Creation of Synthetic Data with Conditional Generative Adversarial Networks

Belén Vega-Márquez(✉), Cristina Rubio-Escudero, José C. Riquelme, and Isabel Nepomuceno-Chamorro

Department of Computer Languages and Systems, University of Sevilla, Sevilla, Spain
bvega@us.es

Abstract. The generation of synthetic data is becoming a fundamental task in the daily life of any organization due to new protection data laws that are emerging. Generative Adversarial Networks (GANs) 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 new data [19]. 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, they treat it as another attribute. This research work has focused on the creation of new synthetic data from the *"Default of Credit Card Clients"* dataset 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 by comparing the results obtained with classification algorithms, both in the original dataset and in the data generated.

Keywords: Synthetic data, Conditional Generative Adversarial Networks, Deep Learning, Credit Card Fraud Data

1 Introduction

The introduction of the new data protection law [7] has supposed that the process of sharing personal data has become increasingly tough and difficult, especially in the medical field, where data is highly personal and can be used to harm patients themselves. 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 between researchers [1].

Facing with this problem, several solutions have been contemplated that seek to find or simulate data that are similar to the real one without involving individuals. Among these solutions, the use of Deep Learning techniques to generate

synthetic data similar to real ones stands out [4,20]. The purpose of this synthetic data is to be used to train machine learning models that can then be used in the real data, so that the training is done without having to make the real data public. 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) [8] have shown to be one of the most successful techniques in the creation of synthetic data from real data, such as generating clinical data on blood pressure [1] or even generating new magnetic resonance images for segmentation tasks [14]. Generative Adversarial Networks in which two networks are trained against each other in a zero-sum game framework. Commonly one network is known as Generator and the other as Discriminator [17].

The purpose of this article is to evaluate the utility of the samples generated by an adversarial neural network with the Credit Card Fraud Detection Data from Kaggle. To work with this dataset, we have used a Conditional Generative Adversarial Network (CGANs) [11] that takes into account the class to which the instances belong. We considered two methods to evaluate the work: the first method is to measure the correlation between the real data and synthetic data. 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 will measure this phenomenon, so that a low correlation index would 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 [3] for the two sets of 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: Sect. 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, and finally, Sect. 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 based on the Credit Card Fraud Detection Data. 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 data sets that have a target class, since they take into account this detail to train the neural network so that the new data fits as closely as possible to the data according to which class each of the instances belongs to [13].

2.1 Generative Adversarial Networks

Generative Adversarial Networks are a deep learning model which comprise two different neural networks, a generator and a discriminator who are simultaneously trained competitively, as in a zero-sum game framework.

The generative network (G) is in charge of learning how to assign elements of a latent space (noise) to a certain data distribution, i.e., what it does is to generate new data that is as close as possible to the real data. On the other hand, the functionality of the discriminator (D) consists in differentiating between elements of the original distribution and those created by the generative network by calculating the probability of belonging to one set or another [8]. 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 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. Figure 1 shows the basic architecture of a GAN network.

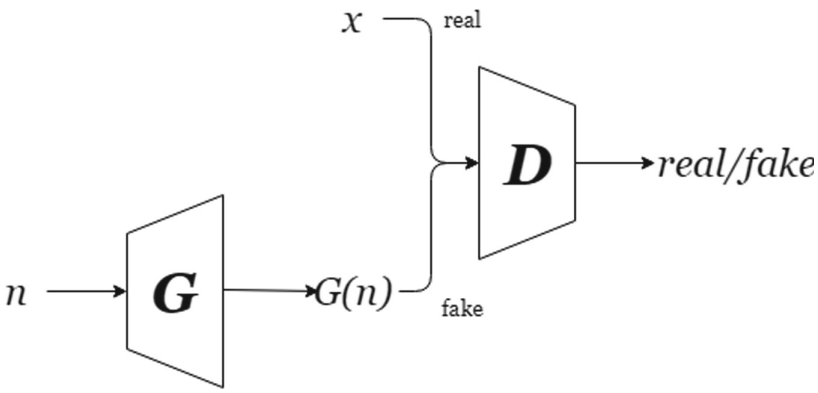

Fig. 1. GAN network architecture

2.2 Conditional Adversarial Networks

Figure 2 illustrates the basic architecture of a conditional adversarial network. 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.

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.

