2020

Synthesising Tabular Datasets Using Wasserstein Conditional GANS with Gradient Penalty (WCGAN-GP)

Manhar Singh Walia
Technological University Dublin

Brendan Tierney
Technological University Dublin, brendan.tierney@tudublin.ie

Susan McKeever
Technological University Dublin, susan.mckeever@tudublin.ie

Recommended Citation

Walia, M.S., Tierney, B. & McKeever, S. (2020). Synthesising tabular datasets using Wasserstein Conditional GANS with Gradient Penalty (WCGAN-GP).*AICS 2020: 28th Irish Conference on Artificial Intelligence and Cognitive Science*, Dublin Ireland.

Synthesising Tabular Data using Wasserstein Conditional GANs with Gradient Penalty (WCGAN-GP) ⋆

Manhar Walia[1], Brendan Tierney[2], and Susan McKeever[3]

[1] TU Dublin, Dublin, Ireland D18128811@tudublin.ie
[2] TU Dublin, Dublin, Ireland brendan.tierney@tudublin.ie
[3] TU Dublin, Dublin, Ireland susan.mckeever@TUDublin.ie

Abstract. Deep learning based methods based on Generative Adversarial Networks (GANs) have seen remarkable success in data synthesis of images and text. This study investigates the use of GANs for the generation of tabular mixed dataset. We apply Wasserstein Conditional Generative Adversarial Network (WCGAN-GP) to the task of generating tabular synthetic data that is indistinguishable from the real data, without incurring information leakage. The performance of WCGAN-GP is compared against both the ground truth datasets and SMOTE using three labelled real-world datasets from different domains. Our results for WCGAN-GP show that the synthetic data preserves distributions and relationships of the real data, outperforming the SMOTE approach on both class preservation and data protection metrics. Our work is a contribution towards the automated synthesis of tabular mixed data.

Keywords: Synthetic Data · Generative Adversarial Network · GAN · WCGAN-GP · Tabular Data Generation · Euclidean Distance.

1 Introduction

Real-world data is commonly used in the demonstration and evaluation of novel technologies in areas such as software development or data analytics. Machine learning algorithms require sample data to learn from, but data accessibility, insufficient data and privacy constraints have set barriers to the development of certain models. Traditionally, real-world data was anonymized using approaches like k-anonymity, l-diversity, or t-closeness to minimize any disclosure risks. But these privacy perturbation approaches have still been linked to poor privacy protection and semantic suitability [4]. These approaches also result in the loss of usability of the modified data. As a result, the generation of realistic, usable synthetic data offers a solution to overcoming the hurdles of data dissemination.

Data Synthesis has traditionally been done via user specification of the dataset feature characteristics and statistical distributions using a variety of commercial tools such as Mockaroo[4]. More recently, deep learning networks (GANs)

⋆ Supported by TU Dublin.
[4] https://www.mockaroo.com/

[13] have been applied to automatically generating a dataset based on a seeding (real) dataset. GANs are built using an architecture of two neural networks that compete against each other in an adversarial manner with an attempt to generate new samples. Since their inception in 2014, GANs have seen tremendous success in synthesizing realistic images and text [11].

Early GANs suffer from training problems like vanishing gradients and mode-collapse [21, 23], resulting in poor training performance and limited diversity in new samples. A modified GAN variant that addresses these issues, WCGAN-GP (Wasserstein Conditional GANs with Gradient Penalty) [14, 3] is studied for tabular data generation in this paper and its ability to generate a high quality data is examined.

Using three real-world datasets, the quality of data generated by WCGAN-GP is tested on data utility and privacy metrics, and compared to the both the ground truth datasets and Synthetic Minority Oversampling Technique (SMOTE) [7]. We demonstrate that WCGAN-GP outperforms SMOTE in generating data that preserves data patterns along with higher privacy protection.

The contributions of the paper are summarised as follows: (1) A comprehensive proof-of-concept to showcase the success of WCGAN-GP in the generation of synthetic tabular data. (2) A comparison of WCGAN-GP to SMOTE on data utility and privacy metrics across different mixed-type datasets. (3) Contrary to the belief that GANs suffer from training problems, we demonstrate that WCGAN-GP provides a strong modelling performance and stable training on structured data.

