Synthetic Data Generation Using GAN For Customer Ticket Service

Soroush Baghernezhad
Ferdowsi University of Mashhad
Mashhad, Iran
soroush.baghernezhad3@gmail.com

Abstract

We conducted an extensive investigation into the application of a vanilla Generative Adversarial Network (GAN) for synthetic data generation using a customer ticket service dataset from a large-scale company. Our study resulted in the successful generation of high-fidelity data, particularly in the context of time-independent information. This research contributes to the understanding of GAN performance and its potential in real-world data generation tasks.

Keywords: Synthetic Data, Data Generation, GAN, Customer Ticket Service, Time series

ACM Reference Format:
Soroush Baghernezhad. 2023. Synthetic Data Generation Using GAN For Customer Ticket Service. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

In our data-driven world, having ample data at your disposal translates to a significant analytical advantage. Data serves as the foundation for machine learning systems, enabling them to understand patterns and perform a wide array of tasks, from predicting stock market prices to creating entirely new images. Conversely, a shortage of data leads to problems, some evident and others less so. For example, data shortages can lead to biases, like gender bias, which can have far-reaching effects on policy making [1]. Yet, two significant challenges arise: the first involves gathering high-quality data, which can be tough, particularly in areas with limited data, such as those relying on direct human responses, which require substantial participant involvement[2]. The second challenge revolves around data privacy, especially in sensitive domains like healthcare. Failing to uphold strict privacy standards can erode trust and pose risks to both data subjects and custodians. Therefore, there's a growing need to create synthetic data, capable of filling data gaps and preserving privacy. This effort enhances the training of more accurate machine learning models tailored to various applications. In this undergoing Task

2 GANs

Generative Adversarial Networks (GANs) [3] are a type of artificial intelligence algorithm designed to address the challenge of generative modeling, which involves understanding the underlying probability distribution of a given set of training examples and generating more data instances based on this distribution. GANs have emerged as particularly effective generative models, excelling in the creation of lifelike, high-resolution images. They have found applications in various research domains, presenting unique challenges and research prospects due to their distinctive foundation in game theory, unlike other generative modeling approaches that rely on optimization methods. [4]

2.1 Vanilla GAN

A Vanilla Generative Adversarial Network (GAN) is a fundamental and foundational deep learning model composed of two neural networks, a generator and a discriminator, engaged in an adversarial training process. The generator aims to produce synthetic data samples, such as images, that closely resemble real data, while the discriminator evaluates these samples to distinguish genuine data from the generated ones. Discriminator acts similar to critic in Actor-Critic [5] algorithm in RL whereas both try to minimize the error of their actor(generator). Through iterative training, the generator becomes increasingly proficient at creating realistic data, while the discriminator becomes better at distinguishing real from fake data. This adversarial dynamic leads to the refinement of both networks, ultimately resulting in a generator capable of producing highly realistic data samples. (See fig 1)

2.2 WGAN

Wasserstein GAN, also known as WGAN, represents a variation of generative adversarial networks that optimizes an estimate of the Earth-Mover's distance (EM) as opposed to the Jensen-Shannon divergence utilized in the conventional

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

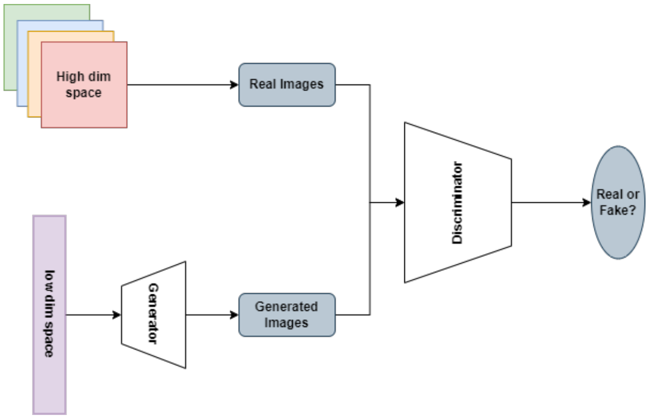

Figure 1. Base architecture of GAN composed of a Generator network and Discriminator network.

GAN framework. This approach results in a more stable training process, mitigating the issue of mode collapse while also providing interpretable learning curves that prove invaluable for debugging and the exploration of hyperparameters.[6] Mode collapse is a problem that arise from generator when it learns to deceive the discriminator by generating a sample which satisfy the objective of discriminator but does not correlate with real data.

