Digital Health and Informatics Innovations for Sustainable Health Care Systems
J. Mantas et al. (Eds.)

doi:10.3233/SHTI240571

Large Language Models for Synthetic Tabular Health Data: A Benchmark Study

Marko MILETIC[a] and Murat SARIYAR[a,1]
[a]*Bern University of Applied Sciences, Switzerland*
ORCiD ID: Murat Sariyar https://orcid.org/0000-0003-3432-2860

Abstract. Synthetic tabular health data plays a crucial role in healthcare research, addressing privacy regulations and the scarcity of publicly available datasets. This is essential for diagnostic and treatment advancements. Among the most promising models are transformer-based Large Language Models (LLMs) and Generative Adversarial Networks (GANs). In this paper, we compare LLM models of the Pythia LLM Scaling Suite with varying model sizes ranging from 14M to 1B, against a reference GAN model (CTGAN). The generated synthetic data are used to train random forest estimators for classification tasks to make predictions on the real-world data. Our findings indicate that as the number of parameters increases, LLM models outperform the reference GAN model. Even the smallest 14M parameter models perform comparably to GANs. Moreover, we observe a positive correlation between the size of the training dataset and model performance. We discuss implications, challenges, and considerations for the real-world usage of LLM models for synthetic tabular data generation.

Keywords. Synthetic data generation, tabular data, large language models, GAN

1. Introduction

In the healthcare sector, various types of data, including structured tabular data, are collected. Stringent laws such as Switzerland's human research legislation and the European GDPR require privacy preservation. To meet these requirements, efficient methods for generating synthetic data are essential to facilitate open research, particularly for healthcare institutions with limited resources. Synthetic data plays a crucial role in advancing AI applications in healthcare by offering augmented and representative alternatives to real data, thereby reducing privacy concerns. This is essential for progress in diagnosis, treatment, and advancements in patient care [1].

Recent state-of-the-art methods for generating synthetic tabular data using transformer-based Large Language Models (LLMs) are promising alternatives to Generative Adversarial Networks (GANs). Unlike GANs, LLMs can deal with the complexity of high dimensionality out-of-the-box due to its autoregressive attention-based mechanism, promising to offer effective and efficient synthetic tabular data generation [2,3]. The traditional GAN architecture requires preprocessing steps to encode data into a suitable format for training, depending on data types and distributions. A significant advantage of LLMs is their capability to bypass the need for one-hot

[1] Corresponding Author: Murat Sariyar, Bern University of Applied Sciences, Quellgasse 21, CH2502 Biel/Bienne, Switzerland; E-mail: murat.sariyar@bfh.ch.

encoding of categorical data, which would increase the data dimensionality, by treating tabular data as full text during preprocessing. However, a drawback is that they may not be as efficient in the training process compared to GANs, even though they themselves suffer from issues like local-minima, mode-collapse, unstable training, and – depending on the specific architecture – the necessity of separately synthesizing categorical and numerical data. Avoiding the latter issue is crucial for capturing the interplay and correlations between columns [4-7]. Training instability of GANs can be addressed, for instance, by using the Wasserstein loss with gradient penalty [5,6]. To address the efficiency challenge in LLMs, techniques such as token sequence compression and different token padding strategies have been proposed [2].

In the following, we will compare LLM models from the Pythia LLM Scaling Suite, with model sizes ranging from 14M to 1B [8], against two models: the reference CTGAN model and a model trained on the original dataset. The generated synthetic data are used to train random forest estimators for making predictions on the real-world data in classification tasks. Our comparison will consider how the quantity of training data influences the utility of synthetically generated data.

2. Methods

2.1. Datasets

Three tabular datasets with different characteristics were obtained from the UCI Machine Learning Repository (CDC Diabetes Health Indicators[2]; Adult[3]) and Kaggle (Smoking and Drinking Dataset with Body Signals[4]). Two of these datasets pertain to healthcare and are commonly utilized for classification tasks, while the third (Adult) represents demographic data. All three datasets contain binary target variables.

