

Received 27 January 2025, accepted 2 March 2025, date of publication 10 March 2025, date of current version 17 March 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3549680

RESEARCH ARTICLE

Utility Meets Privacy: A Critical Evaluation of Tabular Data Synthesizers

JULIAN HÖLLIG^{ID} AND MICHAELA GEIERHOS^{ID}

Research Institute CODE, University of the Bundeswehr Munich, 85577 Neubiberg, Germany

Corresponding author: Julian Höllig (julian.hoellig@unibw.de)

ABSTRACT Evaluating synthetic data requires careful consideration of both utility and privacy. This study analyzes 12 synthesizers across 17 tabular health datasets, providing large-scale, comparable evaluation results. A novel utility-privacy score integrates a privacy measure into the evaluation, quantifying the trade-off between the two. Membership inference analysis is extended for robust privacy assessment, with reusable code provided for further research. Key findings include: 1) despite its simplicity, SMOTE achieves the best results of all synthesizers, but no single synthesizer consistently outperforms all others across all datasets; 2) utility and privacy are inherently correlated, with improvements in one compromising the other; 3) some synthesizers exhibit greater robustness to variations in data properties, such as sample and feature size, and some properties have a stronger impact than others. These findings underscore how closely the utility of a synthesizer is tied to individual datasets and privacy considerations, and highlight the importance of incorporating these aspects into future research and adopting broader, more diverse evaluation frameworks.

INDEX TERMS Membership inference analysis, privacy evaluation, tabular data synthesizer, utility-privacy trade-off.

I. INTRODUCTION

Synthetic data is increasingly recognized as a viable solution for creating artificial analogs of sensitive datasets to facilitate data exchange for analytical purposes without compromising privacy or violating data protection regulations. The basic premise is that synthetic data does not correspond to real individuals or subjects, thereby mitigating the risk of disclosure of sensitive information. To ensure the utility of synthetic data, its performance must be evaluated by comparing its accuracy on specific analytical tasks to that of the real dataset. The closer the synthetic data's results match those of the real dataset, the greater its utility. Although numerous synthesizers have been developed to achieve this goal, a comprehensive cross-evaluation of these methods is lacking, leaving it unclear which approach is optimal for a given analytical task. Furthermore, the evaluation of privacy disclosure of synthetic datasets is often neglected due to the assumption that sensitive information cannot be reconstructed from synthetic data, despite the

lack of supporting evidence. In this work, we address these research gaps by performing a cross-evaluation of 12 synthesizers with different underlying techniques, many of which have not been previously compared. We introduce a utility-privacy score (**ups**) that reflects both the privacy protection and the analytical performance of synthetic data, allowing for a valid interpretation of synthesizer quality. Utility is evaluated by computing the machine learning performance of models trained on the synthetic data. Privacy is considered here as the strength of resistance to membership inference. We extend an existing membership inference analysis (cf. Section II-B2) by adding new features and provide the implementation for further research. In addition, we provide a statistically significant quantification of the relationship between the utility and privacy of synthetic data. Furthermore, we evaluate the robustness of the synthesizer performance to changing data properties as an additional evaluation dimension and demonstrate its relevance.

The following research questions will be addressed.

RQ 1 Which synthesizer has the best utility-privacy trade-off, and how do utility and privacy affect each other?

The associate editor coordinating the review of this manuscript and approving it for publication was Rongbo Zhu^{ID}.

RQ 2 How is the performance of synthesizers affected by different data properties?

A. PRIVACY PROTECTION

Protecting sensitive data has always been a concern, and it has become even more important as the collection and analysis of digital data continues to grow. The nonprofit Identity Theft Resource Center has seen a steady increase in data compromises in recent years, with a 78% increase in 2023 over 2022 [1]. Legislators have responded with binding privacy guidelines for the collection and processing of sensitive data, such as the General Data Protection Regulation [2], the Health Insurance Portability and Accountability Act [3], and the California Consumer Privacy Act [4]. Sensitive information is defined here as information that can be used to directly or indirectly identify an individual [2]. At the same time, data sharing for analytics, especially machine learning, is becoming a critical aspect of innovative business and social applications. Sensitive data is collected and protected by government, military, intelligence, healthcare, and industrial organizations. In healthcare in particular, machine learning promises game-changing benefits, but privacy concerns challenge data sharing and impede progress by researchers or industry experts [5], [6]. For example, one hospital cannot easily share data about a rare disease to collaborate with a second hospital, slowing the development of potential solutions and preventing lives from being saved [7]. Nor can the same hospital share the data with third-party machine learning experts to benefit from their expertise. The legally compliant way to share data is through data processing agreements [2], but these are complicated and time-consuming, delaying early development of solutions [8].

B. SYNTHETIC DATA GENERATION

In order to comply with the legal framework while efficiently taking advantage of advances in machine learning and data analytics, alternative measures had to be found to make sensitive data exchangeable. One promising solution to this challenge is artificial data synthesis, first introduced by [9]. Kiran and Kumar describe data synthesis as “a technique where the artificial data is generated (synthesized) from the real data so that the privacy elements in the dataset are removed without affecting the statistical properties of the dataset” [10]. The idea is that synthetic data is not real, and therefore does not require privacy, although it is still useful for data analysis and machine learning in particular. Scientists at the Massachusetts Institute of Technology (MIT) support this claim by evaluating the effectiveness of data synthesized from real data for machine learning tasks [11]. For this purpose, several real and corresponding synthetic datasets were distributed to different groups of data science experts who developed machine learning models from the data. In a series of tests, 70% of the time there were no significant differences in performance between models trained on real and synthetic data. Other studies also confirm the usability

of synthetic data for data analysis [11], [12]. This suggests that synthetic data could be an alternative to establishing data processing agreements or, as suggested by [8], could be used for development until an agreement is finalized.

Many data synthesizers have been developed since Rubin’s seminal paper [9], and with the increasing availability of computing resources, new methods are often based on deep generative models for synthesizing high-dimensional data types such as text and images. Tabular data synthesizers are more diverse, using techniques ranging from simple, non-statistical approaches to large, complex deep learning models. In Appendix A, we present previous work that compares the utility of data generated by tabular synthesizers. However, they do not cover the full range of available synthesizers, and often the number of test datasets is too small for an adequate utility evaluation.

C. UTILITY-PRIVACY EVALUATION

The evaluation of the synthesizers in this study integrates both utility and privacy. Utility is evaluated through machine learning performance on various healthcare classification tasks. Each of the real-world datasets is used as the basis for data synthesis and to train a baseline machine learning model that competes with the models trained on synthetic data, resulting in a utility score (**us**). Privacy is assessed by an optimized membership inference algorithm that quantifies the exposure risk of individuals in the real datasets by analyzing the corresponding synthetic dataset, resulting in a privacy score (**pps**). The trade-off between utility and privacy is expressed by calculating the average of their respective scores, forming the **ups**. Fig. 1 shows an overview of the proposed scoring concept.

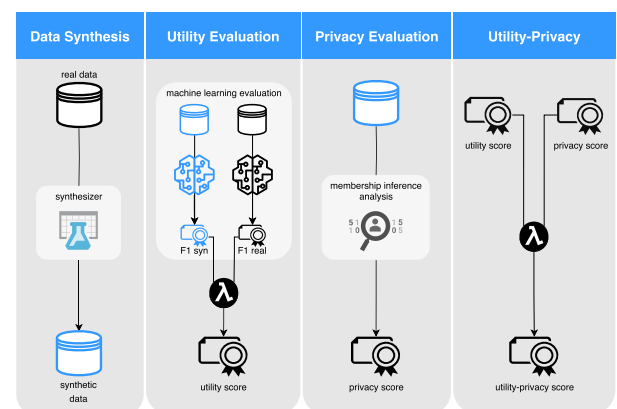


FIGURE 1. Conceptual overview of synthesizer evaluation. Blue colored elements contain or are generated from synthetic data. The lambda functions represent arbitrary functions (for simplicity).

D. OUTLINE

Section II provides an overview of the synthesizers evaluated in the study, explains the rationale behind their selection, and gives the technical details of the evaluation metrics, with a focus on the optimized privacy metric. Section III defines the

experimental setup, including the steps required to reproduce the study, and presents the results. Following a discussion of the findings and limitations of the study in Section IV, supplementary material on related work and descriptions of the individual synthesizers used in this study are given in the Appendices A and B.

II. METHODS

This section presents the synthesizers analyzed in this study and the metrics used to evaluate them. First, we provide an overview of the synthesizers, categorizing them based on their underlying techniques and clarifying their origins. Second, we present the utility and privacy metrics along with our modifications, detailing the calculation and formalizing how they contribute to our utility-privacy metric.