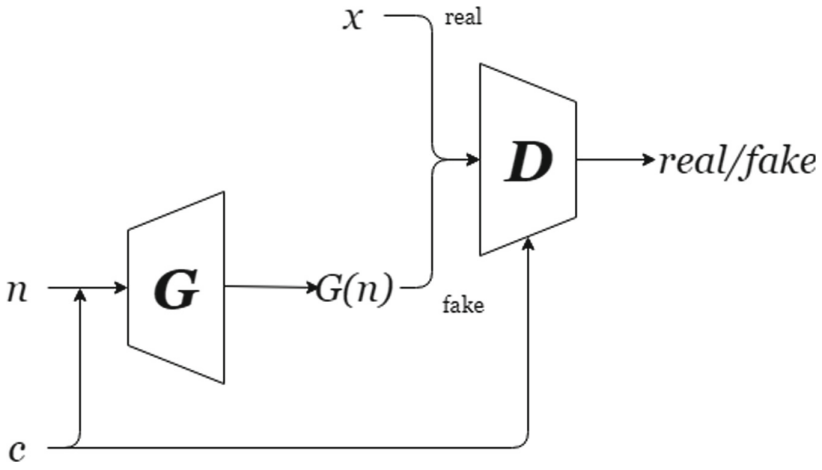

Fig. 2. CGAN network architecture

After this approach, Conditional Adversarial Networks (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 can be any type of additional information, such as data from another nature or some class label [11].

2.3 Dataset

The dataset chosen for this study was *"Default of Credit Card Clients Dataset"* available in [10]. The dataset contains 30000 examples and 25 variables with 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).

2.4 Software and Experimental Setting

The CGAN network used in this study has been implemented with the Keras library [5]. 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 [12]. The executions were performed on an Intel machine, specifically Intel(R) Core(TM) i7-8700 CPU @ 3.20 GHz, with 64 GB of RAM and 12 cores.

3 Results

3.1 Generating New Credit Card Data with CGANs

First, in order to obtain better results, a preprocessing step was performed due to the fact that the data does not fit a normal distribution. This preprocessing consists of a standard normalization.

To apply CGAN architecture to the Credit Card Data Fraud dataset the GAN-Sandbox [6] 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 a Jupyter Notebook in [18].

As mentioned above, the neural network in turn 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 the real data with which the model is going to be trained.
   - 6 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, Fourth and Fifth Dense layer: 120 neurons and rectified linear activation function
     - Sixth Dense layer: 20 neurons which correspond to the number of the columns of the dataset used to train the model
2. Discriminator Network
   - 1 Input layer: the input layer receives the fake data generated by the generator network
   - 1 Dense layer with 120 neurons and rectified linear activation function
   - 1 Dropout layer with a dropout rate of 0.1
   - 1 Dense layer with 60 neurons and rectified linear activation function
   - 1 Dropout layer with a dropout rate of 0.1
   - 1 Dense layer with 30 neurons and rectified linear activation function
   - 1 Dense layer with 1 neuron and sigmoid activation function

After CGAN training with 5000 epochs, 30000 new instances were generated in order to have the same number of real and fake examples. Figure 3 shows FOUR scatterplot of 4 variables of the dataset. To the right of the image are the graphs corresponding to the new data generated, differentiating by colors the classes to which they belong. In the left column you can see the graphs corresponding to the real data. It can be seen that a priori the data obtained is not similar to real data, but it was not a problem due the fact that one of the objectives of this work is to obtain false data that behave in the same way as real ones in classification task, but without being able to establish a relationship between these two sets of data.