2 Related Works

Synthetic data can be generated in two ways. Firstly, by statistical modelling to learn from user-specified distributions or directly from real data. Secondly, by using deep learning to learn from the real data with minimal user inputs.

2.1 Statistical Modelling Approaches

The statistical modelling methods can be classified into process-driven and data-driven [12]. The process-driven methods generate data using handcrafted distributions and do not use real data. These methods require human intervention and are prone to human bias [25]. The data-driven approaches generate synthetic data via the automated learning of the intrinsic patterns from real data. [9] implemented data synthesizers based on machine learning algorithms, but the approaches pose disclosure risks if the classification accuracy is high.

SMOTE is originally developed for oversampling and address the imbalance problem [7]. But it is also used in generating synthetic data to replace the real data [17]. SMOTE is faster to run and can generate a good quality of synthetic data without the need for any hyperparameter optimisation.

2.2 Deep Generative Modelling Approaches

The success of deep generative models in the field of natural language processing has motivated the use of neural networks for data generation. GANs have shown remarkable performance in generating synthetic images and time-series data [8, 10]. However, GANs have had limited testing on structured data [11]. [26] use GANs to create synthetic database and tested it on numerical data. [18] use GANs for data generation - but three out of the four datasets are synthetically created. Further, researchers have noted limitations with GANs when generating labelled data [22] and proposed Conditional GANs (CGANs), where class labels are taken into account [19]. [27] apply CGANs on a numerical data to generate synthetic data, but the scatterplots of the synthetic data indicated signs of mode collapse.

MedGAN [8] uses auto-encoders but is tested for binary and numeric data. Its design does not support different data types in the same model and requires separate models for each data type [5]. TableGAN and VEEGAN work well with numerical data but suffer from mode collapse with categorical data [29].

Even though GANs have shown success in image generation, their training is not easy and unstable [30]. Arjovsky et al. [1] have cited problems of vanishing gradients. There are variants like WGAN and WGAN-GP that provide a more stable training framework [14, 6]. [2] have noted that WGANs can still suffer from unstable training and vanishing gradients. WGAN-GP enforces a regularization term in the form of gradient penalty. WGAN-GP has been implemented on large-scale image and language datasets and shown to provide superior performance over WGANs [14]. WGAN-GP is easily extended to WCGAN-GP by inputting the condition vector, that is target labels. This enables the GAN to learn the distributions specific to each class label and produce higher quality samples for both labels. The ability of WCGAN-GP to draw samples from images has been explored but has not been tested on tabular datasets. The resultant paucity in the current literature is something this research seeks to address.

3 WCGAN-GP Model

For the purposes of presenting our work, we present a brief overview of WCGAN-GP here, but further details can be found in the original work of WGAN-GP [14]. WCGAN-GP uses Wasserstein distance and Gradient Penalty to reduce the occurrence of failure modes associated with GANs. WCGAN-GP is similar to WGAN-GP and the only change is where critic (discriminator) and generator are both conditioned on an extra information of class labels. In WCGAN-GP, the discriminator is called as a critic. Rather than classifying samples as real or fake, the critic predicts values that are large for real and small for fake samples. The structure of WCGAN-GP is shown in Fig. 1.

WCGAN-GP uses gradient penalty to force the norm of gradients to be 1 and comply with 1-Lipschitz constraint. This helps in overcoming the training instability of GANs that occurs when the critic outputs explosive gradients. Thus,

the weights are clipped using the 1-Lipschitz function and the rate of change is bounded. This metric results in faster convergence as the training provides reasonable gradients and the critic becomes more stable and less explosive.

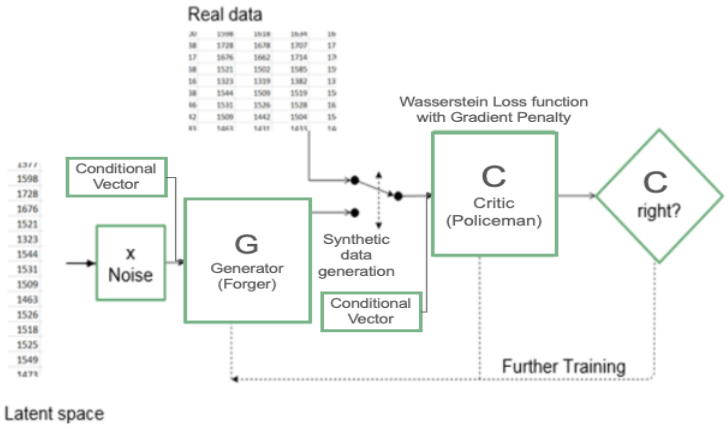

Fig. 1. Structure of WCGAN-GP.