A. SYNTHESIZERS

The synthesizers selected for this study have been developed over the course of 21 years (2002–2023) and thus cover different approaches with different underlying techniques, prioritizing those that have shown superior performance in previous studies. Some of the synthesizers had to be adapted to our experimental requirements, especially with respect to the input datasets and the evaluation framework. Table 1 shows the selected synthesizers and their underlying techniques, the open source tools used for implementation, and the year they were first introduced in a data synthesis publication.

TABLE 1. Data synthesizers evaluated in this paper. Labeled methods (*) required adaptation for this study.

Method	Techniques	Tool	Year
SMOTE	Local Densities	imblearn [13]	2002
kNNMTD*	Local Densities	kNNMTD [7]	2022
MC-GEN*	Local Densities	MCGEN [14]	2023
GaussCop	Statistical	SDV [11]	2016
PrivBayes	Statistical	dpart [15]	2017
CART	Machine Learning	dpart	2005
ARF	Machine Learning	arf [16]	2022
TVAE	Deep Learning	SDV	2019
CTGAN	Deep Learning	SDV	2019
CorGAN*	Deep Learning	cor-gan [17]	2020
DP-HFlow	Deep Learning	pzflow [18]	2022
GReaT	LLM	be-great [19]	2023

The following subsections briefly describe the underlying techniques. Short descriptions of the individual synthesizers can be found in Appendix B.

1) LOCAL DENSITIES

Synthesizers using local densities define clusters within a real dataset and generate synthetic data from these clusters using relatively simple procedures. For example, SMOTE draws a random data point from the nearest neighbors of another data point and creates a synthetic data point along the segment connecting the two.

2) STATISTICAL

Statistical synthesizers estimate data density using statistical models, such as Bayesian networks, and generate synthetic data by sampling from these estimates. Unlike local density synthesizers, these approaches rely on the entire real dataset for density estimation.

3) MACHINE LEARNING

Machine learning synthesizers use traditional machine learning models to estimate data density. For example, CART iteratively trains tree-based models to predict synthetic feature values based on known features.

4) DEEP LEARNING

Deep learning synthesizers follow a similar approach to traditional machine learning synthesizers, but they use generative neural networks that have higher learning capacities, allowing them to model more complex data density distributions. There are several types of generative models for tabular data synthesis, such as autoencoders, normalizing flows, and diffusion models. The most commonly used model type is the Generative Adversarial Network (GAN) [7], [20], [21].

5) LARGE LANGUAGE MODEL (LLM)

LLM-based approaches take advantage of the knowledge embedded in LLMs by integrating tabular data into a text generation framework. Two open source models, GReaT [19] and TabuLa [22], have recently been made available to the research community.

B. EVALUATION METRICS

In previous research, the trade-off between utility and privacy of synthesizers is often not thoroughly investigated (cf. Appendix A). In our evaluation, we assess both utility and privacy, and integrate the results into the utility-privacy score **ups**. The metrics used in the evaluation are explained in the following sections.

1) UTILITY SCORE (US)

Given the increasing importance of machine learning in data analysis, in recent years, the usefulness of synthesizers has often been evaluated by comparing the performance of two machine learning models on a classification task, where one model is trained on synthetic and one on the corresponding real dataset. In this context, the predictions of the models are usually evaluated in terms of accuracy [19], [23], [24], [25] or F1 [19], [21], [26]. We prefer the F1 score for this study because, unlike accuracy, it considers the performance of models across all classes equally. Accuracy can be misleadingly high when evaluating models on class-imbalanced data, as it reflects performance on the majority class more than on the minority classes, even though both are equally important. We use the scikit-learn implementation of

the F1 score.¹ The formula provided in (1) determines the F1 score (**F1**) for each class label by balancing the number of correct predictions (true positives **TP**) against incorrect predictions (false positives **FP**, false negatives **FN**), with the final F1 score being the average across all class labels.

We chose machine learning models with low complexity to evaluate the performance, which allows comparability with the results of related work (cf. Appendix A). The models were all implemented using the scikit-learn library [27] and include Naive Bayes, Decision Tree, Random Forest, Logistic Regression, and k-Nearest Neighbors (kNN).

It is important to relate **F1_s** obtained by the synthetic data model to **F1_r** obtained by the real data model, since the synthetic dataset is derived from the corresponding real dataset, with the goal of approximating its utility. The utility score **us** is the ratio of these F1 scores (cf. to (2)), with values ranging from 0 to 1, as long as **F1_s** does not exceed **F1_r**. Higher values indicate higher utility of the synthetic data.

$$\mathbf{F1} = \frac{2 \times \mathbf{TP}}{2 \times \mathbf{TP} + \mathbf{FP} + \mathbf{FN}} \quad (1)$$

$$\mathbf{us} = \frac{\mathbf{F1}_s}{\mathbf{F1}_r} \quad (2)$$

2) PRIVACY PROTECTION SCORE (PPS)

To evaluate the privacy of the synthesizers, we subject the synthesized datasets to a membership inference analysis similar to the work of [28], which elaborates on the cosine similarity threshold and disclosure risk calculations (cf. Appendix A). The analysis considers a privacy attack scenario wherein an adversary uses a synthetic dataset alongside a public real dataset containing identifiable records, with the objective of inferring membership status within an unknown protected dataset from which the synthetic dataset was originally generated. If the adversary is able to match synthetic and public records with high similarity, it can infer that the identity in the public record corresponds to the individual represented in the protected dataset from which the synthetic record was derived, thereby compromising that individual's privacy. In our experimental study, the training dataset used to generate the synthetic dataset serves as both the public and the protected real dataset. The complete overlap of these datasets poses the greatest risk in the attack scenario that we assume for simplicity. To compare the similarity of two data points, their cosine similarity is computed. To account for categorical features in the cosine similarity calculation, matching categorical entries are both set to one, and mismatching entries are one set to zero and the other set to one. Algorithm 1 describes the membership inference analysis formally, supplemented by explanations of the analysis steps in the following paragraphs. Additionally, a visual overview is provided in Fig. 2.

¹https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html, Accessed: Sep. 20, 2024.

Algorithm 1 Membership Inference Analysis

$\mathcal{R}$ $\triangleright$ real dataset (train + test)
 $\mathcal{T} \subseteq \mathcal{R}$ $\triangleright$ train dataset
 $\mathcal{S}$ $\triangleright$ synthetic dataset
 $\mathcal{M}$ $\triangleright$ set of matches
pr $\triangleright$ precision of matches
re $\triangleright$ recall of matches
 d $\triangleright$ disclosure risk
 θ $\triangleright$ similarity threshold
pps $\triangleright$ privacy protection score
 $P(\mathcal{X} \subseteq \mathcal{R})$ $\triangleright$ preprocess numerical values
 $C(r, \mathcal{X} \subseteq \mathcal{R})$ $\triangleright$ compute nearest neighbor similarity
 $J(s, \mathcal{X} \subseteq \mathcal{R}, \theta)$ $\triangleright$ return matching records
 $U(t, \mathcal{M})$ $\triangleright$ return unique match

Step 1
 $\mathcal{R} \leftarrow P(\mathcal{R})$
 $\mathcal{T} \leftarrow P(\mathcal{T})$
 $\mathcal{S} \leftarrow P(\mathcal{S})$
 $\theta \leftarrow \sum_{r \in \mathcal{R}} C(r, \mathcal{R}) * \frac{1}{|\mathcal{R}|}$
for each $s \in \mathcal{S}$ **do**
 $\mathcal{R}_\theta \leftarrow J(s, \mathcal{R}, \theta)$
 $\theta_s \leftarrow \sum_{r \in \mathcal{R}_\theta} C(r, \mathcal{R}_\theta) * \frac{1}{|\mathcal{R}_\theta|}$
 $\mathcal{T}_{\theta_s} \leftarrow J(s, \mathcal{T}, \theta_s)$
 $d \leftarrow \begin{cases} 0, & \text{if } \mathcal{T}_{\theta_s} = \emptyset \\ \frac{1}{|\mathcal{T}_{\theta_s}|}, & \text{otherwise} \end{cases}$
 $\mathcal{M} \cup \{(s, \mathcal{T}_{\theta_s}, d)\}$
end for
Step 2
 $\mathbf{pr} \leftarrow \sum_d * \frac{1}{|\mathcal{M}|}$ $\triangleright d \in \{d \mid (s, \mathcal{T}_{\theta_s}, d) \in \mathcal{M}\}$
 $\mathcal{M}_u \leftarrow \{U(t, \mathcal{M}) \mid t \in \mathcal{T}\}$
 $\mathbf{re} \leftarrow \sum_d * \frac{1}{|\mathcal{T}|}$ $\triangleright d \in \{d \mid (s, \mathcal{T}_{\theta_s}, d) \in \mathcal{M}_u\}$
 $\mathbf{pps} \leftarrow 1 - (\frac{\mathbf{pr} + \mathbf{re}}{2})$
Step 3
return **pps**

a: COMPUTE DISCLOSURE THRESHOLD (STEP 1)