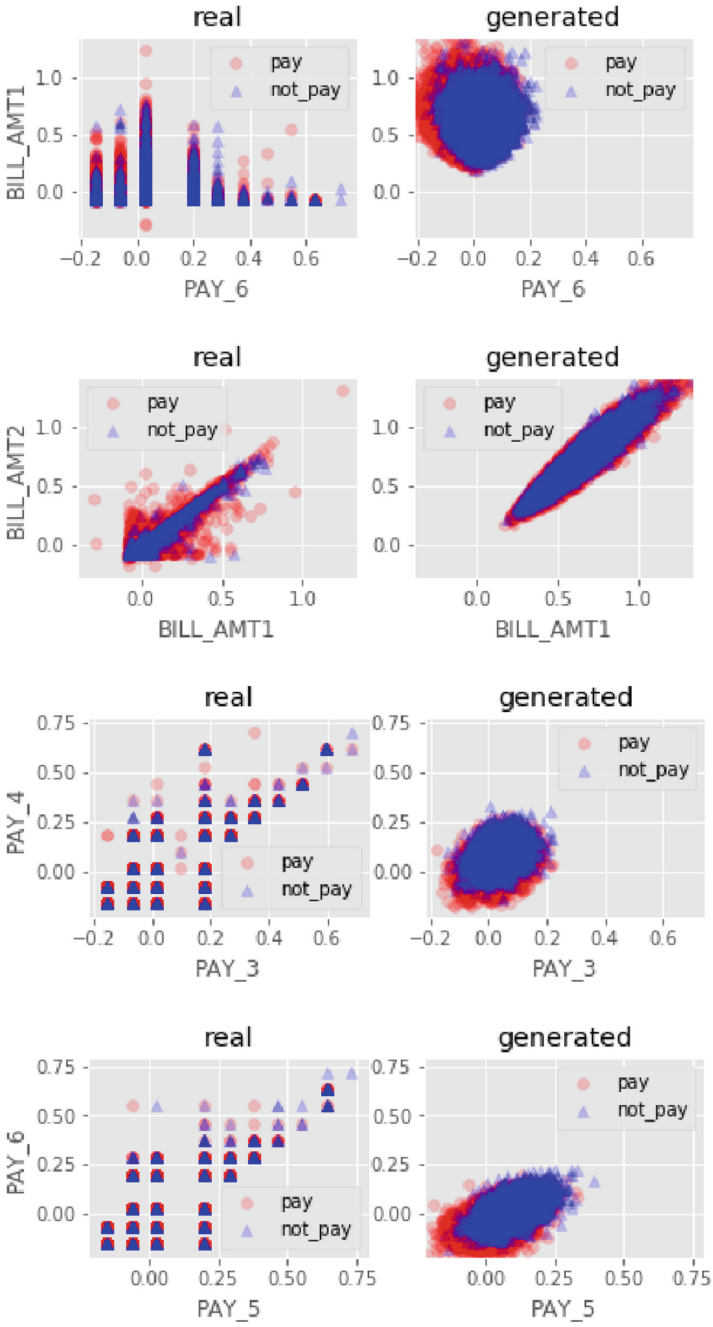

Fig. 3. Scatterplot of some attributes of dataset distinguished by real and generated data

3.2 Similarity of the Data

One of the procedures we followed to check whether or not 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 banks.

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

Pearson's Correlation Coefficient [16] 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, that is, when X increases, Y too, -1 indicates there is a total negative linear correlation (when X increases, Y decreases or vice versa) and finally, a zero value means there is no correlation between the two variables.

Secondly, covariance [9] is defined as the expected value of variations of two variables from their expected values, that is, covariance measures how much variables change together. The sign of the covariance can be interpreted as follows: positive sign means two variables change in the same direction, negative sign means they change in different opposite 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.

Finally, Spearman's correlation coefficient [15] 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 summarise the strength between the two variables when it is not supposed that 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.

Table 1 shows the Pearson's correlation coefficient, covariance and Spearman's correlation coefficient between real and fake dataset for each of the column in the study. The table shows that all three coefficients are very close to zero or even zero when comparing variables in data sets two by two, indicating that there is no real correlation between the variables of the original dataset and the newly created dataset. This aspect is very important, since, as mentioned above, in areas such as medicine or banks, user data are affected by the new data protection law and must be carefully treated. 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.

Table 1. Similarity between variables

Column	Pearson index	Covariance	Spearman's index
LIMIT BAL	0.046451	0.000306	0.051436
AGE	−0.003058	0	0.000192
PAY 0	0.127477	0.000784	0.113877
PAY 2	0.086022	0.000580	0.066049
PAY 3	0.034958	0.000154	0.027432
PAY 4	0.075370	0.000449	0.056721
PAY 5	0.071400	0.000433	0.052301
PAY 6	0.062410	0.000351	0.045897
BILL AMT1	0.004100	0	0.002037
BILL AMT2	0.005826	0.000108	0.003781
BILL AMT3	0.007589	0.000175	0.005216
BILL AMT4	0.006202	0.000106	0.007436
BILL AMT5	0.004516	0	0.005910
BILL AMT6	0.001630	0	0.003899
PAY AMT1	0.017059	0.000154	0.034533
PAY AMT2	0.003157	0	0.012173
PAY AMT3	−0.001662	0	0.012225
PAY AMT4	−0.012410	0	−0.026648
PAY AMT5	0.010240	0	0.012197
PAY AMT6	−0.015963	−0.000117	−0.017851

3.3 Classification Results

After verifying that the data generated did not correlate to the real data, it was checked whether both sets of data behaved in the same way when faced with the classification task. The XGBoost [2] algorithm was used to perform this task. XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. It can be seen in Table 2 that the results obtained are practically equal for the two sets of data even without having any correlation between them. These results show that the generated data can be used indistinctly for the same tasks as the real ones obtaining the same results.