2.3 TGAN

To generate realistic time-series data while preserving temporal dynamics and variable relationships, a novel framework has been introduced. This framework combines the flexibility of unsupervised learning with the control afforded by supervised training. By optimizing a learned embedding space through both supervised and adversarial objectives, the network is encouraged to emulate training data dynamics during the generation process. Empirical assessments confirm that this framework consistently outperforms existing benchmarks in terms of similarity and predictive accuracy across various time-series datasets. [7]

3 Proposed Method

In this section, we proposed a method for synthetic data generation regarding to customer ticket service. For the sake of simplicity, we have not considered this data set as a time-series and we ignored the dependency of 'customer-satisfaction' and 'task-type' features to time.

3.1 Preparing Data

Initially, data were downloaded and datetime objects that were inserted as string type were converted to python's datetime object, and then they were converted to timestamps. Then we performed profiling on dataset to see the overall statistics of data and correlation between columns.

Customer's given score	Number of records
5	100547
4	36844
3	42061
2	38475
1	88492

Highly correlated columns were omitted and categorical features were mapped to a corresponding number.
continuous features(datetime) were discretized into 24 equally sized bins starting from zero. Then all rows were power transformed to make data Gussian-like. **yeo-johnson**

3.2 Defining Model

Generator model have 172, 423 trainable params with 4 dense layers as below:

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(32, 32)]	0
dense (Dense)	(32, 128)	4,224
dense_1 (Dense)	(32, 256)	33,024
dense_2 (Dense)	(32, 512)	131,584
dense_3 (Dense)	(32, 7)	3,591

Discriminator model have 168, 449 trainable params with 4 dense layers as follow:

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(32, 7)]	0
dense_4 (Dense)	(32, 512)	4,096
dropout (Dropout)	(32, 512)	0
dense_5 (Dense)	(32, 256)	131,328
dropout_1 (Dropout)	(32, 256)	0
dense_6 (Dense)	(32, 128)	32,896
dense_7 (Dense)	(32, 1)	129

3.3 Training and Evaluating Model

We trained the synthesizer for 10, 000 epochs with batch size of 32 and the learning rate of $5e - 4$. We evaluated the generated data set feature-wise via table-evaluator tool.

4 Results

After the training, model accuracy was %79.69 with high variance.
In Fig 2, it can be seen that the real data has a higher ALM and lower STD than the fake data, suggesting that it is more concentrated around its mean and less spread out. This suggests that the fake data is more noisy.
Figs 3 & 4 illustrate a high similarity between the generated features by the synthesizer. however, in mid ranges, the synthesizer cannot generate accurate data and it face some inaccuracies regarding the discontinuing and categorical nature of the values; this problem can be mitigated by some post-process on generated data set.

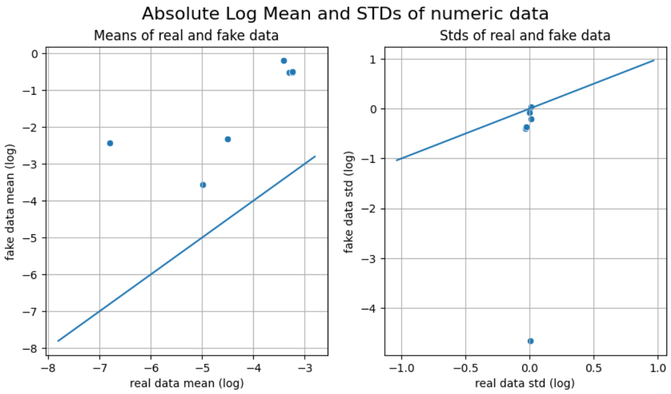

Figure 2. ALM and STD of both data.

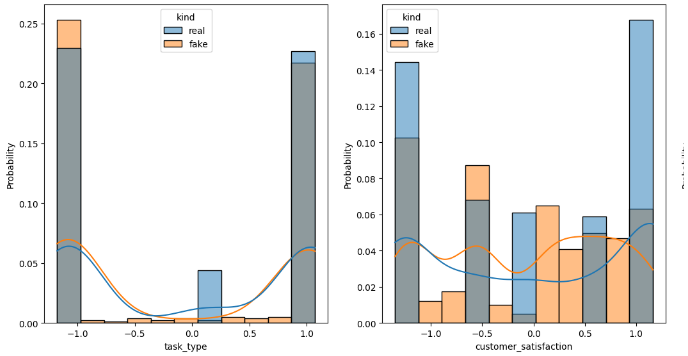

Figure 3. Distribution per feature for task type and customer satisfaction features.