To identify matches between synthetic and real data points, we need to compute a similarity threshold. The crossing of this threshold simulates a disclosure risk for an individual. Since there are no heuristics for determining an acceptable similarity threshold, it is set based on the average similarity between neighboring records in the original dataset. Let $\mathcal{R} \leftarrow \{r_1, r_2, \dots, r_m\}$ be the entire real dataset consisting of a training ($\mathcal{T} \leftarrow \{t_1, t_2, \dots, t_n \mid t_k \in \mathcal{R}\}$) and test dataset, and $\mathcal{S} \leftarrow \{s_1, s_2, \dots, s_n\}$ is the synthetic dataset. First, the numerical features of all datasets are rescaled to a range between zero and one by a min-max normalization ($P(\mathcal{X} \in \mathcal{R})$). Then, the average similarity between the most similar records in the real dataset is calculated to obtain a global threshold (θ). Second, for each synthetic record (s) from a given synthetic dataset, a subset of real records with similarities to s exceeding

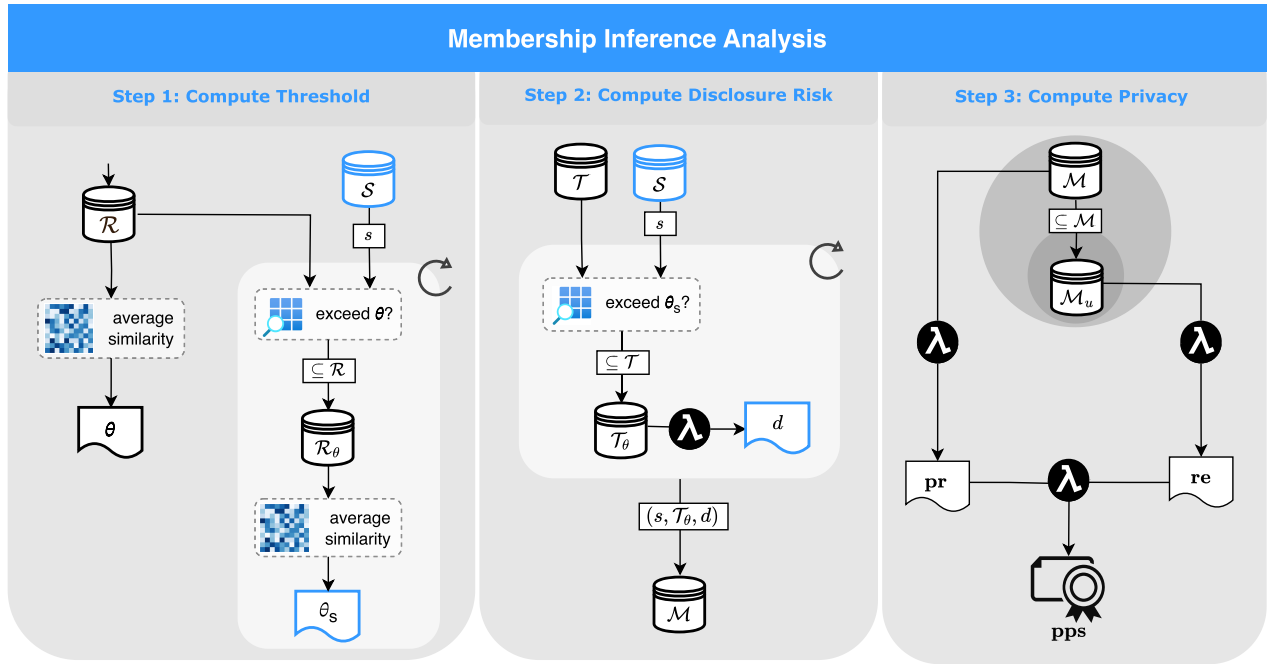


FIGURE 2. Overview of the membership inference analysis. Elements with a blue frame contain or are generated from synthetic data. The lambda symbols represent arbitrary functions (for simplicity). Variables are defined like this: $\mathcal{R}$:= real data; $\mathcal{S}$:= synthetic data; θ := threshold; $\mathcal{T}$:= training data; d := disclosure risk; $\mathcal{M}$:= matches, pr := precision, re := recall, pps := privacy protection score.

the global threshold is identified ($\mathcal{R}_\theta \subseteq \mathcal{R}$). And third, the average similarity between the most similar records of this subset is computed, resulting in the local threshold for the corresponding synthetic record (θ_s). Similarly, a local threshold is determined for each synthetic record. By creating these individual thresholds, differences between local cluster densities in the real dataset are incorporated into the following disclosure risk assessment, resulting in more accurate privacy measurements.

b: COMPUTE MEMBERSHIP DISCLOSURE RISKS (STEP 2)

A membership disclosure risk (d) is identified if the cosine similarity of a specific synthetic record (s) and any training record (t) exceeds the local threshold of s . Hereby, we introduce weighted disclosure risks for each synthetic record based on the number of training records exceeding this threshold ($d \leftarrow \frac{1}{|\mathcal{T}_{\theta_s}|}$). For example, matching one training record represents a disclosure risk value of 1, or 100%, matching five training records represents a 20% risk. The underlying idea is that as the number of candidates increases, the adversary will infer the membership status of a given individual with decreasing certainty, indicating that the disclosure risk of individuals represented in the public records decreases. The resulting disclosure risks and corresponding information are collected in a set of matches ($\mathcal{M} \leftarrow \{(s_1, \mathcal{T}_{\theta_1}, d_1), \dots, (s_k, \mathcal{T}_{\theta_k}, d_k)\}$).

c: COMPUTE PRIVACY PROTECTION SCORE (STEP 3)

To obtain the cumulative disclosure risk introduced by the synthetic dataset, precision (pr) and recall (re) are calculated

over all matches. Precision is defined here as the average disclosure risk over all matches. Recall is determined by the sum of the disclosure risks over all unique matches ($\mathcal{M}_u \subseteq \mathcal{M}$) divided by the number of training records. A unique match refers to exactly one training record; it is the match with the highest disclosure risk among all matches involving that record. For example, if the training record t_1 was contained in multiple matches, the match with the smallest set of training records, and thus the highest disclosure risk, would be considered in the recall calculation. To compute the **pps**, recall and precision are averaged and subtracted from one. The values of the score can range from 0 to 1. A value closer to 1 indicates better privacy protection by the synthetic dataset. Keeping the **pps** in the same range as the **us** will be useful when combining the metrics into a common measure.

3) UTILITY-PRIVACY SCORE (UPS)

The **ups** is defined as the weighted average of **us** and **pps** (cf. to (3)). Since **us** and **pps** values typically range from 0 to 1, **ups** values fall in the same range. A value closer to 1 indicates a better performance of a synthesizer with respect to the trade-off between privacy and utility of the synthesized data. The weights ($\alpha \leftarrow 0.5$, $\beta \leftarrow 0.5$) can be used to increase the influence of either **us** or **pps**, depending on the desired privacy level of the data. Determining a heuristic for the optimal weight configuration may be application specific and could be part of future research. In this study, equal weights are used, although the **us** has slightly more influence on the **pps** in cases where

TABLE 2. Tabular health datasets used for evaluation containing numerical (Num.) and categorical (Cat.) data types.

Dataset Name	# Records	# Features	# Num.	# Cat.	# Classes	# Missing values	Origin
Breast Cancer	286	9	0	9	2	9	UCI
Breast Tissue	106	9	9	0	6	-	UCI
Cardiotocography	2,126	25	20	5	3	-	UCI
Dermatology	366	34	34	0	6	8	UCI
Diabetes	768	8	8	0	2	-	Kaggle
Echocardiogram	132	9	8	1	2	-	UCI
Eye	14,980	9	8	1	2	-	UCI
Heart	297	13	5	8	2	-	Kaggle
Kidney	768	8	8	0	2	1,012	UCI
Lymphography	148	18	0	18	4	-	UCI
Patient	90	8	0	8	3	3	UCI
Retinography	1,152	19	16	3	2	-	UCI
Sani	303	55	21	34	2	-	UCI
Stroke	5,110	11	4	7	2	201	Kaggle
Thoracic Surgery	470	16	3	13	2	-	UCI
Tumor	339	17	0	17	21	225	UCI
Thyroid	9,172	28	7	21	6	-	UCI

the synthetic datasets perform better than the corresponding original datasets, resulting in a **us** greater than 1. If necessary, the **ups** can be customized by adjusting or replacing the utility and privacy measures of **us** and **pps**, as long as their score ranges remain between 0 and 1.

$$\mathbf{ups} = (\alpha \times \mathbf{us}) + (\beta \times \mathbf{pps}) \quad (3)$$

III. EXPERIMENTS

This section describes the experiments conducted to address the research questions of this study. It details the data, preprocessing steps, and experimental setup, followed by an analysis of the results. The implementation code for reproducibility is available on Github.²

A. DATASETS

The datasets used to evaluate the synthesizers are summarized in Table 2. They are structured in a tabular format and are limited to categorical and numerical data types, as the synthesizers are not designed to support non-tabular data formats, such as text or image-based data. Each dataset contains health-related information, making the evaluation particularly relevant to readers in the healthcare field, given the strict constraints imposed by privacy laws and policies in this area [3], [29]. The datasets are publicly available on either Kaggle [30] or UCI [31], have been used in previous data synthesis studies, include class labels, and vary in sample and feature size.