4 Approach and Methodology

In this section, we used three labelled real-world datasets to evaluate WCGAN-GP against a second synthesis technique, SMOTE. The quality of synthetic data was then assessed using a variety of data utility and privacy metrics.

4.1 Datasets

To test our synthesis approach, we selected three real-world datasets from three different domains - Default of Credit Card[5], Cardiovascular Disease[6] and Adult Census[7]. These datasets contain mixed data-types and were chosen for their differences in data type distributions, allowing us to detect whether the methods perform better or worse for specific types of data. Further, the datasets can be potentially categorized as medium-sized datasets and are labelled allowing us to test that the synthesis approach preserves class data patterns. Two of the datasets have imbalanced classes, which is a common occurrence in real-world domains. The properties of the datasets are summarized in Table 1.

4.2 Data Pre-Processing

We performed the data preparation steps: Missing values occur in the Adult Census dataset and imputation was done using mode substitution for both SMOTE and WCGAN-GP. In order to generate good quality synthetic data, the input

[5] https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset
[6] https://www.kaggle.com/sulianova/cardiovascular-disease-dataset
[7] https://www.kaggle.com/uciml/adult-census-income

Table 1. Properties of Datasets used in experiments.

Dataset	# Rows	# Features	Categorical Features	Numerical Features	# Classes	% Balance
Credit Card	30,000	23	9	14	2	22 : 78
Cardiovascular	70,000	11	6	5	2	50 : 50
Adult Census	31,562	14	9	5	2	24 : 76

data to the GANs needs to be in an appropriate representation. Each categorical variables is label-encoded to convert to numerical format for both SMOTE and WCGAN-GP. The continuous variables (including the label-encoded categorical variables) were then standardized to bring all the variables into the same range [5]. This results in faster convergence, better processing and ease of reproducibility. It also ensures that each feature gets an equal importance and avoid any biases due to the scale of any specific attribute. Standardization was a crucial pre-processing step with GANs, but with SMOTE, it was not a necessity to standardize or transform the numerical data as the results were not impacted with this transformation.

4.3 Experiment Design

WCGAN-GP: To design the WCGAN-GP model, we used hyperparameter tuning guidelines from GAN [13], WGAN [2] and WGAN-GP [14] sources, all of whom have provided recommendations and guidance on parameter settings that have been proved to be successful in many tasks. The implementation of experiments was carried out using Python 3.7 and specifically, Keras and TensorFlow. For all datasets, the same WCGAN-GP architecture was implemented.

Network Architecture: The depth of generator and critic was set to 3. For generator, the size of nodes in hidden layer were ordered in an ascending size, that is d, d*2, d*4 (where d is 128). The critic had the same hidden nodes but ordered in a descending size [26]. As the input was not an image, the two neural networks did not require convolution layers and thus, were built using dense layers. Leaky ReLU was used as the activation function for each layer except the output layer which used linear activation [28]. The use of dropout in generator was done to minimize the over-fitting. The batch size was 64, learning rate was 0.0001 and Adam Optimizer was used to minimize the loss function. The momentum term $\beta 1$ and $\beta 2$ were set as 0.5 and 0.9 respectively [14]. The model was trained for 5000 epochs as over-training had started to deteriorate the quality of synthetic data. The random noise vector had a length of 32. Once the training was completed, the synthetic data was generated with an exact size as of real data. The list of settings for WCGAN-GP model is outlined in Table 2.

Data Generation: Once the model had been trained, the trained generator was used to produce synthetic data samples.

