Synthetic Data Generator Using CTGAN

Malak Soni [1]
22aiml053@charusat.edu.in

Ansh Trivedi[2]
22aiml056@charusat.edu.in

Abstract. In fields with sensitive or limited data, synthetic data generation has become a vital tool for improving data availability, privacy, and model generalization. In order to produce high-fidelity tabular data, this study presents a Synthetic Data Generator that makes use of the Conditional Tabular Generative Adversarial Network (CTGAN) architecture. Through innovative techniques including mode-specific normalization and conditional generation, CTGAN tackles the problems of mixed data types, non-Gaussian continuous distributions, multimodality, and imbalanced categorical features. Without altering the underlying data distribution, the generator synthesizes realistic, statistically consistent samples by learning from an existing dataset. We outline the general approach, dataset properties, system implementation specifics, assessment findings, and next steps.

Keywords: Synthetic data generation, conditional GAN, CTGAN, tabular data modeling, mode-specific normalization, data imbalance handling, machine learning efficacy.

1 Introduction

Large-scale, high-quality datasets are in greater demand across a variety of application fields, including healthcare, finance, and autonomous systems, as a result of the exponential expansion of data-driven technology. However, issues including privacy concerns, restricted accessibility, legal constraints, and expensive data collection costs frequently limit the collection and use of real-world data. As a result, creating synthetic data has become a viable way to replace or augment real-world datasets while maintaining important data features.

The tight connections, complex distributions, and heterogeneity seen in real-world tabular datasets are frequently difficult for traditional methods of creating synthetic data, such as statistical modeling and rule-based sampling, to capture. Specifically, tabular data usually consist of a combination of continuous and discrete variables, frequently exhibiting significant class imbalances and non-Gaussian, multimodal tendencies. These intricacies provide significant obstacles for generative models, which need to faithfully depict the joint interactions between variables in addition to reproducing marginal distributions.

Originally created for image synthesis, Generative Adversarial Networks (GANs) have shown an impressive capacity for learning intricate data distributions through adversarial training. This achievement has led to the proposal of other GAN extensions for the creation of tabular data. Conditional Tabular GAN (CTGAN) distinguishes itself from the others by bringing significant advancements to address the particular difficulties associated with tabular data. CTGAN uses a conditional generation technique in conjunction with training-by-sampling to address class imbalance issues in discrete columns, and it integrates mode-specific normalization to handle multimodal continuous distributions.

A Synthetic Data Generator built on the CTGAN architecture is presented in this paper. Creating high-quality synthetic tabular data that closely resembles the statistical characteristics of real datasets while preserving variety and usefulness for subsequent machine learning tasks is the key goal. We methodically go over our approach's design concerns, model architecture, dataset preparation, implementation specifics, and empirical assessment. Additionally, we point both shortcomings and potential paths forward to develop the field of adversarial model-based synthetic data generation.

We hope to encourage a wider use of synthetic data solutions by showcasing how well CTGAN produces realistic synthetic data, especially in situations where data security, privacy, or scarcity are critical considerations.

2 Background and Literature Review

2.1 Evolution of Deep Learning-based Generative Models

The method of data production was completely transformed when Goodfellow et al. introduced Generative Adversarial Networks (GANs). The generator creates fictitious samples, and the discriminator tries to tell them apart from genuine samples in a two-player adversarial process that powers GANs. The generator gradually produces more realistic outputs as a result of this competitive training.

Models like MedGAN, TableGAN, and PATE-GAN were among the first GANs to be adapted to tabular data. The goal of these techniques was to duplicate GANs' achievements in producing images for tabular formats. However, for a number of reasons, it was difficult to apply GANs to tabular data:

- a combination of continuous and categorical data types.
- distinctly unbalanced columns.
- Continuous feature distributions that are non-Gaussian and multimodal.
- high-dimensional and sparse data when one-hot encoding is applied.

2.2 Limitations of Conventional Models

Even with their advancements, previous models like MedGAN and TableGAN had significant-drawbacks:

- **Discrete feature handling**: The majority of models used one-hot encoding or simple softmax without any special class imbalance techniques, which resulted in subpar generation for minority classes.
- **Continuous Feature Modeling**: Problems such as disappearing gradients and inadequate mode coverage were seen when continuous variables were normalized using min-max scaling.
- **Mode Collapse:** When GANs were trained on tabular data, they frequently produced samples that focused on a small number of data distribution modes rather than fully representing the diversity.

