Benchmarking Tabular Data Synthesis for User Guidance

Maria F. Davila[1,2]

Supervised By: Wolfram Wingerath[2] and Fabian Panse[3]

[1]OFFIS - Institute for Informatics, Escherweg 2, Oldenburg, 26121, Germany

[2]Carl von Ossietzky University of Oldenburg, Ammerländer Heerstraße 114-118, 26129, Oldenburg, Germany

[3]Hasso Plattner Institute, Prof.-Dr.-Helmert-Straße 2-3, 14482 Potsdam, Germany

Abstract

This paper presents our research on evaluating the suitability of a Tabular Data Synthesis (TDS) tool for use-case specific requirements. The main goal is to develop a platform that allows users, for example researchers from other fields, to select a suitable TDS tool for their real-world application. In the course of developing such a platform, three contributions are currently planned: Firstly, a decision guide for users formulated by compiling the reported performance of leading tools against a set of functional and non-functional requirements. Secondly, a benchmarking framework for TDS tools based on these identified requirements. Lastly, a customizable tool selection platform, developed through extensive benchmarking of predominant TDS tools. This platform must provide a number of possible tools based on specific use case constraints and allow for community-based expansion, thereby offering a dynamic and adaptable solution for TDS tool selection.

Keywords

Tabular data synthesis, Deep generative models, Artificial data generation, Benchmarking, Customized tool selection

1. Introduction

Tabular data synthesis (TDS) is a method which creates realistic artificial data that mirror the distribution and structure of real relational datasets, and it provides a solution [1] for data scarcity in data-driven applications.

Data synthesis is mainly popular for images [2] and text [3], however TDS has demonstrated impressive outcomes in the generation of highly realistic artificial tables [4, 5, 6]. The goal of this research is to determine how the fitness-for-use of a TDS tool can be assessed, given concrete real-world applications. The main contributions of this research are: **1)** A **decision guide** to select a suitable TDS tool for an application, developed by compiling the reported performance of the predominant tools on our identified functional and non-functional requirements. **2)** A **benchmarking framework for TDS tools**, using our previously identified functional and non-functional requirements as performance indicators. **3)** A **customizable tool selection platform**, developed by benchmarking predominant TDS tools (cf. Contribution 2). The platform is customizable because it outputs a number of suitable tools. The suitability is estimated based on use-case specific constraints. The platform is expandable because it allows community-based updates.

2. Related Work

Our focus is tabular data, which is structured data organized into rows representing individual data points, and columns representing different features. Our work can be classified as a recommendation system for TDS tools. However, to the best of our knowledge, there are no TDS tools' recommender systems.

Platforms such as Synthetic Data Vault (SDV) [7] and its enterprise version DataCebo, Gretel AI [8] and Mostly AI [9] offer the possibility to generate tabular data. They implement some leading TDS models to fit the widest range of applications possible, yet we find there is currently no universal TDS tool which works well for all datasets. These platforms do not report on the specific limitations of their models for each application, partly because there is no standard framework.

Relevant surveys for our work include Hernandez's [10] review for health records, Fan's [11] analysis of Generative Adversarial Networks (GAN) across different data types, Figueira's [12] survey on evaluation methods, Brophy's [13] exploration on time series generation, Koo and Kim's [14] review on generative diffusion models, and Lin's [15] review with focus on time-series diffusion.

3. Purposes of TDS

Reviewing the different surveys and TDS tool papers, we identified five reasons for the synthesis of tabular data.

- **Missing value imputation:** Incomplete entries often occur in real-world datasets, potentially distorting analysis. TDS is employed to fill these gaps with plausible values.

Published in the Proceedings of the Workshops of the EDBT/ICDT 2024 Joint Conference (March 25-28, 2024), Paestum, Italy

maria.davila@offis.de (M. F. Davila)

www.offis.de/offis/person/maria-fernanda-davila-restrepo.html (M. F. Davila)

- **Dataset balancing:** Some classes having significantly more instances than others can cause bias in data-driven models. TDS could balance the dataset by generating records for the underrepresented classes.
- **Dataset augmentation:** Increasing the dataset size by creating new artificial records help, for example, to increase robustness in machine learning models.
- **Privacy protection:** Generating artificial data that adheres to privacy regulations allows secure data sharing while safeguarding sensitive information.
- **Customized data generation:** Generating data with specific external constraints allows the creation of scenario-based data, particularly valuable when original data is unavailable. For example, generating environmental datasets for different future scenarios is essential for data-driven meteorological models [16].

The protection of privacy can be the sole reason for the synthesis (e.g., if sensitive data need to be shared), but it can also be combined with any of the other purposes.