Reverse Transformation: As the synthetic data generated using WCGAN-GP was in a standardized range because of the initial transformations applied on the

Table 2. Implemented Critic and Generator Model Configurations for WCGAN-GP.

critic C	generator G
Input - Dimension of real data	Input - Random Noise: 32
512, Leaky RELU (alpha: 0.2)	128, Leaky RELU (alpha: 0.2) , Dropout (0.3)
256, Leaky RELU (alpha: 0.2)	256, Leaky RELU (alpha: 0.2) , Dropout (0.3)
128, Leaky RELU (alpha: 0.2)	512, Leaky RELU (alpha: 0.2) , Dropout (0.3)
Output - 1, Linear activation	Output - Dimension of real data, Linear activation
Other Parameters: Learning rate: 0.0001; Adam Optimizer; Batch size: 64; Epochs: 5000	

input data, the synthetic data needed to be reverse transformed (with respect to the initial transformations) to ensure that the synthetic data looked like the real data. The inverse transformations were specific to the initial transformations performed for each variable and were done after the data was generated.

SMOTE: SMOTE was chosen as the comparative approach due to its popularity and common usage. SMOTE does not require any parameter optimizations. As there is no need to build or train any model, the synthetic data can be generated instantly.

SMOTE was used to generate data using the following method. After preprocessing, the original data (with n instances) was replicated to create copies of the dataset and made imbalanced in a ratio of 2 to 1. A new target label was assigned with label as 1 for majority class (2*n instances) and 0 for minority (n instances). SMOTE was run to generate synthetic data samples using the imbalanced-learn library. This generated new synthetic samples with n new instances. As a final step, the new samples were extracted to form a synthetic dataset with the exact size of real data.

4.4 Evaluation Metrics

The metrics we used to determine the similarity of the synthetic datasets to the original datasets, and their preservation of privacy are as follows:

Visual Evaluation (Utility Metric): Univariate analysis was performed to observe the Box and Whisker plots for the numerical and histogram distributions for categorical variables. Further, bi-variate analysis was done to compare the scatterplots between variables in synthetic against variables in real data. These visualisations helped to affirm whether the relationships were preserved in the synthetic data and indicated any existence of mode collapse. Finally, the correlations between the columns of each dataset were also assessed using heatmaps.

Classification Performance (Utility Metric): The synthetic data is a good representation of the real labelled dataset if it performs in the same way as the real data does when used to create and test a machine learning model [15]. This approach involves comparing the performance of a machine learning model trained and tested on real (TRTR) and synthetic data (TSTS) [16].

Decision Tree, Random Forest, Support Vector Machine, and Adaboost were selected as the classifiers because of their common usage and not for any specific performance on the datasets. XGBoost was chosen as it has gained popularity in many machine learning competitions for its speed & performance [24].

The real and synthetic data were split in a 5-fold cross validation. There were three different training-testing settings performed. Setting A REAL: train the predictive models on real training data, test the performance of trained model on real test set. Setting B SMOTE: For the synthetic data generated by SMOTE, train on the generated synthetic train data and test on synthetic test data. Setting C WCGAN-GP: For the synthetic data generated by WCGAN-GP, train on the generated synthetic train data and test on synthetic test data. For the evaluation metric, F1 score (harmonic mean of precision and recall) was recorded as it is one of the widely used metrics to evaluate classification models.

Euclidean Distance to The Nearest Record (Privacy Metric): Euclidean distance was used to evaluate the disclosure risk as it can offer perspective on the similarity of the records between datasets [20]. Euclidean distance to nearest record (d) is the mean distance between synthetic sample and its closest record in original data. A record with zero distance would imply leakage of information and low privacy. The desired outcome is a high mean and low standard deviation. Although Euclidean distance was a metric used for privacy, it was only an indicator of the level of privacy and didn't provide any guarantees at individual row level.

Duplicate records between Real and Synthetic (Privacy Metric): It checked if there were any duplicates between samples in synthetic and real data.

5 Experimental Results

In this section, the results are presented and it is shown that WCGAN-GP showed a better balance between privacy and data utility. Across all datasets, WCGAN-GP performed on par or better than SMOTE on utility and privacy metrics. Note that only relevant visuals and results are presented in this section.