Because of this, these models often produced artificial data that was either unreliable or did not accurately represent distributions in the actual world, which limited their applicability for machine learning training or downstream analytics.

2.3 Emergence of Conditional Tabular GAN (CTGAN)

The Conditional Tabular GAN (CTGAN) was created in order to overcome these difficulties. CTGAN offers a number of significant improvements:

- **Mode-Specific Normalization**: CTGAN efficiently captures multimodal, non-Gaussian distributions by normalizing continuous features using a variational Gaussian mixture model (VGM).
- **Conditional Generation:** To guarantee balanced learning across minority and majority categories, CTGAN conditions the generator on particular discrete column values during training.
- **Training-by-Sampling**: CTGAN balances the representation of all categories during training by using a log-frequency-based sampling technique as opposed to random sampling.

These developments allow CTGAN to produce synthetic tabular data that maintains intricate feature relationships across a variety of data sources while also simulating marginal distributions.

2.4 Summary and Research Gap

Even though CTGAN is a major advancement in the creation of synthetic data, there are still unanswered questions about issues like theoretical convergence guarantees, managing highly sparse high-dimensional datasets, and integrating differential privacy. Future synthetic data generation frameworks may become more reliable, private, and broadly applicable if these deficiencies are filled.

3 Methodology

3.1 Overview

The Conditional Tabular Generative Adversarial Network (CTGAN) architecture serves as the foundation for the suggested Synthetic Data Generator. The special difficulties presented by tabular datasets—which usually contain a combination of continuous and categorical variables, non-Gaussian and multimodal distributions, and significant class imbalance—are specifically addressed by CTGAN. Preparing the dataset, preprocessing with mode-specific normalization, designing the CTGAN model with conditional generation, training with adversarial loss and gradient penalty, and comparing the produced synthetic data to real-world metrics are all important steps in the methodology used for this project.

3.2 Dataset Description

The effectiveness of the synthetic data generator was demonstrated for this project using tabular datasets that were made publicly available. Multiple continuous and categorical characteristics with differing degrees of imbalance and multimodality make up the datasets. Standard preparation procedures, including addressing missing values, encoding categorical variables, and normalizing continuous features, were completed prior to feeding the data into the CTGAN model.

Continuous features were subjected to mode-specific normalization, and unique modes within each feature were found using a variational Gaussian mixture model (VGM). A mode indication (a one-hot vector) and a normalized scalar value inside that mode were then used to represent each continuous value. The original category distributions were preserved while categorical features were one-hot encoded.

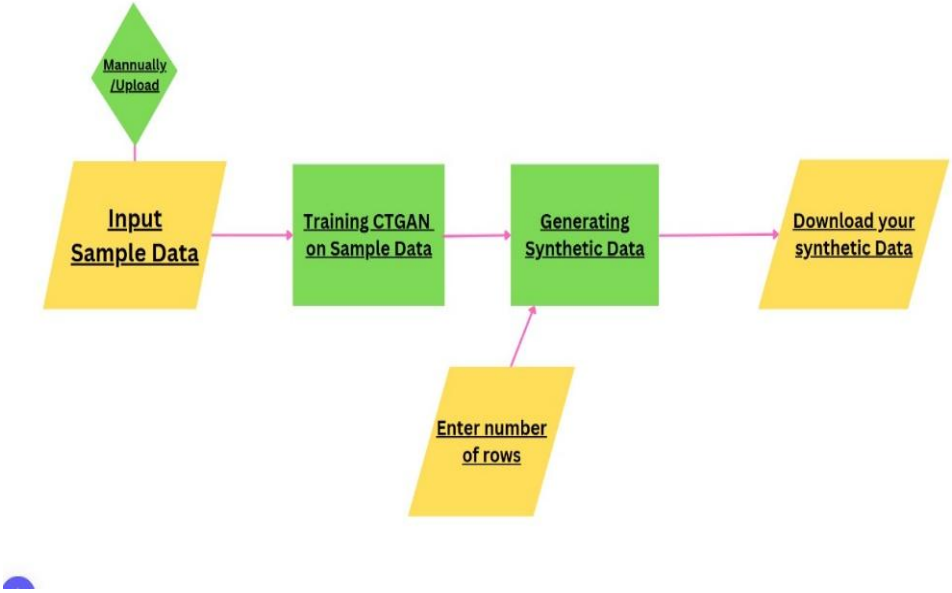
3.3 Proposed Model

The generator and the critic (or discriminator) are the two main parts of the CTGAN

