Generation of synthetic data with Conditional Generative Adversarial Networks

BELÉN VEGA-MÁRQUEZ[*], *Department of Computer Languages and Systems, University of Sevilla, 41012, Sevilla, Spain.*

CRISTINA RUBIO-ESCUDERO, *Department of Computer Languages and Systems, University of Sevilla, 41012, Sevilla, Spain.*

ISABEL NEPOMUCENO-CHAMORRO, *Department of Computer Languages and Systems, University of Sevilla, 41012, Sevilla, Spain.*

Abstract

The generation of synthetic data is becoming a fundamental task in the daily life of any organization due to the new protection data laws that are emerging. Because of the rise in the use of Artificial Intelligence, one of the most recent proposals to address this problem is the use of Generative Adversarial Networks (GANs). These types of networks have demonstrated a great capacity to create synthetic data with very good performance. The goal of synthetic data generation is to create data that will perform similarly to the original dataset for many analysis tasks, such as classification. The problem of GANs is that in a classification problem, GANs do not take class labels into account when generating new data, it is treated as any other attribute. This research work has focused on the creation of new synthetic data from datasets with different characteristics with a Conditional Generative Adversarial Network (CGAN). CGANs are an extension of GANs where the class label is taken into account when the new data is generated. The performance of our results has been measured in two different ways: firstly, by comparing the results obtained with classification algorithms, both in the original datasets and in the data generated; secondly, by checking that the correlation between the original data and those generated is minimal.

Keywords: Synthetic Data, Conditional Generative Adversarial Networks, Deep Learning

1 Introduction

The introduction of the new data protection law [10] has supposed that the process of sharing personal data has become increasingly tough and difficult, especially in the medical field, where data is very sensitive. Because of this, scientists and doctors have to establish agreements between themselves before sharing any personal data. These requirements slow down or even prevent the exchange of data among researchers [4].

To face this problem, several solutions have been contemplated, seeking for finding or simulating data that are similar to the real one without involving sensitive information. Among these solutions, the use of Deep Learning techniques to generate synthetic data similar to the real one stands out [6, 29]. The purpose of this synthetic data is to be used to train machine learning models that can then be used on real data, so that the training is done without having to make public the real data.

[*] E-mail: bvega@us.es

The precision of this technique is measured by comparing the results obtained with real data and synthetic data, so that they are as similar as possible.

Generative adversarial networks (GANs) [11] and its variants have attracted many researchers in their research work due to its elegant theoretical basis and its great performance in the generation of synthetic data from real data [28], such as generating clinical data on blood pressure [4] or even generating new magnetic resonance images for segmentation tasks [22]. GANs are a type of Deep Learning model in which two networks are trained against each other in a zero-sum game framework. Commonly one network is known as Generator an the other as Discriminator [26].

The purpose of this article is to evaluate the utility of the samples generated by an adversarial neural network from five different datasets. To work with these sets of data, we have used a Conditional Generative Adversarial Neural network [16], that takes into account the class to which the instances belong. We have based the evaluation of our proposal on two different methods: the first method is to measure the correlation between the real data and the synthetic. As mentioned above, the objective of the use of these techniques is the privacy of the data, so it is advisable that the transformation process is unidirectional so that real data can not be obtained from false data. Pearson's correlation index, covariance and Spearman's index will measure this phenomenon, so that a low value will be optimal, meaning that the two sets of data are not correlated and cannot be inferred from each other. The second method is to compare the accuracy obtained with a classification algorithm, specifically the XGBoost for the two sets of data, i.e. real and generated data. If this accuracy is similar, it means that the model trained with the false set serves to reach conclusions about the real set without having to use it for training.

The article is organised as follows: Section 2 provides a detailed description about the methodology used in all the process. Section 3 shows the results obtained with the previous techniques previously described, finally, section 4 shows the conclusions that have been obtained after the research.

2 Methodology

Our aim in this study is to provide a Deep Learning approach to simulate new data from different datasets of different topics. We used a type of GAN known as Conditional Generative Adversarial Network (CGAN) which is the key technique in our approach. This is because this type of networks shows very good results in datasets that have a target class, since they take into account this data configuration to train the neural network. The new data fits as closely as possible to the data according to which class each of the instances belongs to [20].