The datasets are available in CSV, Excel and TXT formats, which are converted to Pandas dataframes on loading, automatically extracting the class column. Missing values are replaced: numerical with the median, and categorical with random values based on their frequency in the corresponding feature column. Features with string values, ordered integers, or a small number of unique values (10% or less of total records) are considered categorical features, all other features

are numerical. Unusable features such as dates and IDs are removed. As in previous work [19], [23], [26], we have refrained from normalizing the data, as there is evidence that raw data is preferable to using preprocessed data for synthesizers [24]. For compatibility, string values are integer encoded, and classes with fewer than five records are removed from the datasets. The following synthesizers use class-wise data generation, which requires that the datasets be partitioned based on class labels: SMOTE, DP-HFlow, Gaussian Copula Synthesizer (GaussCop), MC-GEN, TVAE, CTGAN, CorGAN, CART, PrivBayes.

B. EXPERIMENTAL SETUP

Each dataset is randomly divided into train and test sets of equal size. While this results in less training data for the synthesizers, it ensures that sufficient data is available for evaluation even when working with small datasets. The training set is used for data synthesis, machine learning model training, and privacy analysis. The test set is used to evaluate the performance of the machine learning models. To account for random variation between different synthesis runs, ten independent random splits are drawn, resulting in ten synthesis runs and ten evaluations for each synthesizer-dataset combination. Additional splits were avoided to maintain feasible experimental run times. Each evaluation returns a single **us**, **pps** and **ups** averaged over all ten splits. The standard deviation of the resulting **ups** over ten splits was 5.9%.

Each synthesizer is evaluated based on all available datasets. For a fair utility evaluation, the corresponding real and synthetic datasets have the same size. In most cases, the synthesizers are applied with the hyperparameters given in the corresponding library (cf. to references in Table 1). The batch sizes of all synthesizers based on deep learning techniques (DP-HFlow, CTGAN, TVAE, CorGAN, and GReaT) have been set to eight, because the default batch sizes exceed the sample size of at least one dataset,

²https://github.com/juhoUnibw/syn_data

which is likely to lead to poor training performance. The privacy budgets of the synthesizers that integrate differential privacy (PrivBayes, CART, and MC-GEN) were set to 0.5 to balance performance and privacy of the synthetic data. For those synthesizers that only process continuous data (MC-GEN and CorGAN), the encoded categorical features of each dataset are internally converted to floats and reconverted after synthesis.

All machine learning models (Naive Bayes, kNN, Decision Tree, Random Forest, and Logistic Regression) are applied with the hyperparameters defined in the corresponding scikit-learn libraries, since hyperparameter tuning does not significantly affect the performance of models trained on synthetic data [24]. The models are trained separately on the synthetic and real datasets, resulting in five ‘real’ and five ‘synthetic’ models with five F1 scores each. The average real and synthetic F1 scores determine the utility score of a synthetic dataset as defined in (2). Then, the membership inference analysis produces the **pps** (cf. Algorithm 1).

The experiments were run on an Nvidia GPU RTX 4090 with 24GB RAM, on Debian 12.2.0, using CUDA 12.7. The source code for the experimental pipeline and almost all synthesizers are run with Python 3.11.2, except for MC-GEN (Java (OpenJDK) 17.0.13) and ARF (R 4.2.2). A list of the Python libraries can be found in the requirements file of the Github repository.

C. RESULTS AND ANALYSIS

In this section, we present the experimental results and analyze them with respect to the research questions posed in this study. For RQ 1, the goal is to determine which synthesizer performs best in terms of utility and privacy metrics, and to explore the correlation between these two aspects. We first compare the **ups**, **us**, and **pps** scores obtained by all synthesizers to identify those with the most consistent and superior performance. We then examine the interplay between **us** and **pps** and quantify the strength of their relationship. For RQ 2, the focus is on assessing how the performance of a synthesizer varies with different dataset properties. Therefore, we stratify the datasets based on certain characteristics, such as sample size, and analyze the variations in **ups** across these groups.

1) SYNTHESIZER PERFORMANCES AND UTILITY-PRIVACY TRADE-OFF

Table 3 shows a summary of the evaluation results for each synthesizer, averaged over all splits and datasets. SMOTE achieves the best utility-privacy trade-off (**ups**) and the highest utility (**us**), where the performance of the synthetic data even exceeds that of the original. CorGAN ensures the best privacy protection (**pps**) in our experiments. However, the results vary significantly between datasets (cf. to standard deviations in column ‘std(ups)’), suggesting that they should be interpreted as indicative. The average **ups** standard deviation is 9.7% (**us**=18.8%, **pps**=10.9%), so the ranking of synthesizers can change on unseen datasets. This observation

is further supported by the number of datasets where each synthesizer achieved the highest **ups** (# best), demonstrating that no synthesizer consistently outperforms the others across all datasets.

TABLE 3. Evaluation results averaged over all datasets. The best results are highlighted.

Synthesizer	ups	us	pps	F1_s	# best	std(ups)
SMOTE	0.88	1.02	0.73	0.65	3	0.07
CART	0.85	0.71	0.99	0.44	3	0.09
CTGAN	0.84	0.95	0.73	0.60	2	0.06
MC-GEN	0.84	0.86	0.82	0.56	3	0.12
PrivBayes	0.84	0.73	0.95	0.47	2	0.10
TVAE	0.82	1.00	0.64	0.63	1	0.06
GaussCop	0.82	0.92	0.72	0.57	1	0.09
DP-HFlow	0.81	0.80	0.81	0.52	1	0.09
ARF	0.77	0.80	0.74	0.52	1	0.14
kNNMTD	0.74	0.99	0.49	0.63	0	0.07
GReaT	0.73	0.74	0.71	0.46	0	0.12
CorGAN	0.72	0.44	1.00	0.27	0	0.15

Analyzing the relationship between **us** and **pps** highlights the critical importance of incorporating privacy measures in the evaluation of synthetic data. Synthesizers such as SMOTE, kNNMTD, and TVAE produce data with perfect or even superior utility compared to the real data. However, relatively low **pps** values have been returned for these synthesizers, indicating a trade-off between utility and privacy. Fig. 3 shows a scatter plot of all **(us, pps)** value pairs across datasets and synthesizers. A polynomial regression curve fitted to these values shows a clear trend: privacy decreases as utility increases. This observation is quantitatively validated by Pearson’s correlation coefficient (cf. to (4), where $x = \mathbf{us}$ and $y = \mathbf{pps}$) from SciPy,³ yielding a value of -0.53 (p-value = $6.82\text{e-}16$). This result shows a moderate to strong negative correlation between **us** and **pps**. In other words, increased utility in synthetic data often comes at the expense of decreased privacy for individuals in the original dataset.

$$r = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{\sum (x - \bar{x})^2 \sum (y - \bar{y})^2}} \quad (4)$$

2) INFLUENCE OF VARYING DATA PROPERTIES ON SYNTHESIZERS

To gain a deeper understanding of the synthesizers, we analyze how their performance is affected by four data properties: sample size, feature size, data type, and class size. For each property, the datasets are divided into two groups based on a specific condition. For sample and feature size, the nine smallest datasets are assigned to one group, and the remaining datasets form the other group. The data type property classifies the datasets based on the majority data type (numeric versus categorical) within each dataset. The class size property divides the datasets based on the number of classes (two classes versus more than two classes). Except

³<https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html#scipy.stats.pearsonr>, Accessed: Dec. 18, 2024.

TABLE 4. Average ups for all data property groups, which are defined as follows: Smpl:= sample sizes; Feat:= feature sizes; Num or Cat:= dominance of numerical or categorical features; Class:= class sizes. The arrows (↓ or ↑) represent lower or higher values.