In the preprocessing step, missing data were removed from all datasets to ensure consistency in sample sizes for subsequent utility calculations. The synthetic generation of missing data was initially considered, but it was ultimately omitted due to concerns about the reliability and accuracy of the generated data. Synthetic data generation techniques may not always capture the complex patterns and nuances present in real-world data, especially when dealing with missing data points. Additionally, the potential for introducing biases or inaccuracies in the synthetic data could have adverse effects on downstream analyses or applications. Therefore, it was decided to omit synthetic generation of missing data to ensure the integrity and validity of the dataset. Some features of the original datasets were excluded to decrease training time.

Table 1. Description of datasets. Dataset abbreviations are provided in parentheses. "Cont." denotes continuous columns, "Cat." represents categorical columns, and "Bin." indicates binary columns.

Dataset	Size	Features	Cont.	Cat.	Bin.
Adult (ADT)	32,561	15	4	9	2
CDC Diabetes Health Indicators (CDI)	253,680	22	3	4	15
Smoking and Drinking Dataset (SDD)	991,346	19	5	10	4

[2] https://archive.ics.uci.edu/dataset/891/cdc+diabetes+health+indicators

[3] https://archive.ics.uci.edu/dataset/2/adult

[4] https://www.kaggle.com/datasets/sooyoungher/smoking-drinking-dataset

2.2. Experimental Setup

We utilize the Pythia LLM models (14M, 31M, 70M, 160M and 1B) with the Tabula Framework – a further development of the GReaT Framework [2,3,8]. We employ token sequence compression and use left padding [2]. The sampling process of the LLM models is conditioned on the binary target variable, which serves as a start token (e.g., "Class 0" or "Class 1"). As the GAN competitor, we apply CTGAN, a framework to learn the distribution of tabular data [4]. It utilizes mode-specific normalization and a conditional generator to generate rows. Values for the model hyperparameters are taken from the respective original papers and settings in the corresponding software implementations. The model hyperparameters, sourced from their original papers and software implementations, are applied. Both LLMs and CTGAN are trained for 400 epochs, with their respective original batch sizes. We train each Pythia LLM model based on either a randomly initialized state (in this case, pre-training is conducted via next-token prediction) or on a pretrained starting point (in this case, fine-tuning is conducted). This approach enables us to evaluate the significance of the architecture alone versus the presence of a language model. We employ random sampling to generate training datasets from the original data, with each containing 500, 1000, 2500, and 5000 rows, respectively. From the remaining original data, we randomly sample corresponding test data with the same size. For assessing the quality of the generated synthetic data, we use the classification accuracy on the test set that is achieved by a random forest model trained on the synthetic train set. The average random forest accuracy score and its standard deviation across 100 runs of synthetic data generation for each trained model variant is reported.

3. Results

Table 2 contains the results for Original (random forests are trained on the original training data), CTGAN as well as LLMs with different parameter sizes. When comparing Original and CTGAN models with LLMs it becomes clear that the LLM based approach surpasses them across the three different datasets. Some models even slightly surpass the utility of the original data. This is a sign that the models do indeed not only learn the structure of the of the original datasets during training but rather learn the underlying distributions as well as the correlations between the variables. The overall trend indicates that LLM models tend to perform better in capturing complex joint probability distributions as the number of parameters increases. The same holds true for the sample size of the training dataset, suggesting a positive correlation between both factors.

An interesting observation can be made regarding the CDI and SDD datasets: if the datasets contain a large portion of categorical columns, it appears that it is not necessary to employ an LLM model with high complexity. This observation suggests that depending on the dataset and the types of features it contains, smaller LLMs may be sufficient. Even though GANs have significantly faster training and generation times compared to LLM-based approaches, it's important to note that the training time for both increases with the sample size provided for training. Whether randomly initialized and subsequently fine-tuned LLMs exhibit better accuracy scores than models that were pre-trained and then fine-tuned on the specific dataset cannot be determined definitively.