2.1 Generative Adversarial Networks

GANs are a Deep Learning model which comprises two different neural networks, a generator and a discriminator which are simultaneously trained competitively, in a zero-sum game framework.

Figure 1 shows the basic architecture of a GAN network. The generative network (G) is in charge of learning how to assign elements of a latent space *n* (noise) to a certain data distribution, i.e. what it does is to generate new data G(n) that is as close as possible to the real data (x). On the other hand, the functionality of the discriminator (D) consists in differentiating between elements of the original distribution, *x* and those created by the generative network by calculating the probability of belonging to one set or another [11]. To summarise, the discriminator network is a standard convolutional network that can categorise the examples fed to it, a binomial classifier labelling instances as real or fake. The generator is an inverse convolutional network, in a sense, while a standard convolutional

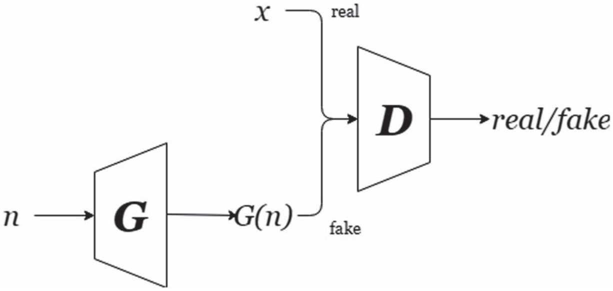

FIGURE 1. GAN network architecture. *n* represents random noise that serves as input to the generator network (*G*), while *x* exemplifies real data which is used, in conjunction with fake data *G(n)* to train the discriminator *D*.

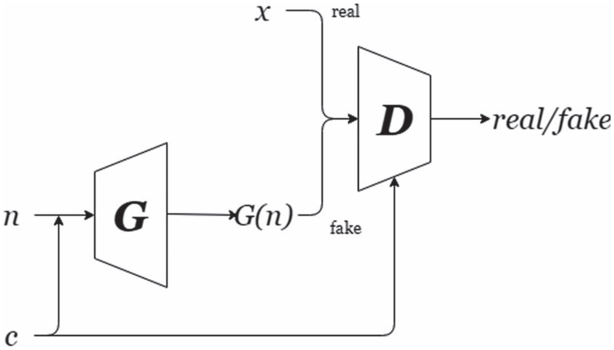

FIGURE 2. CGAN network architecture. *n* represents random noise that serves as input to the generator network (*G*), while *x* exemplifies real data which is used, in conjunction with fake data *G(n)* to train the discriminator *D*. *c* is the class taken into account in order to generate better instances in comparison with original GAN architecture.

classifier takes an example and downsamples it to produce a probability, the generator takes a vector of random noise and upsamples it to an instance. The first throws away data through downsampling techniques like max pooling, and the second generates new data.

2.2 Conditional Generative Adversarial Networks

A GAN does not take into account any type of condition with respect to the data. Usually the synthetic data to be generated has a type of property that distinguishes it, which must also be used to obtain synthetic data as close as possible to the real ones.

After this approach, CGANs arised. CGANs are an extension of GAN where some condition is taken into account. This condition implies that both the discriminator and the generator have to take into account some additional information, let's call it *c*, where *c* it can be any type of additional information, such as data from another nature or some class label [16].

Figure 2 illustrates the basic architecture of a CGAN. It can be observed that the structure is very similar to the typical adversarial neural networks; however, there is one more factor to take into account, and that is the class *c* to which the instance belongs.

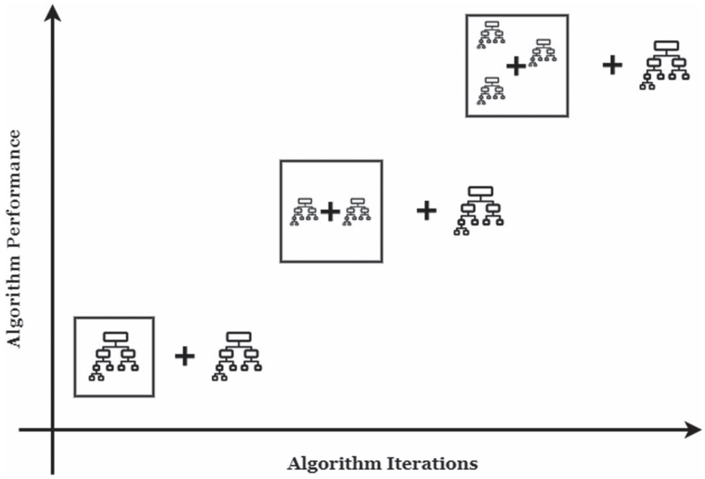

FIGURE 3. A summary of gradient boosting trees methodology.