4. Challenges of TDS

For all domains of data synthesis where privacy protection is of interest, one challenge is the *privacy vs. utility trade-off* [17]. Data utility describes the data's effectiveness in fulfilling its intended purpose, besides privacy. Balancing data privacy and utility is a fundamental challenge in data synthesis, because enhancing privacy often diminishes data utility and vice versa [18].

The challenge in TDS is to accurately capture the main information and structure of the input dataset, to be able to replicate it in a synthetic dataset that is useful for the desired purpose. We summarize the challenges of accurately capturing and replicating this information as follows:

- **Handling Missing Values:** Capturing the real column distribution and column correlations in a dataset with missing values is challenging, because the gaps distort statistical properties.
- **Addressing Class Imbalance:** In a dataset with class imbalance the model could over fit, or suffer mode collapse. However, it is crucial to effectively learn from such imbalances to identify and understand outliers, which are often indicative of anomalies [19].
- **Diversity of Column Types:** Different from images composed by pixels and text composed by words and phrases, tabular data often contains various column types, such as numerical, categorical, text, temporal, and mixed.
- **Complex Distributions:** Real-world columns can have complex distributions and capturing the real distribution of a column is crucial to generate realistic data.
- **Complex Column Relations**: Real-world datasets include correlations between columns, and sometimes these columns belong to different tables within the dataset. Capturing and preserving these relations is particularly challenging, for example, when there are multiple tables interconnected through foreign key references [20].
- **Temporal dependencies:** Temporal columns add complexity to the synthesis process. This is particularly difficult for long-term relations because it requires the model to retain information over long periods of time.

5. TDS Tool Requirements

Our focus are deep generative models, a subcategory of data-driven TDS tools, which leverages deep learning techniques to model joint probability distributions of the dataset. Compiling existing surveys shows that the currently predominant tools belong to the following categories: Variational Autoencoders (VAE), Generative Adversarial Networks (GAN), Normalizing Flows (NF), Graph Neural Networks (GNN), Diffusion Probabilistic Models, and Transformers (often LLMs). The sampling method SMOTE is also often included because it achieves good performance for its simplicity [5].

We compiled a list of requirements reported for the different TDS tools, as shown in Table 1.

Table 1
List of characteristics that TDS tools should be able to capture and replicate from a dataset.

Class	Requirements
Column Number	Univariate
	Bivariate
	Multivariate
Column Type	Categorical
	Numerical continuous
	Numerical discrete
	Text
	Temporal
	Mixed
Column Distribution	Typical statistical distributions (Gaussian, uniform, exponential, Poisson, binomial, logistic, etc.)
	Skewed
	Multi-modal
Relations	Inter-column relations
	Inter-table relations
	Temporal relations: short-term
	Temporal relations: long-term
	Integrity constraints

Combining the purposes described in Section 3, the requirements listed in Table 1, and a compilation of TDS tools (Table 3), we mark what requirements each TDS tool was reported to fulfill. As a result, we created a first attempt for a decision guide for users who wants to synthesize tabular data for a real-world application. The result is shown in Table 2.

Table 2

Tabular decision guide to support the selection of TDS tools, based on the compilation of the reported TDS tool fulfillment of functional and non-functional requirements. We ranked the tool's computational costs based on how the author's compared their tools to the others.

	What columns are present in the original dataset?	**What is the purpose of the artificial data?**	**Approaches**	**Competitive Advantages**	**Relative Computational Cost**
Which tabular data synthesis tool is best suited for my application?	Categorical and Numerical	Privacy Protection	medGAN		Low
			DP-GAN	Differential privacy	Low
			PATE-GAN	Differential privacy	Low
		Missing value imputation, dataset balancing or data augmentation	TVAE		Low
			GOGGLE		Medium
			TableGAN	Includes privacy	Low
			CTGAN	Complex distributions	Medium
		Customized generation	GANBLR+		Medium
			CTGAN	Complex distributions	Low
	Categorical, Numerical and Mixed (categorical and numerical)	Privacy	CTAB-GAN		Medium
			CTAB-GAN+	Differential privacy	Medium
			Kamino	Differential privacy	Medium
		Missing value imputation, dataset balancing or data augmentation	TabDDPM	Complex distributions	High
			Kamino	Integrity constraints	Medium
			REaLTabFormer	Inter-table constraints, less pre-processing	High
		Customized generation	CTAB-GAN		Medium
			CTAB-GAN+	Differential privacy	Medium
	Categorical, Numerical and Temporal	Privacy	DoppelGANger		Medium
		Missing value imputation, dataset balancing or data augmentation	TimeGAN		Medium
			TimeVAE		Medium
			TSGM		High
			DoppelGANger	Long-term	Medium
		Customized generation	***RESEARCH GAP***		
	Categorical, Numerical, Mixed and Text	Any	GReaT		High
			REaLTabFormer	Inter-table constraints	High

In the course of creating the guide, we identified some research gaps: **1)** there are no reported tools for customized generation of time series datasets, **2)** many advanced tools are reported to violate integrity constraints [18], **3)** transformer models capture correlations well with significantly less pre-processing but the results seem to not fully capture the complex distributions of single column [6], **4)** there are no tools to effectively generate multiple related tables, preserving the column correlations and integrity constraints, **6)** there is no transparency in reporting the computational costs or resources required.