5.1 Utility Metric: Visual Evaluation

Box-Plots: Across all datasets, the numerical samples synthesized from SMOTE and WCGAN-GP had a similar distribution as compared with the real data. Both the approaches were able to capture the basic properties and had a similar range, median, IQR and so on. However, WCGAN-GP model generated few additional out-of-range values (or outliers) in the synthetic data. For instance, the results for credit card dataset are shown in Fig. 2. 'Age' has more outliers towards both the extremes and also contains negative values. This suggests the need for synthetic data treatment after the data is generated using WCGAN-GP to ensure that the data makes sense from logical and business perspective.

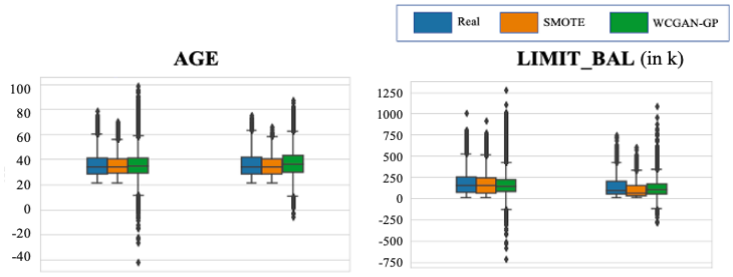

Fig. 2. Box-plots for the numerical variables in Credit Card Dataset. Blue indicates the real data points, orange synthetic from SMOTE and green synthetic from WCGAN-GP.

Histograms: Both SMOTE and WCGAN-GP were able to approximately capture the frequency distributions of categorical variables, but with exceptions. The class imbalance problem was reproduced and remained intact in synthetic datasets. For both approaches, the distributions were nearly similar in most cases when compared with the real data. However, the frequency distributions were not a perfect replica and did not exactly match for either approach. For instance, Fig. 3 shows that the frequencies for 'Workclass', 'Gender', and 'Marital status' in Adult Census data do differ by a certain magnitude, when compared against the real data. It is observed that WCGAN-GP had a hard time capturing the distributions as compared to SMOTE for categories with multi-levels. Similar trend was reproduced in all the three datasets.

Fig. 3. Frequency distributions of categorical variables in Adult Census. Blue indicates the real data, orange synthetic from SMOTE and green synthetic from WCGAN-GP.

Scatterplots: It can be seen from Fig. 4 that the generated data using SMOTE and WCGAN-GP seemed to establish and maintain the relationships between the variables. This pattern was repeated in all the three datasets. It is also inferred that WCGAN-GP had not suffered from mode collapse, which is a common training problem. The samples produced were diverse enough and the model was able to learn and reproduce the distributions of the real-world data.

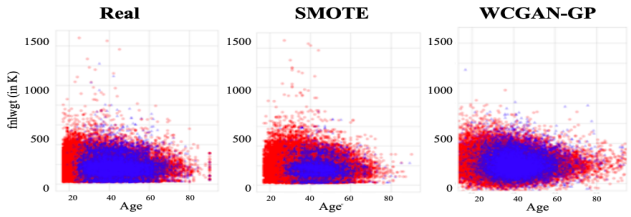

Fig. 4. Scatterplot for 'Age' vs. 'fnlwgt' variable in Adult Census Dataset. Blue indicates records with income >50K, red indicates records with income <=50K.

Correlation Matrix: The results of correlation matrix between the columns of each dataset are presented in Fig. 5. It is observed that the column correlations in synthetic datasets were nearly similar to the original data correlations.

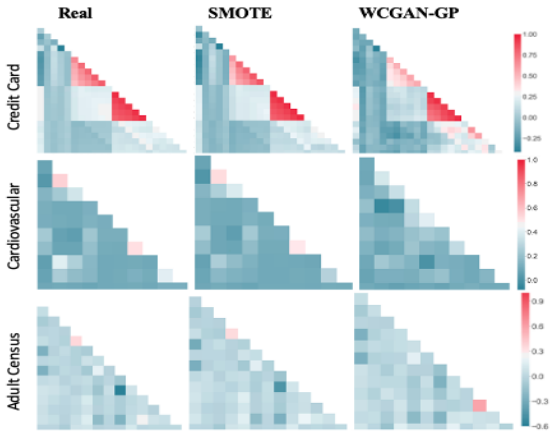

Fig. 5. Correlation Matrix using the real data and synthetically generated datasets.