2.3 Classification Task

The main objective of our work is to generate fake data that is as similar as possible to the real data. In order to measure the performance of our proposal, we have decided to check whether the behaviour of classifying the generated data is similar to the behaviour of real data.

For the classification task we chose a Tree Gradient Boosting algorithm, particularly the Extreme Gradient Boosting (XGBoost) implementation [5] which has demonstrated to have a really good performance in classification tasks [12]. Tree gradient boosting is an ensemble algorithm which converts several weak classifiers (in this case trees) into a strong classifier so as to improve global classification performance [21]. As can we seen in Figure 3, a gradient boosting is an additive model where a tree is added and combined with other trees in each iteration to reduce the loss.

2.4 Datasets

In an attempt to ensure that our methodology fits well with any type of data, we have decided to create synthetic data from five different types of datasets which are described below. These type of data has been widely used in classification related studies [3, 15, 19]

- **Default of Credit Card Clients Dataset** [14]: The dataset contains information about default payments, demographic factors, credit data, history of payment and bill statements of credit card clients in Taiwan from April 2005 to September 2005. The label class is called *Default payment* and it indicates if the next month the payment will be carried out or not (1 or 0).
- **Tokyo Earthquake Dataset** [2, 3]: This dataset collects demographic data from earthquakes in Tokyo between 2013 and 2015. Each one of the instances is made up of seismic indicators and a label, the label for every event is defined as a logical value (1 or 0): one, if the maximum magnitude for events is larger than a predefined threshold (in this case 5); or zero, otherwise.

TABLE 1. Description of the datasets.

Dataset	# of instances	# of attributes	Percentage of each class
Credit Card	30000	20	80/20
Earthquake	994	94	75/25
Electricity	10000	13	65/35
Diabetes	768	8	65/35
Wine	6497	11	95/5

- **Simulated Electrical Grid Stability Dataset** [1]: one of the milestones of electricity companies is to predict electricity demand from variables such as price for stable electricity grids. One of the newest techniques is the Decentral Smart Grid Control (DSGC) based on mathematical equations with input values. This dataset collects a simulation of these input values and indicates through a label whether the power grid will be stable or not.
- **Pima Indians Diabetes Dataset** [25]: The objective of the dataset is to predict whether or not a patient has diabetes, based on medical diagnostic measurements. The dataset consists on several medical predictor variables and one target variable named *Outcome*.
- **Wine Quality Dataset** [8]: Data related to red and white variants of the Portuguese ”Vinho Verde” wine. This dataset contains chemical data that defines the quality of the sample using a numerical scale. For this study, a threshold of five has been chosen to define whether the wine is of quality or not, thus having a binary label, i.e. one means high quality, zero means poor quality.

Finally, Table 1 summarizes the number of features and instances together with the percentage of each class. We can observe that there is a wide variety of data distributions: balanced and unbalanced datasets, number of instances from 768 to 30000 and number of attributes from 8 to 94.

2.5 Software and experimental setting

The CGAN network used in this study has been implemented with the Keras library [7]. Keras is a high-level neural networks API, written in Python and capable of running on top of Tensorflow. The classification of the data has been carried out with the scikit-learn library [18]. The executions were performed on an Intel machine, specifically Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz, with 64 GB of RAM and 12 cores.

3 Results

3.1 Generating New Data with CGANs

First, in order to obtain better results, a preprocessing step was performed. This preprocessing consists of an standard normalization due to the fact that Deep Learning models work better with normalized data and and without null values [17].

To apply CGAN architecture to the datasets the GAN-Sandbox [9] package was used. GAN-Sandbox has a number of popular GAN architectures implemented in Python using the Keras library and a Tensorflow backend. All the results obtained are available as Jupyter Notebooks in [27].