Table 3

Overview of tools considered for our assessment, grouped by their corresponding methodology

Methodology	**Approaches**
Sampling	SMOTE [21]
GAN	medGAN [22], PATE-GAN [23], DP-GAN[24], TableGAN [17], CTGAN[25], CTAB-GAN[26], CTAB-GAN+ [4], GANBLR+[27] TimeGAN[28], DoppelGANger[29]
VAE	TVAE[25], TimeVAE[30]
Diffusion	TabDDPM[5], TSGM [31]
Graph NN	Goggle[32]
Transformer	GReaT [6], REalTabFormer [33]
Probabilistic	KAMINO [18]

6. Research Question

The main goal is to develop a platform that allows users to select a suitable TDS tool for their use-case specific requirements. This divides into the research questions:

RQ1 What are the use-case specific functional and non-functional **requirements** that drive the selection process between TDS tools?

RQ2 How can the suitability of a specific tool be evaluated based on the identified requirements using a standardized **benchmarking** framework? Which metrics can be used to assess the performance of the tools?

RQ3 How can users be effectively **guided** in their decision process of a suitable TDS tool?

The main research focus is the development of a framework to benchmark TDS tools. Each of the research questions result in a contribution as follows:

C1 A **decision guide** for users choosing a suitable TDS tool for their application, developed by compiling the reported TDS tool fulfillment of functional and non-functional requirements.

C2 A **benchmarking framework for TDS tools**, based on our previously identified functional and non-functional requirements.

C3 A **customizable tool selection platform**, developed by benchmarking predominant TDS tools (cf. Contribution 2).

7. Conclusions

Synthetic tabular data is a solution for the scarcity and lack of diversity of real-world datasets for data-driven applications. From the research gaps we identified, our focus is addressing how to evaluate the suitability of TDS tools for use-case specific requirements.

We successfully identified the main purposes, challenges and some of the requirements for TDS tools, however the decision-making process has so many dimensions that it cannot be adequately presented as a table. For that reason, we aim to break down the decision-making process involved in selecting a TDS tool for real-world applications into functional and non-functional requirements. Examples of functional requirements is the ability to handle multiple column types and distributions. Examples of non-functional requirements is the resource efficiency of the tool (time and memory), or its scalability. Afterwards, those requirements will base a benchmarking framework for TDS tools, used to evaluate the tools' suitability for specific use cases. Finally, a customizable tool selection platform will be developed to capture the full complexity of the decision-making process and truly guide users in selection a suitable tool for their real-world application.

The planned contributions add to the research field by providing a baseline to benchmark TDS tools, allowing researchers to easily identify gaps. This broadens the research space from machine learning researchers to other data-centric fields, such as data management. The contributions also bring TDS closer to users outside this research field, by simplifying the tool selection process.

References

[1] K. H. L. Minh, K. H. Le, Airgen, RTSI (2021). doi:10.1109/RTSI50628.2021.9597364.

[2] OpenAI, Dall-e: Creating images from text (2021).

[3] OpenAI-ChatGPT, Version 4 (2023).

[4] Z. Zhao, A. Kunar, R. Birke, L. Y. Chen, Ctab-gan+, arXiv.2204.00401 (2022). doi:10.48550/arXiv.2204.00401.

[5] A. Kotelnikov, D. Baranchuk, I. Rubachev, A. Babenko, Tabddpm, arXiv:2209.15421 (2022). doi:10.48550/arXiv.2209.15421.

[6] V. Borisov, K. Seßler, T. Leemann, M. Pawelczyk, G. Kasneci, Language models are realistic tabular data generators, arXiv.2210.06280 (2023). doi:10.48550/arXiv.2210.06280.

[7] N. Patki, R. Wedge, K. Veeramachaneni, The synthetic data vault, DSAA (2016) 399–410. doi:10.1109/DSAA.2016.49.

[8] K. Boyd, Create synthetic time-series data with doppelganger and pytorch, 2022.

[9] Mostly ai, 2023. URL: https://mostly.ai/.