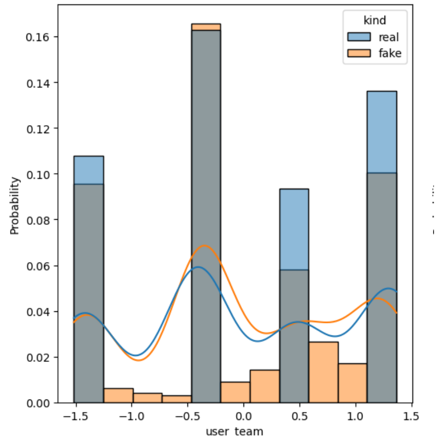

Figure 4. Distribution per feature for user team feature.

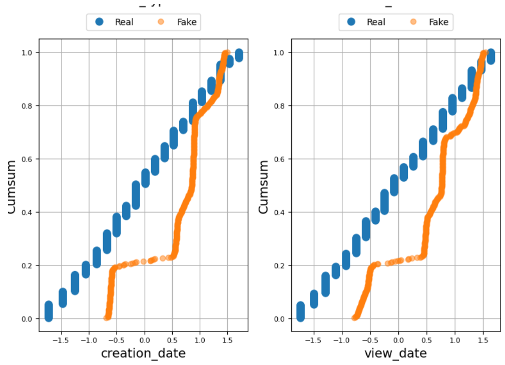

Figure 5. Cumulative sums per feature for view and creation date.

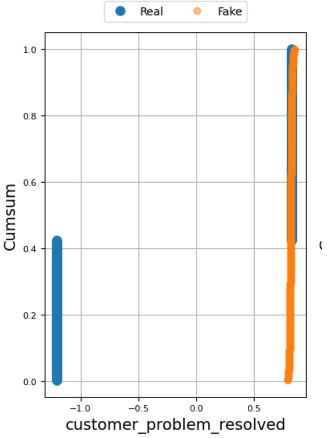

Figure 6. Cumulative sums per feature for customer problem resolved.

Despite seemingly good performance on categorical features, our model was not able to capture time-dependent and continuous data, it also could not distinguish binary data.
In fig 5 there is a stair-shaped pattern which is forming but after 10000 epochs it could not generate even discretized time series. Fig 6 shows the imbalance between binary feature of true and false indicating whether the problem was solved.

In conclusion, Vanilla GAN performed well on categorical data but it was weak against continuous and binary data.

5 Future Studies

For further exploring synthetic data generation on the under going data set, using other variants of GANs are recommended. Other direction of study can be focusing on Variational Auto Encoders [8], diffusion based models [9] or large language models [10].
Feature engineering and parameter tuning of current model can also improve the fidelity and accuracy of generated data.

References

[1] C.-P. Caroline, *Invisible Women : Data Bias in a World Designed for Men*, Second. New York: Abrams Press, 2019.

[2] A. Figueira and B. Vaz, "Survey on synthetic data generation, evaluation methods and gans," *Mathematics*, vol. 10, no. 15, 2022, ISSN: 2227-7390. DOI: 10.3390/math10152733. [Online]. Available: https://www.mdpi.com/2227-7390/10/15/2733.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27, Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, 139–144, 2020, issn: 0001-0782. doi: 10.1145/3422622. [Online]. Available: https://doi.org/10.1145/3422622.

[5] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen, and K. Müller, Eds., vol. 12, MIT Press, 1999. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[6] M. Arjovsky, S. Chintala, and L. Bottou, *Wasserstein gan*, 2017. arXiv: 1701.07875 [stat.ML].

[7] J. Yoon, D. Jarrett, and M. Van der Schaar, "Time-series generative adversarial networks," *Advances in neural information processing systems*, vol. 32, 2019.

[8] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Foundations and Trends® in Machine Learning*, vol. 12, no. 4, pp. 307–392, 2019. doi: 10.1561/2200000056. [Online]. Available: https://doi.org/10.1561%2F2200000056.

[9] S. Azizi, S. Kornblith, C. Saharia, M. Norouzi, and D. J. Fleet, *Synthetic data from diffusion models improves imagenet classification*, 2023. arXiv: 2304.08466 [cs.CV].

[10] V. Veselovsky, M. H. Ribeiro, A. Arora, M. Josifoski, A. Anderson, and R. West, *Generating faithful synthetic data with large language models: A case study in computational social science*, 2023. arXiv: 2305.15041 [cs.CL].