Synthesizer	Smpl↓	Smpl↑	Feat↓	Feat↑	Num	Cat	Class↓	Class↑
SMOTE	0.87	0.88	0.87	0.89	0.88	0.88	0.86	0.92
CART	0.84	0.85	0.90	0.79	0.86	0.84	0.88	0.77
CTGAN	0.84	0.85	0.83	0.86	0.90	0.83	0.83	0.89
MC-GEN	0.83	0.85	0.89	0.79	0.90	0.83	0.87	0.76
PrivBayes	0.86	0.82	0.88	0.79	0.91	0.82	0.84	0.83
TVAE	0.82	0.82	0.81	0.83	0.81	0.82	0.81	0.86
GaussCop	0.79	0.86	0.79	0.85	0.79	0.83	0.80	0.88
DP-HFlow	0.82	0.79	0.84	0.77	0.84	0.80	0.82	0.77
ARF	0.70	0.84	0.80	0.73	0.77	0.77	0.81	0.67
kNNMTD	0.70	0.78	0.75	0.73	0.73	0.74	0.74	0.75
GReaT	0.71	0.75	0.80	0.64	0.70	0.74	0.76	0.65
CorGAN	0.68	0.76	0.76	0.67	0.69	0.73	0.77	0.59
AVG	0.79	0.82	0.83	0.78	0.82	0.80	0.82	0.78

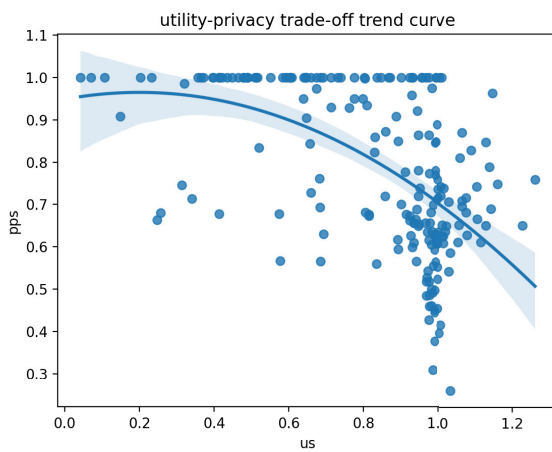


FIGURE 3. Scatter plot and regression curve of all (us, pps) value pairs, showing the relationship between utility and privacy of synthetic data.

for the data type property, the resulting groups are of similar size. Table 4 shows the average ups per group for each data property. Group differences of 5% or more are highlighted. The comparison shows that MC-GEN, GaussCop, ARF and CorGAN are particularly affected by different data properties. For example, MC-GEN's performance decreases by 11% on average for datasets with more than two classes, and ARF's performance increases by 14% for datasets with larger sample sizes. In terms of the overall impact of data properties on synthesizer performance, feature and class size properties have the most significant impact. On average, they cause a performance difference of 4.5%. In contrast, other synthesizers achieve fairly consistent performance regardless of the data property, in particular TVAE, SMOTE, and kNNMTD.

Fig. 4 compares the robustness of each synthesizer, based on the average ups difference between data property groups. The resulting values are subtracted from 1 and shown in percent, with higher values representing more robust synthesizers. When evaluating utility, robustness and privacy, SMOTE achieves the best overall results among the synthesizers evaluated, despite using a much simpler data generation technique than most other synthesizers.

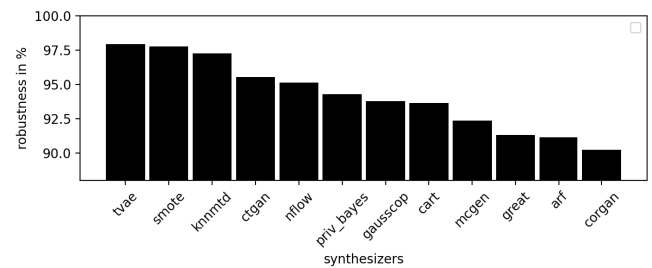


FIGURE 4. Degree of robustness of each synthesizer to varying data characteristics. Higher values indicate greater robustness.

IV. CONCLUSION

In this study, we conduct a comprehensive and progressive evaluation of tabular data synthesizers, focusing on the critical trade-off between machine learning utility and privacy protection. By analyzing the performance of 12 synthesizers on 17 health datasets, we provide a large-scale comparison that provides a robust evaluation benchmark. Our novel contribution lies in the integration of a privacy measure into the evaluation process, introducing a utility-privacy score (ups) that quantifies the balance between these competing goals. In addition, we extend a membership inference analysis algorithm [28] by incorporating further privacy risk considerations, and contribute reusable code for future research.

The goal of RQ 1 was to identify the best synthesizer in terms of the utility-privacy trade-off, and to explore the correlation between utility and privacy. Interestingly, one of the simplest and oldest synthesizers, SMOTE, emerged as the best performer by achieving the highest average ups. This finding underscores that complexity alone does not guarantee superior results. However, no single synthesizer consistently outperformed the others, and results varied significantly across the datasets. The results show that the selection of the optimal synthesizer is closely tied to the individual dataset and task. Regarding the interplay between utility and privacy, we provide quantitative evidence for a clear negative correlation, emphasizing the relevance of privacy measures as part of the evaluation process.

The goal of RQ 2 was to investigate the dependence of synthesizer performance on different data properties. Our analysis shows that performance is significantly affected by variations in feature and class sizes. In this context, we propose the robustness of the synthesizers against varying data properties as an additional evaluation parameter.

Limitations of this study include the lack of a qualitative assessment of privacy, which leaves ambiguity about when an adversary can confidently identify a member. The study does not evaluate the utility and privacy of synthetic data beyond machine learning performance and membership inference, nor does it address other quality dimensions such as realism, statistical fidelity, or fairness. The research is limited to tabular health datasets with categorical and numeric characteristics, and does not guarantee consistent applicability to other data types, domains or utility analyses.

Further research could expand efforts to improve privacy protection and thereby foster greater confidence in the protection of individual data. This can be achieved by incorporating more sophisticated and diverse privacy attack scenarios into the evaluation process. Ideally, privacy evaluation metrics should be aligned with the strategies used by data snoopers to compromise sensitive information, requiring regular updates and re-evaluations. Rather than setting an absolute threshold to distinguish between protected and unprotected data, it may be beneficial to introduce different levels of privacy that correspond to different regulatory standards. In this context, future studies could develop heuristics by exploiting the weights in the **ups** calculation (cf. Section II-B3). It may also be useful to examine the correlation between privacy-enhancing model parameters and privacy measures, for example between different privacy budgets in differentially private models and the **pps**. This would lead to a more precise determination of the required privacy protection. Regarding the choice of optimal synthesizers for given datasets and analytical goals, a decision framework could help to make the appropriate choice. Specifying performance differences for different data properties, as proposed in this study, introduces a potential selection criterion.

APPENDIX A RELATED WORK

A literature review of synthetic tabular data evaluation studies shows that data synthesis is often motivated by the goals of preventing privacy disclosure and improving data accessibility and shareability. In existing studies, synthesizers are evaluated by applying different utility measures along with some privacy considerations. The following sections provide an overview of utility measures for synthetic data evaluation, privacy protection approaches, and the range of existing synthesizers.

A. UTILITY MEASURES

Drechsler and Reiter [32] identify two types of data utility measures: narrow and broad. Broad measures evaluate statistical differences between real and synthetic data distributions.

Narrow measures evaluate differences between task-specific models derived from real versus synthetic data. Broad measures that are typically used for data utility evaluation are the propensity score [24], [26], [33], the Pairwise Correlation Distance (PCD) [7], [22], [26], [34], and the Kullback-Leibler Divergence (KLD) [34], [35]. Widely used narrow measures are the Confidence Interval Overlap (CIO) [33], [35], [36], [37], and machine learning performance metrics, which are used to compare models trained on synthetic versus real data. In the context of tabular data, mostly simple machine learning models have been used for this purpose, such as linear/logistic regression models [20], [24], [26], SVM [22], [25], [38], kNN [23], [38], Naive Bayes [23], Decision Trees [19], [20], [22], Random Forests [7], [19], [22], and Boosting models [20], [25]. These models are usually evaluated using either the accuracy [7], [24], [25] or the F1 score [22], [24], [38].

Both broad and narrow measures have been used in several studies. Often a combination of both is used [7], [21], [23], but individual use of either broad [12], [33], [36] or narrow measures [20], [25], [38] is also common. With TabSynDex, Chundawat et al. [39] have recently introduced a single utility metric that combines several broad measures with machine learning effectiveness, which may provide a more robust evaluation. To understand the scope and value of individual utility metrics, it is worth exploring whether different utility measures are correlated. For example, if metrics A and B have a strong correlation, then measuring one of the metrics may be sufficient to evaluate the utility of synthetic data. Of particular note are correlations between broad measures and machine learning performance metrics, which would suggest that statistical similarity between real and synthetic data ensures comparable results for certain analytics tasks. However, limited research has been conducted in this area, with sobering results. The correlations observed between different broad utility metrics are weak or absent [26]. For the correlation between broad measures and machine learning performance, a weak correlation was found for the propensity score [24]. For correlations between narrow measures, Khan et al. found a strong negative correlation between CIO and machine learning performance [35]. Although it is useful to identify a single, common metric to measure synthetic data utility for each analytics task, overall there is little evidence of strong correlations for specific tasks, making the utility of broad utility measures questionable. Therefore, we narrow the evaluation in our study to one specific analysis (classification tasks) to provide a more targeted assessment of synthetic data utility.