TABLE 2. XGBoost Classification results in terms of Accuracy, F1-Score and Area Under the curve.

		Accuracy	F1-Score	AUC
Credit Card	*Real*	0.8204	0.4706	0.7789
	Fake	0.8519	0.6129	0.8801
	Real+Fake	0.8304	0.5121	0.8258
Earthquake	*Real*	0.9637	0.9254	0.9810
	Fake	0.9889	0.9780	0.9996
	Real+Fake	0.9616	0.9231	0.9887
Electricity	*Real*	0.9998	0.9998	0.9999
	Fake	0.9748	0.9803	0.9961
	Real+Fake	0.9866	0.9895	0.9985
Diabetes	*Real*	0.7656	0.6459	0.8294
	Fake	0.8268	0.7368	0.8950
	Real+Fake	0.7857	0.6738	0.8441
Wine	*Real*	0.9602	0.9797	0.7866
	Fake	0.9627	0.9809	0.8598
	Real+Fake	0.96167	0.9804	0.8339

As mentioned above, the neural network is composed of two networks, the discriminator and the generator which have the following structure:

1. Generator Network:
   - 1 Input layer: the input layer receives noise data and the class of the real data.
   - 4 Dense layers with the parameters specified below:

     - First Dense layer: 30 neurons and rectified linear activation function.
     - Second Dense layer: 60 neurons and rectified linear activation function.
     - Third Dense layer: 120 neurons and rectified linear activation function.
     - Fourth Dense layer: as many neurons as number of columns in the real dataset.
2. Discriminator Network:
   - 1 Input layer: the input layer receives the fake data generated by the generator network and the real data.
   - 4 Dense layers with the parameters specified below:
   - First Dense layer with 120 neurons and rectified linear activation function.
   - Second Dense layer with 60 neurons and rectified linear activation function.
   - Third Dense layer with 30 neurons and rectified linear activation function.
   - 1 Dense layer with 1 neuron and sigmoid activation function in charge of deciding whether the entry instance is real or not.

After CGAN training with 50000 epochs, new instances were generated in order to have the same number of real and fake examples. Figure 4 shows scatter plots of the first two variables of each dataset. Each row of the image represents one dataset, to the right of the image we can see the graphs corresponding to the new data generated, differentiating by colours the classes to which they belong. In the left column we see the graphs corresponding to the real data. It can be seen that a