5.2 Utility Metric: Classification Performance

As shown in Table 3, WCGAN-GP performed at par or better than SMOTE in machine learning tasks. Across datasets, F1 scores of the predictive model built on WCGAN-GP's synthetic data was comparable to the model built on real data. On the contrary, F1 scores of model on SMOTE's synthetic data were far-off from the model on real data, with the exception of cardiovascular dataset.

The quality of synthetic data by WCGAN-GP for data usability was impacted by the presence of data-types in a data. Synthetic data for Credit Card (had more numerical than categorical variables) provided comparable machine learning performance with real data. On the contrary, the F1 scores of models on synthetic data for Cardiovascular or Adult Census data (having more categorical than numerical variables) were significantly different with the F1 scores of models on real data. The sub-optimal performance in these datasets could be due to presence of more categorical variables and the gaps noted between the frequency

distributions of categorical columns of synthetic data (from WCGAN-GP) and real data. Overall, WCGAN-GP fared better than SMOTE in classification tasks as the results for WCGAN-GP data were closer to the results of real data.

Table 3. F1 Scores of predictive models for Real data, Synthetic data from SMOTE, and from WCGAN-GP on different datasets.

Classifier	Credit Card			Cardiovascular			Adult Census		
	Real	SMOTE	GAN	Real	SMOTE	GAN	Real	SMOTE	GAN
Decision Tree	40%	32%	39%	63%	63%	57%	49%	60%	53%
Random Forest	47%	34%	41%	71%	70%	65%	53%	69%	62%
XgBoost	47%	23%	42%	72%	67%	67%	50%	64%	61%
AdaBoost	44%	26%	42%	71%	64%	66%	53%	64%	62%
Linear SVM	13%	2%	20%	28%	21%	27%	19%	49%	35%
Average	38%	23%	37%	61%	57%	56%	45%	61%	55%

5.3 Privacy Metric: Euclidean Distance to The Nearest Record

Table 4 shows that synthetic data using WCGAN-GP consistently had a higher Euclidean distance as compared to the data generated using SMOTE. WCGAN-GP achieves a better privacy-preservation performance than SMOTE as it learns the real data distributions using neural networks and generates a privacy-preserving version of the real dataset that excludes the sensitive information.

Table 4. Euclidean Distance, the value indicates distances (mean, standard deviation).

Dataset	SMOTE	WCGAN-GP
Credit Card	1.18 ± 0.95	3.07 ± 1.24
Cardiovascular	0.37 ± 0.38	0.81 ± 1.37
Adult Census	1.45 ± 0.57	2.59 ± 0.50

5.4 Privacy Metric: Duplicate Records with WCGAN-GP

There were no duplicate records between real and synthetic datasets generated using WCGAN-GP. Whilst this is a somewhat blunt metric, it does strike out the possibility of copied records in synthetic data. There were less than 0.1 percent of identical matches found with SMOTE.

6 Conclusion

The main objective was to investigate the efficacy of WCGAN-GP for generation of tabular datasets with mixed data types, whilst preserving the patterns and privacy of the seeding datasets. The results showed that WCGAN-GP offer a promising framework to generate continuous and categorical data as the synthetic data showed preservation of patterns, distributions and relationships of the real dataset. The synthetic data from WCGAN-GP showed comparable

performance with real data in the classification tasks, substantially better than SMOTE. The synthetic data from WCGAN-GP also offered a better privacy protection than SMOTE.

A potential direction for future research is to combine WCGAN-GP with privacy preserving mechanisms like differential privacy to provide formal and stronger privacy guarantees. Further, more nuanced metrics can be produced and applied in future work to ensure compliance with data protection regulations. Another interesting avenue for future work would be the use softmax or Gumbel softmax to improve the quality of datasets with categorical data types. Finally, evaluation could be expanded to including unsupervised learning tasks like clustering and segmentation to showcase the generalizability and benefits of WCGAN-GP for synthetic data generation.