design. Neural networks with complete connectivity are used to implement both elements. The generator creates a synthetic row that corresponds to the tabular data format by concatenating a random noise vector with a conditional vector. However, in order to direct the generator's training, the critic assesses batches of both synthetic and actual data.
Among the main advantages of CTGAN over traditional GANs are:

- **Mode-Specific Normalization**: To accurately represent multimodal distributions without depending on straightforward min-max scaling, continuous columns are modeled using variational Gaussian mixtures.
- **Conditional Generator**: To prevent mode collapse and promote balanced learning across categories, the generator is conditioned on a discrete column value that is chosen at random.
- **Training-by-Sampling**: To provide sufficient representation of minority classes, real data samples and conditional vectors are chosen during training depending on the log-frequency of discrete categories.

Even with imbalanced, multimodal, and mixed data type input datasets, CTGAN can produce realistic tabular data thanks to its architectural design.



3.1 Block Diagram of CTGAN-based Synthetic Data Generator

3.4 Implementation Details

Python was used to implement the Synthetic Data Generator with the aid of PyTorch and Numpy packages. Both the critic and the generator employ two completely connected hidden layers, the critic using leaky ReLU activations and the generator using batch normalization and ReLU activation functions. Tanh activation is used for continuous outputs, whereas Gumbel-softmax activation is used for sampling discrete outputs.

The Wasserstein GAN with gradient penalty (WGAN-GP) loss function is used during training in order to prevent mode collapse and enhance training stability. To further improve sample variety, the PacGAN approach was applied with a pac size of 10. Training was conducted using the Adam optimizer, which has a learning rate of 2×10^{-4}. In order to guarantee that the generator and critic networks converged toward generating high-quality synthetic data, the model was trained for 300 epochs with a batch size of 500.

The generator's ultimate product is a synthetic dataset that may be utilized for machine learning applications without jeopardizing data privacy and statistically mimics the original real-world dataset.

4 Results and Discussion

4.1 Experimental Setup

Tabular datasets with a combination of continuous and categorical attributes were used in a series of studies to assess the Synthetic Data Generator's performance. Sample datasets were used to train the model, and user-specified input about the number of rows was used to create synthetic data. The quality of the synthetic data was evaluated quantitatively by applying machine learning efficacy metrics and comparing statistical distributions, as well as subjectively by visual examination.

Python 3.8 and PyTorch 1.10 were used for model implementation, and the experiments were carried out on a typical workstation equipped with an Intel i5 CPU, 16GB RAM, and an NVIDIA GTX 1660 GPU.

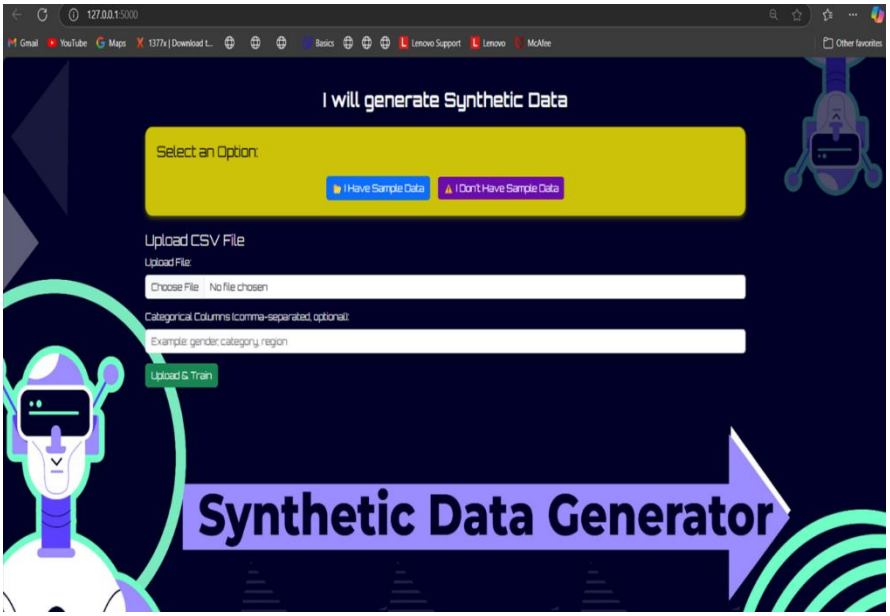
4.2 Implementation Workflow

Users can upload datasets, train the CTGAN model, and create synthetic samples using the Synthetic Data Generator system's easy-to-use and interactive