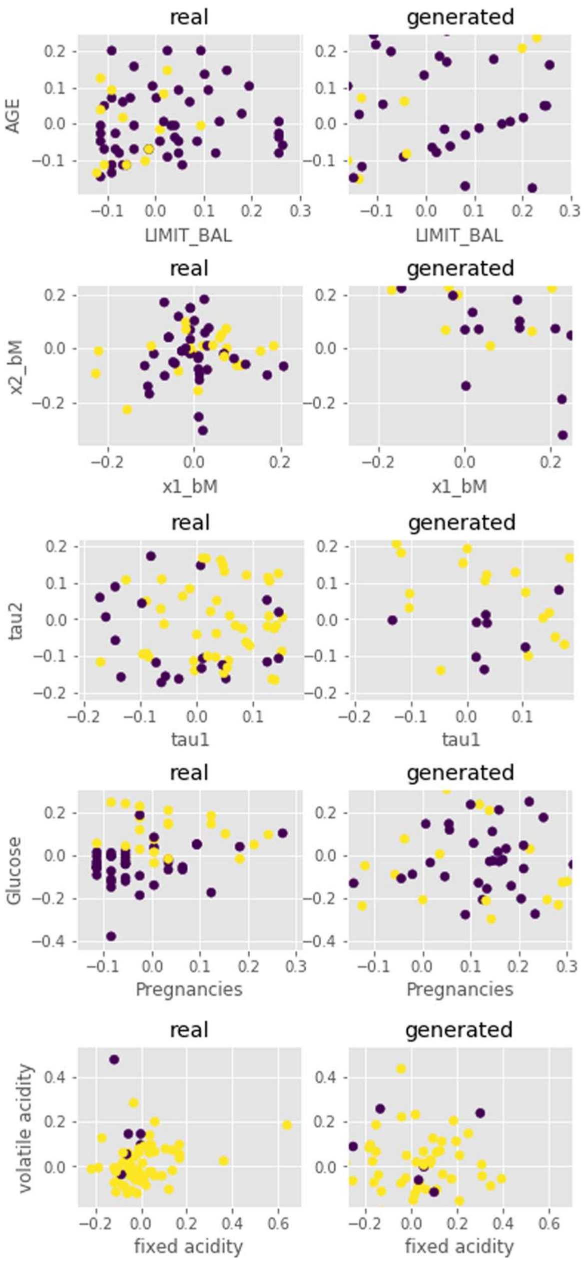

FIGURE 4. Scatterplot of some attributes of the datasets for real and generated data.

priori, the data obtained is not similar to real data, which was our goal, since one of the objectives of this work is to obtain false data that behave in the same way as real data in classification tasks, but without being able to establish a relationship between these two sets of data.

TABLE 3. XGBoost Classification Results for Subsampled Credit Card Data in terms of Accuracy, F1-Score and Area Under the Curve.

		Accuracy	F1-Score	AUC
Subsampled Credit Card	*Real*	0.7134	0.6877	0.7774
	Fake	0.7943	0.7893	0.8715
	Real+Fake	0.7439	0.7269	0.8177

3.2 Classification Results

One of the procedures we followed to check whether or not our method was appropriate was to ascertain that the results in classification tasks were similar for the real and generated data, as well as for the two sets of data joined together. The XGBoost algorithm was used to perform this task. XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. The metrics used to check the performance of our approach are Accuracy, F1-Score and Area Under the Curve (AUC). It can be seen in Table 2 that the results obtained are practically equal for both separate and joined data. These results show that the generated data can be used indistinctly for the same tasks, training in our particular case, as the real ones obtaining the same results.

Although the results have been very good for most cases, in the case of F1-Score for credit card data, although they are still similar, they are quite low. One of the causes of this phenomenon may be the lack of balance in the classes of data. We perform the same experimentation but with balanced data, specifically by subsampling the dataset, specifically samples have been chosen randomly. Table 3 illustrates the new classification scores with the subsampled Credit Card data.

3.3 Similarity of the Data

The last procedure we followed to check whether our method was appropriate was to measure the relationship between the variables in the original dataset and the new dataset generated. The objective was to obtain values that indicated that the correlation between these sets was null or minimal in order to be able to use this technique in controversial fields such as medicine or finances.

In this section we have calculated three correlation measures to find out whether the variables in the real and false datasets are correlated or not. These measures are Pearson's correlation coefficient, covariance and Spearman's correlation index.

Pearson's Correlation Coefficient [24] is a measure of the linear correlation between two variables, X and Y. The value has a range between -1 and 1, where 1 is a total positive linear correlation, i.e., when X increases, Y too. In the case of -1 value, it indicates there is a total negative linear correlation (when X increases, Y decreases or viceversa). Finally, a zero value means there is no correlation between the two variables.

Secondly, covariance [13] is defined as the expected value of variations of two variables from their expected values, i.e., covariance measures how much variables change together. The sign of the covariance can be interpreted as follows: a positive sign means two variables change in the same direction, and a negative sign means they change in different directions. A zero value indicates that both variables are completely independent. As for the magnitude of the value, at an interpretational level, a higher covariance value in absolute value will indicate a stronger linear relationship between the two variables. The disadvantage of this value is that it does not have a maximum or minimum value as it happens with the Pearson's or Spearman's coefficient.