B. PRIVACY PROTECTION

A key quality of synthetic data that is often overlooked or taken for granted is its level of privacy, although privacy is a prerequisite for sharing sensitive data. Therefore, in addition to evaluating the utility of synthetic data, privacy should be a key consideration. Several studies evaluate privacy in the context of synthetic data generation. Some

TABLE 5. Overview of synthesizers grouped by underlying techniques.

Synthesizer	Key components	Synthesis approach	Reference
Local densities			
SMOTE	kNN	Data synthesis through computing linear combinations between the k-nearest neighbors (kNN) of real data points.	[40]
kNNMTD	kNN, MTD	Estimation of local domain ranges in the data by applying Mega-Trend-Diffusion (MTD) to the k-nearest neighbors, resulting in synthetic data through diffusion.	[7]
MC-GEN	GMM	Clustering of the data on sample and feature level. Data synthesis from these clusters using a differentially private Gaussian Mixture Model (GMM).	[25]
Statistical			
GaussianCopula	Copula	Estimation of joint data distribution by capturing the dependencies between variables and their marginal distributions through a multivariate Gaussian Copula. Data synthesis by sampling from the joint distribution.	[11]
PrivBayes	Bayesian network	Approximation of data distribution by modeling conditional probabilities between variables, with noise injection for low-dimensional marginals. Data synthesis through sampling from this distribution.	[41]
Machine learning			
CART	Decision trees	Recursive data splits by decision trees to create homogeneous subsets for all features of the real data. Based on known features, CART iteratively predicts synthetic values for the unknown features by sampling from the density estimations of these subsets.	[42]
ARF	Random Forests	Recursive training of random forests by alternating between generating synthetic data and discriminating real from synthetic data (inspired by GAN method). Synthetic data is sampled from local densities of the tree leaves, estimated using kernel density estimation or maximum likelihood estimation.	[20]
Deep learning			
TVAE	Variational autoencoders	Encoding of tabular data into a latent space with means and variances. Data synthesis by decoding random data samples from this latent space.	[43]
CTGAN	Conditional GAN	The generator uses a conditional vector (masked vector conditioned on random feature value) and random noise to synthesize data, which is compared to real data in the training process.	[43]
CorGAN	GAN, autoencoder	Combination of GAN and autoencoder to capture feature inter-correlations in the real data in the training process. Data synthesis through inference of random vector.	[28]
DP-HFlow	Normalizing flow	Training of mapping simple distributions into complex data distributions through a series of invertible transformations. Data synthesis by applying the inverse transformations on Gaussian noise to receive realistic samples.	[21]
LLM			
GReaT	Transformer	Text-encoding of tabular feature values to enable the use of LLMs. Data synthesis by predicting masked values from the encoded data using a GPT-2-based transformer.	[19]

of these implement privacy protection parameters directly in the synthesizers [12], [21], [25], while one study applies them to the synthetic data [38] to ensure that it is not a replica of the original. Other studies attempt to measure the privacy level of the ‘final’ synthetic data [19], [28], [34], [44], [45]. Borisov et al. [19] analyze the distance to the closest record distribution between real and synthetic datasets to that between real training and test datasets. More precise results can be obtained through membership inference analysis, which evaluates the relationship between individual synthetic and real records to assess potential disclosure risks. For this purpose, Park et al. [45] use the Global Uniqueness (GU) method, which identifies disclosure risks based on equal quasi-identifiers between two records. Goncalves et al. [34] and Torfi and Fox [28] identify a disclosure risk when the cosine similarity between two records exceeds a certain threshold, assuming an identifier-independent attack scenario. The threshold is randomly drawn from a Gaussian with a mean of 0.5 and a standard deviation of 0.01. In our study, we refine

this approach (cf. Section II-B2) by computing similarity thresholds based on the distribution and local cluster densities of each dataset. In addition, we introduce gradual disclosure risks by considering the probability that two matching records lead to a membership inference.

Ensuring reliable privacy may require qualitative reviews and testing against a variety of attack scenarios, which is beyond the scope of existing studies. However, privacy measures should be considered in synthetic data evaluations, as observations from smaller studies suggest that privacy may come at the expense of data utility [23], [25], [44]. Therefore, we integrate our privacy measure as part of a utility-privacy score (cf. Section II-B3) and provide evidence for the correlation between utility and privacy through a comprehensive analysis.

C. SYNTHESIZERS AND EVALUATION

There are papers that focus on the development of new synthesizers [7], [19], [25], as well as those that improve existing ones [22], [37]. In addition, many focus on the

TABLE 6. Results (ups/us/pps) for all dataset-synthesizer pairs, averaged across ten splits.

Datasets	TVAE	GaussianCopula	CTGAN	ARF	DP-HFlow	kNNMTD	MC-GEN	CorGAN	SMOTE	PrivBayes	CART	GReaT
ups												
Breast Cancer	0.86	0.89	0.90	0.80	0.88	0.72	0.95	0.68	0.81	0.87	0.94	0.94
Breast Tissue	0.80	0.80	0.87	0.74	0.98	0.75	0.92	0.54	0.99	0.98	0.87	0.70
Cardiotocography	0.81	0.97	0.94	0.85	0.75	0.71	0.70	0.60	0.88	0.80	0.75	0.55
Kidney	0.75	0.75	0.74	0.92	0.91	0.72	0.92	0.71	0.89	0.78	0.79	0.57
Dermatology	0.79	0.82	0.80	0.47	0.87	0.70	0.80	0.55	0.85	0.87	0.74	0.53
Diabetes	0.80	0.74	0.74	0.80	0.83	0.73	0.80	0.82	0.86	0.78	0.86	0.75
Retinography	0.77	0.90	0.96	0.80	0.75	0.72	0.84	0.82	0.84	0.95	0.89	0.66
Echocardiogram	0.86	0.68	0.87	0.75	0.80	0.73	0.93	0.71	0.81	0.80	0.84	0.74
Heart	0.74	0.63	0.74	0.76	0.74	0.68	0.81	0.72	0.86	0.87	0.82	0.77
Lymphography	0.77	0.82	0.83	0.79	0.78	0.72	0.83	0.71	0.75	0.80	0.80	0.69
Patient	0.79	0.83	0.85	0.81	0.92	0.65	0.88	1.01	0.87	0.98	1.05	0.82
Stroke	0.83	0.84	0.82	0.78	0.81	0.76	0.81	0.73	0.88	1.00	0.93	0.80
Thoracic Surgery	0.79	0.81	0.82	0.80	0.87	0.70	0.88	0.90	0.86	0.69	0.98	0.78
Thyroid	0.99	0.97	0.96	0.84	0.65	0.92	0.85	0.74	0.96	0.62	0.79	0.94
Tumor	0.89	0.83	0.85	0.46	0.63	0.65	0.53	0.52	0.92	0.86	0.70	0.53
Sani	0.89	0.79	0.80	0.72	0.80	0.73	0.83	0.72	1.01	0.68	0.82	0.69
Eye	0.84	0.90	0.82	0.97	0.76	1.00	1.00	0.76	0.88	0.92	0.82	0.94
us												
Breast Cancer	1.12	1.07	1.08	0.92	1.07	0.99	1.16	0.37	1.04	0.89	0.93	1.23
Breast Tissue	0.95	1.00	0.98	0.48	0.96	0.99	0.85	0.07	0.98	0.97	0.73	0.40
Cardiotocography	0.99	0.93	0.87	0.69	0.49	1.01	0.83	0.20	1.02	0.60	0.51	0.41
Kidney	0.98	0.97	0.98	1.00	0.95	0.98	0.98	0.41	0.99	0.55	0.64	0.58
Dermatology	0.98	0.98	0.97	0.26	0.81	0.98	0.60	0.11	0.95	0.74	0.49	0.34
Diabetes	0.98	0.81	0.81	0.92	0.98	1.00	0.95	0.65	1.00	0.93	0.78	0.81
Retinography	0.93	0.80	0.92	0.94	0.66	0.99	0.91	0.64	0.99	0.90	0.78	0.69
Echocardiogram	0.95	0.52	0.86	0.89	0.94	0.98	0.94	0.42	0.97	0.94	0.67	0.49
Heart	0.98	0.68	0.89	0.97	0.97	0.99	0.98	0.44	1.01	0.80	0.64	0.99
Lymphography	0.94	0.94	0.95	0.92	0.65	0.98	0.83	0.41	0.94	0.61	0.60	0.68
Patient	1.03	1.05	1.06	0.99	1.14	0.99	1.08	1.01	1.03	0.96	1.15	1.01
Stroke	1.01	1.02	1.00	1.00	1.00	1.00	1.00	0.47	1.00	1.01	0.87	1.00
Thoracic Surgery	0.99	1.02	1.02	0.99	1.00	1.00	0.99	0.80	1.00	0.37	0.98	0.99
Thyroid	1.13	1.07	1.09	0.69	0.32	0.99	0.76	0.48	1.14	0.23	0.58	1.06
Tumor	1.13	1.02	1.08	0.25	0.57	1.03	0.15	0.04	1.11	0.71	0.40	0.31
Sani	1.11	0.86	0.90	0.68	0.59	0.97	0.66	0.45	1.26	0.36	0.65	0.66
Eye	0.83	0.89	0.71	0.93	0.51	0.99	1.00	0.52	1.01	0.84	0.67	1.00
pps												
Breast Cancer	0.61	0.71	0.72	0.68	0.70	0.45	0.75	1.00	0.59	0.85	0.96	0.65
Breast Tissue	0.65	0.61	0.77	1.00	1.00	0.50	1.00	1.00	1.00	1.00	1.00	1.00
Cardiotocography	0.63	1.00	1.00	1.00	1.00	0.42	0.56	1.00	0.74	1.00	1.00	0.68
Kidney	0.52	0.53	0.50	0.85	0.86	0.46	0.86	1.00	0.78	1.00	0.95	0.57
Dermatology	0.61	0.66	0.63	0.68	0.93	0.43	1.00	1.00	0.74	1.00	1.00	0.71
Diabetes	0.62	0.67	0.68	0.67	0.68	0.45	0.65	1.00	0.72	0.63	0.95	0.68
Retinography	0.60	1.00	1.00	0.66	0.84	0.45	0.78	1.00	0.70	1.00	1.00	0.63
Echocardiogram	0.78	0.83	0.87	0.62	0.66	0.49	0.92	1.00	0.66	0.66	1.00	1.00
Heart	0.49	0.57	0.59	0.54	0.52	0.38	0.64	1.00	0.72	0.95	1.00	0.55
Lymphography	0.61	0.69	0.72	0.66	0.90	0.46	0.82	1.00	0.57	1.00	1.00	0.69
Patient	0.54	0.61	0.65	0.63	0.69	0.31	0.68	1.00	0.71	1.00	0.96	0.62
Stroke	0.65	0.65	0.64	0.55	0.63	0.52	0.61	1.00	0.76	1.00	1.00	0.61
Thoracic Surgery	0.59	0.61	0.63	0.61	0.74	0.40	0.77	1.00	0.71	1.00	0.97	0.57
Thyroid	0.85	0.87	0.83	1.00	0.98	0.85	0.93	1.00	0.79	1.00	1.00	0.81
Tumor	0.65	0.64	0.63	0.66	0.68	0.26	0.91	1.00	0.74	1.00	1.00	0.75
Sani	0.67	0.72	0.70	0.76	1.00	0.49	1.00	1.00	0.76	1.00	1.00	0.73
Eye	0.86	0.91	0.93	1.00	1.00	1.00	1.00	1.00	0.74	1.00	0.97	0.89