References

1. Arjovsky, M., Bottou, L.: Towards principled methods for training generative adversarial networks. arXiv preprint arXiv:1701.04862 (2017)
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)
3. Ba, H.: Improving detection of credit card fraudulent transactions using generative adversarial networks. arXiv preprint arXiv:1907.03355 (2019)
4. Bellovin, S.M., Dutta, P.K., Reitinger, N.: Privacy and synthetic datasets. Stan. Tech. L. Rev. **22**, 1 (2019)
5. Brenninkmeijer, B., de Vries, A., Marchiori, E., Hille, Y.: On the generation and evaluation of tabular data using gans (2019)
6. Cao, Y., Liu, B., Long, M., Wang, J.: Hashgan: Deep learning to hash with pair conditional wasserstein gan. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1287–1296 (2018)
7. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research **16**, 321–357 (2002)
8. Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W.F., Sun, J.: Generating multi-label discrete patient records using generative adversarial networks. arXiv preprint arXiv:1703.06490 (2017)
9. Drechsler, J., Reiter, J.P.: An empirical evaluation of easily implemented, non-parametric methods for generating synthetic datasets. Computational Statistics & Data Analysis **55**(12), 3232–3243 (2011)
10. Esteban, C., Hyland, S.L., Rätsch, G.: Real-valued (medical) time series generation with recurrent conditional gans. arXiv preprint arXiv:1706.02633 (2017)
11. Garcia Torres, D.: Generation of synthetic data with generative adversarial networks (2018)
12. Goncalves, A., Ray, P., Soper, B., Stevens, J., Coyle, L., Sales, A.P.: Generation and evaluation of synthetic patient data. BMC Medical Research Methodology **20**, 1–40 (2020)
13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)

14. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: Advances in neural information processing systems. pp. 5767–5777 (2017)
15. Heyburn, R., Bond, R., Black, M., Mulvenna, M., Wallace, J., Rankin, D., Cleland, B.: Machine learning using synthetic and real data: Similarity of evaluation metrics for different healthcare datasets and for different algorithms. In: Proc. 13th Int. FLINS Conf.(FLINS2018). World Scientific (2018)
16. Jordon, J., Yoon, J., van der Schaar, M.: Measuring the quality of synthetic data for use in competitions. arXiv preprint arXiv:1806.11345 (2018)
17. Kaloskampis, I., Pugh, D., Joshi, C., Nolan, L.: Synthetic data for public good — data science campus. https://datasciencecampus.ons.gov.uk/projects/synthetic-data-for-public-good/ (2020), (Accessed on 10/11/2020)
18. Lu, P.H., Wang, P.C., Yu, C.M.: Empirical evaluation on synthetic data generation with generative adversarial network. In: Proceedings of the 9th International Conference on Web Intelligence, Mining and Semantics. pp. 1–6 (2019)
19. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arxiv 2014. arXiv preprint arXiv:1411.1784 (2014)
20. Park, N., Mohammadi, M., Gorde, K., Jajodia, S., Park, H., Kim, Y.: Data synthesis based on generative adversarial networks. arXiv preprint arXiv:1806.03384 (2018)
21. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
22. Sagong, M.C., Shin, Y.G., Yeo, Y.J., Park, S., Ko, S.J.: cgans with conditional convolution layer. arXiv preprint arXiv:1906.00709 (2019)
23. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in neural information processing systems. pp. 2234–2242 (2016)
24. Sandulescu, V., Chiru, M.: Predicting the future relevance of research institutions-the winning solution of the kdd cup 2016. arXiv preprint arXiv:1609.02728 (2016)
25. Surendra, H., Mohan, H.: A review of synthetic data generation methods for privacy preserving data publishing. Int J Sci Technol Res **6**(3), 95–101 (2017)
26. Tanaka, F.H.K.d.S., Aranha, C.: Data augmentation using gans. arXiv preprint arXiv:1904.09135 (2019)
27. Vega-Márquez, B., Rubio-Escudero, C., Riquelme, J.C., Nepomuceno-Chamorro, I.: Creation of synthetic data with conditional generative adversarial networks. In: International Workshop on Soft Computing Models in Industrial and Environmental Applications. pp. 231–240. Springer (2019)
28. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853 (2015)
29. Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling tabular data using conditional gan. In: Advances in Neural Information Processing Systems. pp. 7335–7345 (2019)
30. Yoon, J., Drumright, L.N., Van Der Schaar, M.: Anonymization through data synthesis using generative adversarial networks (ads-gan). IEEE Journal of Biomedical and Health Informatics (2020)