Table 2. Result of classification with XGBoost

	Accuracy	F1-score	AUC
Real dataset	0.821	0.479	0.661
Generated dataset	0.826	0.509	0.676

4 Discussion

In this paper, we proposed a Conditional Generative Adversarial Network (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. This type of network has been used and not a traditional generative adversarial network (GAN) because the data on which it has been tested had a class label that has been taken into account for the generation of new data.

The results obtained have been evaluated in two different ways: first, 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 banks, in which client data must be treated with special care so as to avoid privacy problems. Secondly, 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 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.

References

1. Beaulieu-Jones, B.K., Wu, Z.S., Williams, C., Lee, R., Bhavnani, S.P., Byrd, J.B., Greene, C.S.: Privacy-preserving generative deep neural networks support clinical data sharing. bioRxiv, p. 159756, Jan 2018. http://biorxiv.org/content/early/2018/12/20/159756.abstract
2. Chen, T., Guestrin, C.: XGBoost. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 2016 (2016)
3. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. CoRR abs/1603.02754 (2016). http://arxiv.org/abs/1603.02754
4. Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W.F., Sun, J.: Generating multi-label discrete electronic health records using generative adversarial networks. CoRR abs/1703.06490 (2017). http://arxiv.org/abs/1703.06490
5. Chollet, F., et al.: Keras (2015). https://keras.io
6. Dietz, M.: GAN-Sandbox (2017). https://github.com/mjdietzx/GAN-Sandbox
7. Generales, C.: Ley orgánica 3/2018, de 5 de diciembre, de protección de datos personales y garantía de los derechos digitales, December 2018. https://www.boe.es/buscar/doc.php?id=BOE-A-2018-16673. Accessed 14 Feb 2019
8. Goodfellow, I.J., Pouget-abadie, J., Mirza, M., Xu, B., Warde-farley, D., Ozair, S., Courville, A., Bengio, Y.: GANs. In: NIPS (2014)
9. Kim, H.Y.: Statistical notes for clinical researchers: covariance and correlation. Restorative Dent. Endod. **43**(1), e4 (2018). http://www.ncbi.nlm.nih.gov/pubmed/29487835. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5816993

10. Lichman, M.: UCI machine learning repository (2013). http://archive.ics.uci.edu/ml
11. Mirza, M., Osindero, S.: Conditional generative adversarial nets. CoRR abs/1411.1784 (2014). http://arxiv.org/abs/1411.1784
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
13. Ramponi, G., Protopapas, P., Brambilla, M., Janssen, R.: T-CGAN: conditional generative adversarial network for data augmentation in noisy time series with irregular sampling. CoRR abs/1811.08295 (2018). http://arxiv.org/abs/1811.08295
14. Rezaei, M., Yang, H., Meinel, C.: Multi-task generative adversarial network for handling imbalanced clinical data. CoRR abs/1811.10419 (2018). http://arxiv.org/abs/1811.10419
15. Schober, P., Boer, C., Schwarte, L.A.: Correlation coefficients. Anesth. Analg. **126**(5), 1763–1768 (2018). http://insights.ovid.com/crossref?an=00000539-201805000-00050
16. Sedgwick, P.: Pearson's correlation coefficient. BMJ **345**, e4483 (2012). https://www.bmj.com/content/345/bmj.e4483
17. Triastcyn, A., Faltings, B.: Generating differentially private datasets using GANs (2018). https://openreview.net/forum?id=rJv4XWZA
18. Vega, B.: Syntheticdata (2019). https://github.com/bvegaus/syntheticData
19. Xie, L., Lin, K., Wang, S., Wang, F., Zhou, J.: Differentially private generative adversarial network, February 2018
20. Yoon, J., Jordon, J., van der Schaar, M.: PATE-GAN: generating synthetic data with differential privacy guarantees. In: International Conference on Learning Representations (2019). https://openreview.net/forum?id=S1zk9iRqF7