utility analysis of pre-existing synthesizers [26], [34], [38]. The synthesizers evaluated exhibit different techniques for data synthesis, ranging from multiple imputation [24], [26], [33], [35], [36], [37], Bayesian models [23], [34], [38], and copulas [23], [24], [26], to GAN-based approaches [7], [20],

[21], and LLMs [19], [22]. Commonly evaluated synthesizers include CART [24], [26], [33], [35], [36], [38], [44], DataSynthesizer [23], [26], [38], linear/logistic regression approaches [33], [35], [38], GaussianCopula [23], [24], [26], and CTGAN [19], [21], [22]. In studies comparing the

utility of different synthesizers, CART often achieves the best results [24], [26], [35], [38], [44]. However, it is uncertain whether universal conclusions can be drawn, as different studies use different experimental setups. Existing work compares at most six synthesizers in the same setup [20], [34]. For example, CART has not yet been compared to recently developed, promising synthesizers such as MC-GEN [25], kNNMTD [7], ARF [20], DP-HFlow [21], GReaT [19], or TabuLa [22]. Furthermore, the datasets used to evaluate synthesizers vary from paper to paper, which also affects the evaluation results. For example, Benaïm et al. [12] report that the utility of synthetic data is higher when derived from larger rather than smaller samples. In existing studies, the number of datasets used in the evaluation ranges from one [36] to 19 [26], the number of records from 48 [7] to 581,000 [20], and the number of features from four [24] to 148 [7]. Moreover, the datasets are drawn from different domains, such as health [7], [12], [19], finance [20], [21], [25], and sociodemographic data [23], [35], [36]. In our work, we use a comprehensive collection of health datasets that have mostly been used to evaluate synthesizers in various existing studies. We evaluate a larger number of synthesizers than previously studied, and compare them under consistent experimental settings.

APPENDIX B EVALUATED SYNTHESIZERS AND INDIVIDUAL RESULTS

This section briefly describes each of the open-source synthesizers used in this study. The main components, approaches, and references for further details are listed in Table 5.

The collective analysis of all synthesizers through the same evaluation pipeline required minor adaptations in the CorGAN, MC-GEN, and kNNMTD implementations, which are described below.

As part of the synthesis process, CorGAN uses an autoencoder model with convolutional layers, applying filters of specified sizes (kernel sizes) that extract features by ‘sliding’ over the data with defined steps. In the open source implementation, these were designed to replicate experiments from Torfi and Fox’s paper [28]. We adapted the implementation to allow dynamic adaptation of stride and kernel sizes based on the feature size of any tabular dataset, allowing evaluation with our pipeline. In addition, we added a function to round categorical feature values of synthetic datasets to the nearest integer, since the original CorGAN implementation does not re-discriminate categorical features after encoding.

The MC-GEN implementation includes its own preprocessing script (in Java) for dataset splitting and preparation. We have replaced this script with the standard preprocessing from our evaluation pipeline. To enable data synthesis for each dataset, sample, feature, and class size hyperparameters are extracted from a dynamically created text file specific to that dataset within the evaluation pipeline.

In the kNNMTD implementation, the diffusion of real data values results in multiple synthetic values for each class, which are randomly sampled to create the final synthetic dataset. These samples were set to be of equal size, resulting in balanced class distributions. We have modified this part of the code to ensure that the synthetic datasets are sampled proportionally to the original class distribution.

The main part of the paper reports the average scoring results across all 17 datasets. While this allows for more general conclusions than reporting dataset-specific results, readers may still be interested in the latter. The experimental results for **ups**, **us** and **pps** for all dataset-synthesizer pairs are given in Table 6.