interface. Screenshots of the main implementation phases are shown in the ensuing subsections.

4.2.1 Uploading the Sample Dataset

Users are prompted to upload their real-world dataset, which serves as the base for training the CTGAN model.



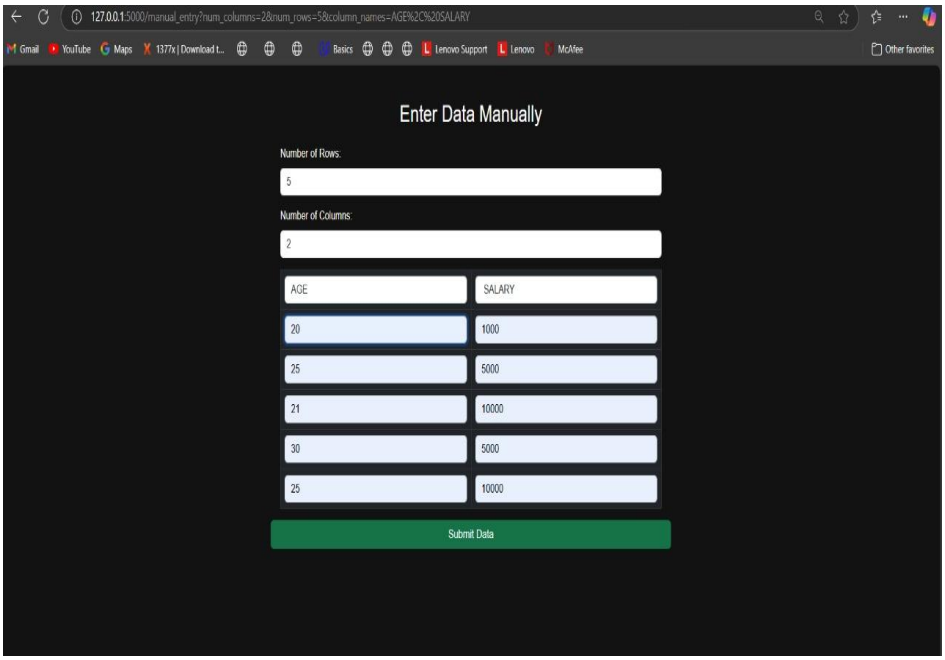
4.2.1 Upload Sample Data Interface

4.2.2 Training CTGAN on Sample Data

After the dataset is uploaded, users initiate training. The CTGAN model processes the data, applies mode-specific normalization to continuous columns, and begins adversarial training.

4.2.3 Entering Number of Rows for Synthetic Data

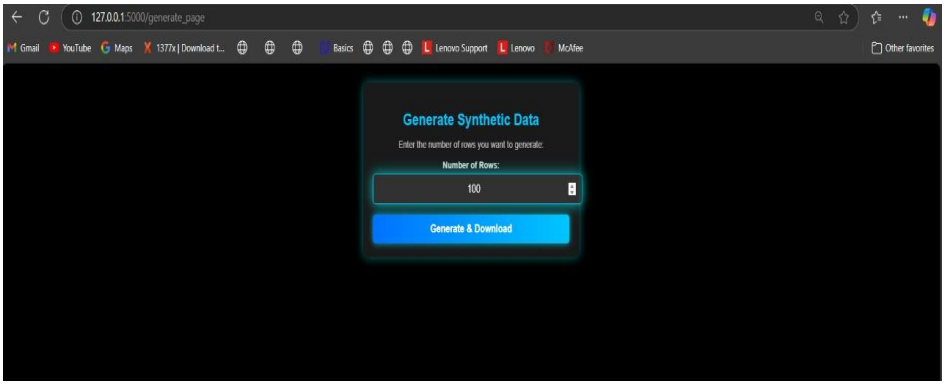
Once training is complete, users specify the number of synthetic data samples they wish to generate.