Table 2. Machine learning utility results for synthetic data generation include the mean accuracy and standard deviation obtained using random forest for all datasets. For LLMs, "a)" signifies models that were randomly initialized and subsequently trained, while "b)" indicates models that were pre-trained and then fine-tuned on the dataset. Best results for a) and b) are given in bold.

Dataset	Original	CTGAN	Pythia 14M	Pythia 31M	Pythia 70M	Pythia 160M	Pythia 1B
ADT	.839	.750±.001	a).731±.038	a).766±.016	a).797±.012	a).803±.011	a).801±.016
(500)			b).784±.018	b).769±.019	b).792±.017	b).800±.016	b).809±.011
ADT	.849	.750±.016	a).758±.013	a).785±.015	a).807±.016	a).819±.012	a).819±.011
(1000)			b).783±.019	b).785±.018	b).809±.016	b).817±.012	b).821±.012
ADT	.856	.793±.007	a).779±.019	a).813±.012	a).820±.015	a).826±.017	a).824±.018
(2500)			b).798±.021	b).814±.015	b).819±.021	b).825±.016	b).829±.014
ADT	.856	.821±.004	a).809±.019	a).830±.006	a).834±.013	a).841±.012	**a).842±.005**
(5000)			b).822±.017	b).832±.012	b).837±.010	b).839±.008	**b).841±.010**
CDI	.861	.860±.000	a).860±.001	a).860±.001	a).860±.002	a).862±.003	a).861±.003
(500)			b).859±.002	b).860±.004	b).859±.003	b).859±.003	b).861±.002
CDI	.861	.860±.000	a).859±.001	a).860±.002	a).858±.003	a).859±.003	a).861±.003
(1000)			b).857±.004	b).858±.003	b).859±.003	b).859±.003	b).861±.002
CDI	.862	.840±.004	a).860±.002	a).860±.003	a).861±.002	a).862±.003	a).859±.003
(2500)			b).858±.003	b).860±.003	b).860±.002	b).861±.002	b).861±.002
CDI	.862	.860±.000	a).855±.003	**a).863±.002**	a).862±.002	a).862±.002	a).862±.002
(5000)			b).861±.003	b).862±.003	**b).863±.002**	b).862±.003	b).861±.002
SDD	.694	.473±.036	a).606±.017	a).664±.009	a).670±.007	a).678±.006	a).678±.007
(500)			b).666±.010	b).670±.007	b).675±.007	b).665±.005	b).679±.005
SDD	.699	.556±.028	a).659±.013	a).678±.005	a).681±.004	a).680±.005	a).685±.005
(1000)			b).675±.008	b).681±.005	b).682±.005	b).680±.005	b).683±.004
SDD	.709	.676±.002	a).679±.004	a).689±.004	a).689±.004	a).687±.004	a).690±.004
(2500)			b).687±.003	b).690±.003	b).689±.004	b).686±.003	b).692±.004
SDD	.711	.568±.006	a).689±.003	**a).695±.003**	a).693±.004	a).692±.003	a).694±.003
(5000)			**b).695±.003**	b).689±.003	b).693±.003	b).694±.003	b).693±.003

4. Discussion and Conclusions

Our benchmark shows that LLM models generally tend to perform marginally better as the number of parameters increases, as well as with the volume of data they are pre-trained or fine-tuned on. The datasets considered represent different levels of complexity, which was crucial to assess the context-dependency of the different synthetic data approaches. For instance, on the CDI or the SDD dataset, increasing the number of parameters does not improve results by a large amount. Further LLMs seem to produce better results when given limited training data compared to the reference CTGAN method. This insight could serve as a valuable guide for practitioners, who want to augment their data with only a limited number of real-world data at hand. A major drawback of LLMs, especially the larger ones, is their high demand for specialized machine learning hardware. For instance, the largest model, with 1 billion parameters, required approximately 40GB of VRAM spread across two GPUs for training, while even the smallest 14M parameter model needed 22GB of VRAM.