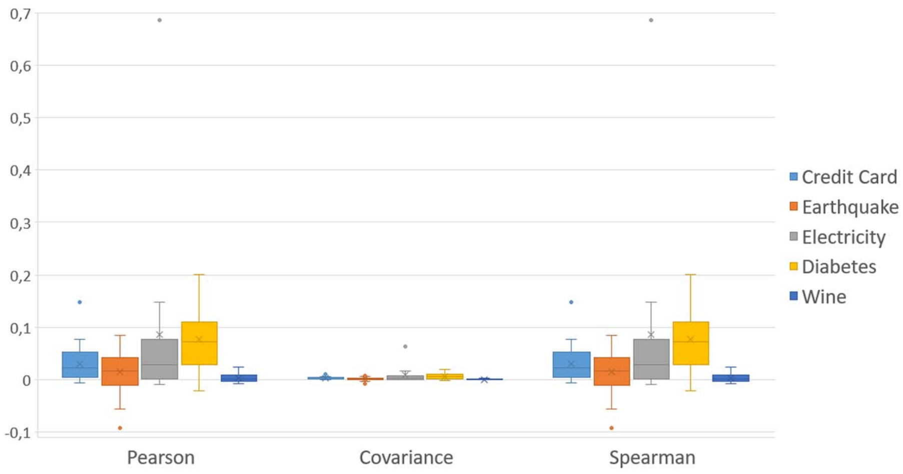

FIGURE 5. Correlation indices by dataset.

Thirdly, Spearman's correlation coefficient [23] is a Pearson correlation coefficient calculated with the ranks of the values of each of the two variables instead of their actual values. It can be used to summarize the strength between the two variables when it is not supposed the two variables are related by linear relationship. As Pearson's correlation coefficient, the measures are between -1 and 1, where -1 indicates a total negative correlation and 1 means perfectly positively correlation.

Finally, Figure 5 shows the Pearson's correlation coefficient, covariance and Spearman's correlation coefficient between real and fake dataset for each of the datasets under study. It can be seen that all the values of the three coefficients are very close to zero or even zero when comparing variables in datasets two by two, indicating that there is no real correlation between the variables of the original dataset and the newly created dataset. In the case of the Electricity dataset, there is only one attribute with a high Pearson and Spearman correlation value, however, it remains a low value as it does not reach the 0.7 value. In general, the measures are near to zero, aspect very important as we mentioned above, in areas such as medicine or finances, where user data are affected by strong privacy needs that limit its use. Moreover, if these data do not present any correlation with the originals but behave in the same way, they can be used indistinctly for any of the tasks in which they are used.

4 Conclusion

In this paper, we proposed a CGAN that generates new synthetic data from training data which can be used indistinctly for the same tasks without having to reveal the actual data. Traditionally, the networks used to generate synthetic data are the so-called GANs; however, we have decided to use a variant in this research in order to obtain the best results. CGANs are based on GANs, the difference is that they take into account a tag or class when generating new data. This characteristic allows the generated data to be more similar to the real ones.

The results obtained have been evaluated in two different ways: first, since the dataset had a label that could be used for classification tasks, the same algorithm, XGBoost, has been tested with the same parameters in the two sets of data. The results have shown that the accuracy of classification is similar in both cases. In addition, it has also been proven that with the union of the two datasets, the results in the metrics used to validate the classification algorithm were similar. Secondly, it has been verified that the correlation between the new data and the original data is minimal, so that they can be used in controversial fields, such as medicine or finances, in which client data must be treated with special care so as to avoid privacy problems.

In conclusion, the research finding of this study have provided some evidence that Deep Learning methods can be used, with good performance, in synthetic data generation. For future work we will consider new variants of adversarial networks to perform this task, as well as the adjustment of parameters to get the most reliable results.

Acknowledgements

We would like to thank Spanish Ministry of Science and Innovation for the financial support under project TIN2017-88209-C2-2-R and Andalusian Government project US-1263341.

References

[1] V. Arzamasov, K. Bohm and P. Jochem. Towards concise models of grid stability. In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, SmartGridComm*. IEEE, 2018.

[2] G. Asencio-Cortés, F. Martínez-Álvarez, A. Morales-Esteban, J. Reyes and A. Troncoso. Using principal component analysis to improve earthquake magnitude prediction in japan. *Logical Journal of the IGPL*, **25**, 949–966, 2017.

[3] G. Asencio-Cortés, F. Martínez-Álvarez, A. Troncoso and A. Morales-Esteban. Medium-large earthquake magnitude prediction in tokyo with artificial neural networks. *Neural Computing and Applications*, **28**, 1043–1055, 2017.

[4] B. K. Beaulieu-Jones, Z. S. Wu, C. Williams, R. Lee, S. P. Bhavnani, J. B. Byrd and C. S. Greene. Privacy-preserving generative deep neural networks support clinical data sharing. *Circulation: Cardiovascular Quality and Outcomes*, **12**, e005122, 2019.

