking opioid medications with deep learning. arXiv preprint arXiv:201004589 (2020)
14. Dong, X., et al.: Predicting opioid overdose risk of patients with opioid prescriptions using electronic health records based on temporal deep learning. J. Biomed. Inf. **116**, 103725 (2021)
15. Rashidian, S., et al.: Detecting miscoded diabetes diagnosis codes in electronic health records for quality improvement: temporal deep learning approach. JMIR Med. Inform. **8** (12), e22649 (2020)
16. Tao, M., Tang, H., Wu, S., Sebe, N., Wu, F., Jing, X.: Df-gan: deep fusion generative adversarial networks for text-to-image synthesis. ArXiv. abs/2008.05865 (2020)
17. Clark, A., Donahue, J., Simonyan, K.: Adversarial video generation on complex datasets. arXiv: Computer Vision and Pattern Recognition (2019)

18. Engel, J., Agrawal, K.K., Chen, S., Gulrajani, I., Donahue, C., Roberts, A.: Gansynth: adversarial neural audio synthesis. ArXiv; abs/1902.08710 (2019)
19. Goodfellow, I., et al.: Generative adversarial nets. In: NIPS (2014).
20. Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W., Sun, J.: Generating multi-label discrete electronic health records using generative adversarial networks. ArXiv; abs/1703.06490 (2017)
21. Rashidian, S., et al.: SMOOTH-GAN: towards sharp and smooth synthetic ehr data generation. In: Michalowski, M., Moskovitch, R. (eds.) Artificial Intelligence in Medicine. AIME 2020. Lecture Notes in Computer Science, vol. 12299. Springer, Cham (2020)
22. Esteban, C., Hyland, S.L., Rätsch, G.: Real-valued (medical) time series generation with recurrent conditional GANs. ArXiv. abs/1706.02633 (2017)
23. Yoon, J., Jarrett, D., Schaar, M.V.D.: Time-series generative adversarial networks. In: NeurIPS (2019)
24. Lee, D., Yu, H., Jiang, X., Rogith, D., Gudala, M., Tejani, M., et al.: Generating sequential electronic health records using dual adversarial autoencoder. J. Am. Med. Inform. Assoc. **27** (9), 1411–1419 (2020)
25. Jordon, J., Yoon, J., Schaar, M.V.D.: Pate-gan: generating synthetic data with differential privacy guarantees. In: ICLR (2019)
26. Baowaly, M.K., Lin, C., Liu, C.-L., Chen, K.-T.: Synthesizing electronic health records using improved generative adversarial networks. J. Am. Med. Inform. Assoc. **26**(228), 41 (2019)
27. Yoon, J., Drumright, L.N., Van Der Schaar, M.: Anonymization through data synthesis using generative adversarial networks (ads-gan). IEEE J. Biomed. Health Informatics. **24**(8), 2378–2388
28. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein gans. In: NIPS (2017)
29. Nguyen, H.D., Tran, K.P., Thomassey, S., Hamad, M.: Forecasting and anomaly detection approaches using LSTM and LSTM autoencoder techniques with the applications in supply chain management. Int. J. Inf. Manag. **57**, 102282 (2021)
30. Chawla, A., Lee, B., Jacob, P., Fallon, S.: Bidirectional LSTM autoencoder for sequence based anomaly detection in cyber security. Int. J. Simulation: Syst., Sci. Technol. (2019)
31. Wong, T., Luo, Z.: Recurrent auto-encoder model for multidimensional time series representation (2018)
32. Mirza, M, Osindero, S.: Conditional generative adversarial nets. ArXiv. abs/1411.1784 (2014)
33. Al-Shawwa, B., Glynn, E., Hoffman, M.A., Ehsan, Z., Ingram, D.G.: Outpatient health care utilization for sleep disorders in the cerner health facts database. J. Clin. Sleep Med. **17**(2), 203–209 (2021)
34. Petrick, J.L., Nguyen, T., Cook, M.B.: Temporal trends of esophageal disorders by age in the cerner health facts database. Ann. Epidemiol. **26**(2), 151–4.e4 (2016)
35. DeShazo, J.P., Hoffman, M.: A comparison of a multistate inpatient ehr database to the hcup nationwide inpatient sample. BMC Health Services Res. **15**(1), 1–8 (2015)
36. Hripcsak, G., Ryan, P.B., Duke, J.D., Shah, N.H., Park, R.W., Huser, V., et al.: Characterizing treatment pathways at scale using the ohdsi network. Proc Natl Acad Sci U S A. **113**(27), 7329–7336 (2016)
37. Shukla, S.N., Marlin, B.M.: Interpolation-prediction networks for irregularly sampled time series. ArXiv ;abs/1909.07782 (2019)
38. Müller, R., Kornblith, S., Hinton, G.E.: When does label smoothing help? In: NeurIPS (2019)

39. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al.: Scikit-learn: Machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
40. Abadi, M., et al.: Tensorflow: a system for large-scale machine learning. In: OSDI (2016)
41. Oliphant, T.E.: Guide to NumPy (2015)
42. McKinney, W.: Data structures for statistical computing in python (2010)
43. Virtanen, P., et al.: Scipy 1.0: fundamental algorithms for scientific computing in python. Nat. Method. **17**(3), 261–272 (2020).
44. Matwin, S., Nin, J., Sehatkar, M., Szapiro, T.: A review of attribute disclosure control. In: Navarro-Arribas G., Torra V. (eds.) Advanced Research in Data Privacy. Studies in Computational Intelligence, vol. 567. Springer, Cham (2015)
45. Surendra, H., MohanH, S.: A review of synthetic data generation methods for privacy preserving data publishing. Int. J. Sci. Technol. Res. **6**, 95–101 (2017)
46. Hittmeir, M., Mayer, R., Ekelhart, A.: A baseline for attribute disclosure risk in synthetic data. In: Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy (2020)
47. Stadler, T., Oprisanu, B., Troncoso, C.: Synthetic data - a privacy mirage. ArXiv. abs/2011.07018 (2020)
48. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: ICML (2013)
49. García-Laencina, P.J., Sancho-Gómez, J., Figueiras-Vidal, A.R.: Pattern classification with missing data: A review. Neural Comput. Appl. **19**, 263–282 (2009)
50. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., et al.: Attention is all you need. ArXiv. abs/1706.03762 (2017)
51. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., et al.: Roberta: a robustly optimized bert pretraining approach. ArXiv. abs/1907.11692 (2019)
52. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2019)
53. Choi, K., Hawthorne, C., Simon, I., Dinculescu, M., Engel, J.: Encoding musical style with transformer autoencoders. In: ICML (2020)
54. Fang, L., Zeng, T., Liu, C.C., Bo, L., Dong, W., Chen, C.: Transformer-based conditional variational autoencoder for controllable story generation. ArXiv abs/2101.00828 (2021)
55. Toreini, E., et al.: Technologies for trustworthy machine learning: A survey in a socio-technical context. ArXiv. abs/2007.08911 (2020)
56. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found Trends Theor. Comput. Sci. **9**, 211–407 (2014)