[10] M. Hernandez, G. Epelde, A. Alberdi, R. Cilla, D. Rankin, Synthetic data generation for tabular health records, Neurocomputing 493 (2022) 28–45. doi:10.1016/j.neucom.2022.04.053.

[11] J. Fan, T. Liu, G. Li, J. Chen, Y. Shen, X. Du, Relational data synthesis using generative adversarial networks, Proceedings of the VLDB Endowment 13 (2020) 1962–1975. doi:10.14778/3407790.3407802.

[12] A. Figueira, B. Vaz, Survey on synthetic data generation, evaluation methods and gans, Mathematics 10 (2022) 2733. doi:10.3390/math10152733.

[13] E. Brophy, Z. Wang, Q. She, T. Ward, Generative adversarial networks in time series, ACM Comput. Surv. 55 (2023). doi:10.1145/3559540.

[14] H. Koo, T. E. Kim, A comprehensive survey on generative diffusion models for structured data, arXiv:2306.04139 (2023). doi:10.48550/arXiv:2306.04139.

[15] L. Lin, Z. Li, R. Li, X. Li, J. Gao, A comprehensive survey on generative diffusion models for structured data, arXiv:2305.00624 (2023). doi:10.48550/arXiv:2305.00624.

[16] A. Nandy, C. Duan, H. J. Kulik, Data-driven scenarios of climate change, Resilient Urban Futures. The Urban Book Series (2021). URL: https://doi.org/10.1007/978-3-030-63131-4_13.

[17] N. Park, M. Mohammadi, K. Gorde, S. Jajodia, H. Park, Y. Kim, Data synthesis based on generative adversarial networks, Proceedings of the VLDB Endowment 14 (2018). doi:10.48550/arXiv:1806.03384.

[18] C. Ge, S. Mohapatra, X. He, I. F. Ilyas, Kamino, Proceedings of the VLDB Endowment 14 (2020). doi:10.48550/arXiv.2012.15713.

[19] R. C. Ripan, I. H. Sarker, Outlier detection approach for effectively classifying cyber anomalies., Hybrid Intelligent Systems (2021). URL: https://doi.org/10.1007/978-3-030-73050-5_27.

[20] P. Han, W. Xu, W. Lin, J. Cao, C. Liu, S. Duan, H. Zhu, C3-tgan, TechRxiv (2023). doi:10.36227/techrxiv.24249643.v1.

[21] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, Smote, Journal of Artificial Intelligence Research 16 (2002) 321–357. doi:10.48550/arXiv.1106.1813.

[22] J. Yoon, J. Jordon, M. van der Schaar, Generating multilabel discrete patient records using generative adversarial networks (2017). URL: https://doi.org/10.48550/arXiv.1703.06490.

[23] J. Yoon, J. Jordon, M. van der Schaar, Pate-gan, International Conference on Learning Representations (2019). URL: https://openreview.net/forum?id=S1zk9iRqF7.

[24] L. Xie, K. Lin, S. Wang, F. Wang, J. Zhou, Differentially private generative adversarial network, arXiv.1802.06739 (2018). doi:10.48550/arXiv.1802.06739.

[25] L. Xu, M. Skoularidou, A. Cuesta-Infante, K. Veeramachaneni, Modeling tabular data using conditional gan, Proceedings NEURIPS 32 (2019).

[26] Z. Zhao, A. Kunar, R. Birke, H. V. der Scheer, L. Y. Chen, Ctab-gan, arXiv:2102.08369 (2021). doi:10.48550/arXiv.2102.08369.

[27] Y. Zhang, N. A. Zaidi, J. Zhou, G. Li, Ganblr++, SDM (2022).

[28] J. Yoon, D. Jarrett, M. van der Schaar, Time-series generative adversarial networks, Advances in Neural Information Processing Systems 32 (NeurIPS) (2019).

[29] Z. Lin, A. Jain, C. Wang, G. Fanti, V. Sekar, Using gans for sharing networked time series data, arXiv:1909.13403 (2019). doi:10.48550/arXiv.1909.13403.

[30] A. Desai, C. Freeman, Z. Wang, I. Beaver, Timevae, arXiv:2111.08095 (2021). doi:10.48550/arXiv.2111.08095.

[31] H. Lim, M. Kim, S. Park, N. Park, Regular time-series generation using sgm, arXiv.2301.08518 (2023). doi:10.48550/arXiv.2301.08518.

[32] T. Liu, Z. Qian, J. Berrevoets, M. van der Schaar, Goggle, The Eleventh International Conference on Learning Representations (2023). URL: https://openreview.net/forum?id=fPVRcJqspu.

[33] A. V. Solatorio, O. Dupriez, Realtabformer, arXiv.2302.02041 (2023). doi:10.48550/arXiv.2302.02041.