[5] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

[6] E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart and J. Sun. Generating multi-label discrete electronic health records using generative adversarial networks. arXiv preprint arXiv:1703.06490, 2017.

[7] F. Chollet. et al. Keras. https://keras.io, 2015.

[8] P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, **47**, 547–553, 2009.

[9] M. Dietz. Gan-sandbox. https://github.com/mjdietzx/GAN-Sandbox, 2017.

[10] C. Generales. Ley orgánica 3/2018, de 5 de diciembre, de protección de datos personales y garantía de los derechos digitales. https://www.boe.es/buscar/doc.php?id=BOE-A-2018-16673, 2018.

[11] I. J. Goodfellow, J. Pouget-abadie, M. Mirza, B. Xu, D. Warde-farley, S. Ozair, A. Courville and Y. Bengio. GANs. *NIPS*, Advances in neural information processing systems, 2014.

[12] J. L. Lopez Guerra, B. Pontes, A. Moreno, C. Rubio, F. Núõez, I. Nepomuceno, J. Moreno, J. Cacicedo, J. M. Praena-Fernandez, G. A. Escobar Rodriguez, C. Parra, J. Riquelme and M. J. Ortiz-Gordillo. Decision support system for lung cancer patients. In *Radiotherapy and Oncology*, vol. 127, pp. S449–S450, Barcelona, Spain. ESTRO 37, April 20–24, 2018.

[13] H.-Y. Kim. Statistical notes for clinical researchers: covariance and correlation. *Restorative Dentistry & Endodontics*, **43**, e4 (Feb), 2018.

[14] M. Lichman. *UCI Machine Learning Repository*, 2013. https://archive.ics.uci.edu/ml/index.php

[15] F. Martínez-álvarez, A. Troncoso, G. Asencio-Cortés and J. C. Riquelme. A survey on data mining techniques applied to electricity-related time series forecasting. *Energies*, 2015.

[16] M. Mirza and S. Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

[17] G. Montavon, G. Orr and K.-R. Müller. *Neural networks: tricks of the trade*, vol. **7700**. Springer, 2012.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, **12**, 2825–2830, 2011.

[19] R. Pérez-Chacón, J. M. Luna-Romera, A. Troncoso, F. Martínez-Alvarez and J. C. Riquelme. Big data analytics for discovering electricity consumption patterns in smart cities. *Energies*, **11**, 683, 2018.

[20] G. Ramponi, P. Protopapas, M. Brambilla and R. Janssen. T-CGAN: conditional generative adversarial network for data augmentation in noisy time series with irregular sampling. arXiv preprint arXiv:1811.08295, 2018.

[21] X. Ren, H. Guo, S. Li, S. Wang and J. Li. A novel image classification method with cnn-xgboost model. In *Digital Forensics and Watermarking*, C. Kraetzer, Y.-Q. Shi, J. Dittmann and H. J. Kim, eds, pp. 378–390. Springer International Publishing, Cham, 2017.

[22] M. Rezaei, H. Yang and C. Meinel. Multi-task generative adversarial network for handling imbalanced clinical data. arXiv preprint arXiv:1811.10419, 2018.

[23] P. Schober, C. Boer and L. A. Schwarte. Correlation coefficients. *Anesthesia & Analgesia*, **126**, 1763–1768, 2018.

[24] P. Sedgwick. Pearson's correlation coefficient. *BMJ*, **345**, e4483, 2012.

[25] J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler and R. S. Johannes. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. *Annual Symposium on Computer Applications in Medical Care*. American Medical Informatics Association, 1988.

[26] A. Triastcyn and B. Faltings. *Generating differentially private datasets using GANs*, 2018.

[27] B. Vega. Syntheticdata. https://github.com/bvegaus/syntheticData, 2019.

[28] L. Xie, K. Lin, S. Wang, F. Wang and J. Zhou. Differentially private generative adversarial network. arXiv preprint arXiv:1802.06739, 2018.

[29] J. Yoon, J. Jordon and M. van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. *International Conference on Learning Representations*. OpenReview, 2019.