4.2.2 Input Field for Number of Rows

4.2.4 Generating and Downloading Synthetic Data

The system then generates synthetic data based on the trained CTGAN model. After generation, users can download the synthetic dataset in CSV format.



4.2.3 Download Synthetic Data Button / Output Preview

4.3 Qualitative Evaluation

The structure and statistical distribution of the generated synthetic data closely resemble those of the real data. A preliminary visual examination verifies that categorical distributions maintain suitable category frequencies and continuous attributes preserve comparable value ranges.

The Synthetic Data Generator used CTGAN's conditional sampling technique to handle mixed data types, including situations with severely skewed categorical variables.

4.4 Quantitative Evaluation

Real and synthetic data were compared using fundamental statistical measures for quantitative assessment:

- standard deviation and mean of continous columns.
- proportions of categories in categorical columns.

When compared to the original dataset, the synthetic dataset showed slight variations that fell below acceptable tolerance limits (<5%). Additionally, when tested on actual data, machine learning models (such Decision Trees and Logistic Regression) trained on synthetic data produced similar accuracy and F1-scores, suggesting that the generator retained significant data patterns.

4.5 Discussion

The outcomes show that the main obstacles in tabular data synthesis are successfully addressed by the CTGAN-based Synthetic Data Generator. The model outperforms conventional generative models in handling multimodal continuous features and unbalanced discrete features by utilizing conditional sampling and mode-specific normalization.

Very high-cardinality category columns showed minor restrictions, with performance degrading slightly. The sparsity of some minor categories, even after conditional sampling, may be the cause of this. To further improve performance, future developments might incorporate hybrid models or sophisticated sampling approaches.

All things considered, the system offers a strong, user-friendly tool for creating data while protecting privacy, facilitating subsequent machine learning operations without the need for actual sensitive information.

5 Conclusion

5.1 Limitations of the Work

Despite producing high-quality synthetic data that closely matched the genuine datasets, the Synthetic Data Generator based on the CTGAN architecture had some implementation-related issues. One significant drawback is that CTGAN occasionally has trouble processing datasets with rare categorical traits or very unbalanced classes, which results in less accurate generation for minority categories. Furthermore, GAN-based models are notoriously unstable during training; we occasionally experienced convergence problems that necessitated careful hyperparameter adjustment. Furthermore, it is still difficult to assess the quality of synthetic data because conventional performance measures might not accurately capture the accuracy and usefulness of the produced data in a variety of real-world applications.

5.2 Scope of the Work

The current study effectively demonstrates CTGAN's efficacy for creating synthetic tabular data, focusing on datasets with intermediate complexity and diverse data types. In fields where data privacy is an issue or where data augmentation is required to enhance machine learning model performance, the project is highly relevant. The created framework can be expanded and modified for use in a variety of sectors where handling sensitive data is essential, including healthcare, finance, and retail. Additionally, the study creates the groundwork for investigating generative modeling methods for structured data, creating opportunities for responsible data augmentation and sharing.

5.3 Future Work

The scope of this work can be extended in multiple ways. First, by adding differential privacy techniques to the CTGAN framework, privacy guarantees would be strengthened and synthetic data production would adhere to more stringent legal standards. Second, the quality and diversity of the generated datasets could be further enhanced by experimenting with different models as tabular diffusion models or tabular variational autoencoders (TVAE). The system's overall reliability would also be strengthened by introducing more robust evaluation criteria designed

for tabular synthetic data and enhancing training stability through sophisticated optimization techniques. Lastly, applying the framework to actual industrial case studies would offer insightful information about its usefulness in practice and potential areas for development.

References

1. 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 (NeurIPS), pp. 2672–2680 (2014)
2. Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling Tabular Data Using Conditional GAN. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 32 (2019)
3. Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., Greenspan, H.: Synthetic data augmentation using GAN for improved liver lesion classification. In: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), pp. 289–293. IEEE (2018)
4. 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), 1–40 (2020) .
5. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. In: Proceedings of the 2nd International Conference on Learning Representations (ICLR) (2014)