The target variable plays a central role in autoregressive LLM models, potentially making them less suitable as general-purpose applications, not focusing on specific classification tasks. For LLMs to address this, they would need to incorporate feature permutation during model training. While this is technically feasible, it may not be practical due to efficiency concerns [2,3]. In contrast, GANs are generally agnostic to the sequence in which variables are learned. The comparison suggests that the approach to conditioning is not arbitrary but rather significant. Further, the assertion in the literature that randomly initialized models outperform pretrained LLMs could not be substantiated. Addressing this would necessitate conducting more extensive tests and conducting specific research into initialization methods for LLMs. It's important to note that the type of data a model is pre-trained on can significantly impact the performance

of the resulting fine-tuned model. Had the Pythia LLMs been trained on a domain-specific dataset rather than the 800G pile dataset [9], different results might have been achieved [2]. This could significantly boost performance, as relationships between variables or concepts would already be encoded in the pre-trained model. In cases where domain-specific datasets for pre-training are unavailable, it is sensible to utilize a randomly initialized LLM. This approach avoids the need for the LLM to unlearn contextual information from the pre-training.

Depending on the use case, feature permutation might become necessary if different targets variables should be allowed, which would lead to significantly longer training times. Hence, the drawbacks of prolonged training times and the need for demanding hardware would likely outweigh the benefits in practice. As a general rule, it's worth noting that when dealing with the complexity of tabular data, opting for larger models may not be rational if the performance of smaller models is sufficient. For actors with limited computational resources, this means enhancing the utility of statistical analysis through augmented real-world data. Further research should be conducted to analyze the potential of LLMs for synthetic tabular data generation compared to other emerging technologies, such as diffusion models [10].

Acknowledgments: This study was funded by BRIDGE, a joint programme of the Swiss National Science Foundation SNSF and Innosuisse (grant number 211751).

References

[1] Giuffrè M, Shung DL. Harnessing the power of synthetic data in healthcare: innovation, application, and privacy. npj Digit Med. 2023 Oct 9;6(1):186.
[2] Zhao Z, Birke R, Chen L. TabuLa: Harnessing Language Models for Tabular Data Synthesis. 2023 [cited 2024 Apr 3]; Available from: https://arxiv.org/abs/2310.12746
[3] Borisov V, Seßler K, Leemann T, Pawelczyk M, Kasneci G. Language Models are Realistic Tabular Data Generators. 2022 [cited 2024 Apr 3]; Available from: https://arxiv.org/abs/2210.06280
[4] Xu L, Skoularidou M, Cuesta-Infante A, Veeramachaneni K. Modeling Tabular data using Conditional GAN. 2019 [cited 2024 Apr 3]; Available from: https://arxiv.org/abs/1907.00503
[5] McKeever S, Walia MS. Synthesising Tabular Datasets Using Wasserstein Conditional GANS with Gradient Penalty (WCGAN-GP). 2020 [cited 2024 Apr 3]; Available from: https://arrow.tudublin.ie/scschcomcon/285/
[6] Yoon J, Drumright LN, van der Schaar M. Anonymization Through Data Synthesis Using Generative Adversarial Networks (ADS-GAN). IEEE J Biomed Health Inform. 2020 Aug;24(8):2378–88.
[7] Lu Y, Shen M, Wang H, van Rechem C, Wei W. Machine Learning for Synthetic Data Generation: A Review [Internet]. arXiv; 2023 [cited 2023 Nov 28]. Available from: http://arxiv.org/abs/2302.04062
[8] Biderman S, Schoelkopf H, Anthony Q, Bradley H, O'Brien K, Hallahan E, et al. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. 2023 [cited 2024 Apr 3]; Available from: https://arxiv.org/abs/2304.01373
[9] Gao L, Biderman S, Black S, Golding L, Hoppe T, Foster C, et al. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. 2021 [cited 2024 Apr 3]; Available from: https://arxiv.org/abs/2101.00027
[10] Kotelnikov A, Baranchuk D, Rubachev I, Babenko A. TabDDPM: Modelling Tabular Data with Diffusion Models. 2022 [cited 2024 Apr 3]; Available from: https://arxiv.org/abs/2209.15421