REFERENCES

- [1] Identity Theft Resource Center. (2024). *Identity Theft Resource Center 2023 Annual Data Breach Report Reveals Record Number of Compromises; 72 Percent Increase Over Previous High*. Identity Theft Resource Center. Accessed: Apr. 6, 2024. [Online]. Available: <https://www.idtheftcenter.org/post/2023-annual-data-breach-report-reveals-record-number-of-compromises-72-percent-increase-over-previous-high/>
- [2] European Union. (2016). *REGULATION (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons With Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC (General Data Protection Regulation)*. Official J. European Union. [Online]. Available: <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679&qid=1736273550791>
- [3] “Standards for privacy of individually identifiable health information,” *Dept. Health Hum. Services, Office of the Secretary, Federal Register*, vol. 65, no. 250, pp. 82462–82829, Dec. 2000. Accessed: Apr. 5, 2024. [Online]. Available: <https://www.hhs.gov/sites/default/files/ocr/privacy/hipaa/administrative/privacyrule/prdecember2000all8parts.pdf>
- [4] R. Bonta. (2024). *California Consumer Privacy Act (CCPA)*. Accessed: Apr. 5, 2024. [Online]. Available: <https://oag.ca.gov/privacy/ccpa>
- [5] T. M. Persons and J. M. McGinnis. (2019). *Artificial Intelligence in Health Care: Benefits and Challenges of Machine Learning in Drug Development*. United States Government Accountability Office Nat. Acad. Med., Washington, DC, USA. [Online]. Available: <https://www.gao.gov/assets/gao-20-215sp.pdf>
- [6] K. Lekadi, G. Quaglio, A. T. Garmendia, and C. Gallin, “Artificial intelligence in healthcare, applications, risks, and ethical and societal impacts,” *Eur. Parliamentary Res. Service, Panel Future Sci. Technol.*, Brussels, Belgium, Tech. Rep. QA-07-22-328-EN-N, 2022. [Online]. Available: [https://www.europarl.europa.eu/RegData/etudes/STUD/2022/729512/EPRS_STU\(2022\)729512_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/STUD/2022/729512/EPRS_STU(2022)729512_EN.pdf)
- [7] J. Sivakumar, K. Ramamurthy, M. Radhakrishnan, and D. Won, “Synthetic sampling from small datasets: A modified mega-trend diffusion approach using K-nearest neighbors,” *Knowl.-Based Syst.*, vol. 236, Jan. 2022, Art. no. 107687, doi: [10.1016/j.knsys.2021.107687](https://doi.org/10.1016/j.knsys.2021.107687).
- [8] B. Howe, J. Stoyanovich, H. Ping, B. Herman, and M. Gee, “Synthetic data for social good,” 2017, *arXiv:1710.08874*.
- [9] D. B. Rubin, “Statistical disclosure limitation (SDL),” *J. Official Statist.*, vol. 9, no. 2, pp. 461–468, 1993.
- [10] A. Kiran and S. Kumar, “Synthetic data and its evaluation metrics for machine learning,” in *Information Systems for Intelligent Systems*, C. So-In, N. D. Londhe, N. Bhatt, and M. Kitsing, Eds., Singapore: Springer, Mar. 2023, pp. 485–494, doi: [10.1007/978-981-19-7447-2_43](https://doi.org/10.1007/978-981-19-7447-2_43).
- [11] N. Patki, R. Wedge, and K. Veeramachaneni, “The synthetic data vault,” in *Proc. IEEE Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Montreal, QC, Canada, Oct. 2016, pp. 399–410, doi: [10.1109/DSAA.2016.49](https://doi.org/10.1109/DSAA.2016.49).
- [12] A. R. Benaïm, R. Almog, Y. Gorelik, I. Hochberg, L. Nassar, T. Mashiah, M. Khamaisi, Y. Lurie, Z. S. Azzam, J. Khoury, D. Kurnik, and R. Beyar, “Analyzing medical research results based on synthetic data and their relation to real data results: Systematic comparison from five observational studies,” *JMIR Med. Informat.*, vol. 8, no. 2, Feb. 2020, Art. no. e16492, doi: [10.2196/16492](https://doi.org/10.2196/16492).

- [13] G. Lemaître, F. Nogueira, and C. K. Aridas, “Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning,” *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 559–563, Jan. 2017.
- [14] M. Li. (2023). *MCGEN-Private-Synthetic-Data-Generator*. Accessed: Sep. 1, 2023. [Online]. Available: <https://github.com/mingchenli/MCGEN-Private-Synthetic-Data-Generator>
- [15] S. Mahiou, K. Xu, and G. Ganey. (2023). *DPART | Differentially Private Auto-Regressive Tabular*. Accessed: Sep. 1, 2023. [Online]. Available: <https://pypi.org/project/dpart/>
- [16] D. Watson, M. N. Wright, J. Kapar, L. Burk, and K. Blesch. (2023). *ARF*. Accessed: Dec. 30, 2024. [Online]. Available: <https://github.com/bips-hb/arf>
- [17] A. Torfi and E. A. Fox, “CorGAN: Correlation-capturing convolutional generative adversarial networks for generating synthetic healthcare records,” 2020, *arXiv:2001.09346*.
- [18] J. F. Crenshaw, Z. Yan, and V. Doster. (2024). *Jfcrenshaw/Pzflow: V3.1.3*. Accessed: Mar. 1, 2024. [Online]. Available: <https://zenodo.org/records/10710271>
- [19] V. Borisov, K. Seßler, T. Leemann, M. Pawelczyk, and G. Kasneci, “Language models are realistic tabular data generators,” in *Proc. 11th Int. Conf. Learn. Represent.*, Jan. 2022. [Online]. Available: <https://openreview.net/forum?id=cEymQNOel>
- [20] D. Watson, K. Blesch, J. Kapar, and M. N. Wright, “Adversarial random forests for density estimation and generative modeling,” in *Proc. 26th Int. Conf. Artif. Intell. Statist.*, vol. 206, Apr. 2023, pp. 5357–5375.
- [21] J. W. Lee, M. Kim, Y. Jeong, and Y. Ro, “Differentially private normalizing flows for synthetic tabular data generation,” in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2022, vol. 36, no. 7, pp. 7345–7353, doi: [10.1609/aaai.v36i7.20697](https://doi.org/10.1609/aaai.v36i7.20697).
- [22] Z. Zhao, R. Birke, and L. Chen, “TabuLa: Harnessing language models for tabular data synthesis,” 2023, *arXiv:2310.12746*.
- [23] M. Hittmeir, A. Ekelhart, and R. Mayer, “On the utility of synthetic data: An empirical evaluation on machine learning tasks,” in *Proc. 14th Int. Conf. Availability, Rel. Secur.*, vol. 29, Aug. 2019, pp. 1–6.
- [24] F. K. Dankar and M. Ibrahim, “Fake it till you make it: Guidelines for effective synthetic data generation,” *Appl. Sci.*, vol. 11, no. 5, p. 2158, Feb. 2021, doi: [10.3390/app11052158](https://doi.org/10.3390/app11052158).
- [25] M. Li, D. Zhuang, and J. M. Chang, “MC-GEN: Multi-level clustering for private synthetic data generation,” *Knowl.-Based Syst.*, vol. 264, Mar. 2023, Art. no. 110239, doi: [10.1016/j.knsys.2022.110239](https://doi.org/10.1016/j.knsys.2022.110239).
- [26] F. K. Dankar, M. K. Ibrahim, and L. Ismail, “A multi-dimensional evaluation of synthetic data generators,” *IEEE Access*, vol. 10, pp. 11147–11158, 2022, doi: [10.1109/ACCESS.2022.3144765](https://doi.org/10.1109/ACCESS.2022.3144765).
- [27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.
- [28] A. Torfi and E. A. Fox, “CorGAN: Correlation-capturing convolutional generative adversarial networks for generating synthetic healthcare records,” in *Proc. 33rd Int. Florida Artif. Intell. Res. Soc. Conf.*, Jan. 2020, pp. 335–340.
- [29] M. Naveed, E. Ayday, E. W. Clayton, J. Fellay, C. A. Gunter, J.-P. Hubaux, B. A. Malin, and X. Wang, “Privacy in the genomic era,” *ACM Comput. Surv.*, vol. 48, no. 1, pp. 1–44, Aug. 2015, doi: [10.1145/2767007](https://doi.org/10.1145/2767007).
- [30] *Your Machine Learning and Data Science Community*. Accessed: Jul. 21, 2024. [Online]. Available: <https://www.kaggle.com/>
- [31] R. L. M. Kelly and K. Nottingham. *The UCI Machine Learning Repository*. Accessed: Jul. 23, 2024. [Online]. Available: <https://archive.ics.uci.edu>
- [32] J. Drechsler and J. P. Reiter, “Disclosure risk and data utility for partially synthetic data: An empirical study using the German IAB establishment survey,” *J. Off. Statist.*, vol. 25, no. 4, pp. 589–603, Jan. 2009.
- [33] B. Nowok. (Oct. 2015). *Utility of Synthetic Microdata Generated Using Tree-based Methods*. UNECE Statist. Data Confident. Work Session. United Nations Economic Commission for Europe (UNECE). [Online]. Available: https://unece.org/fileadmin/DAM/stats/documents/ece/ces/ge.46/20150/Paper_33_Session_2_-_Univ_Edinburgh_Nowok_.pdf
- [34] A. Goncalves, P. Ray, B. Soper, J. Stevens, L. Coyle, and A. P. Sales, “Generation and evaluation of synthetic patient data,” *BMC Med. Res. Methodol.*, vol. 20, no. 1, p. 108, May 2020, doi: [10.1186/s12874-020-00977-1](https://doi.org/10.1186/s12874-020-00977-1).
- [35] M. S. N. Khan, N. Reje, and S. Buchegger, “Utility assessment of synthetic data generation methods,” 2022, *arXiv:2211.14428*.
- [36] J. Taub, M. Elliot, and J. W. Sakshaug, “The impact of synthetic data generation on data utility with application to the 1991 UK samples of anonymised records,” *Trans. Data Privacy*, vol. 13, no. 1, pp. 1–23, Apr. 2020.
- [37] J. Drechsler, S. Bender, and S. Rässler, “Comparing fully and partially synthetic datasets for statistical disclosure control in the German IAB establishment panel,” *Trans. Data Privacy*, vol. 1, no. 3, pp. 105–130, Dec. 2008.
- [38] D. Rankin, M. Black, R. Bond, J. Wallace, M. Mulvenna, and G. Epelde, “Reliability of supervised machine learning using synthetic data in health care: Model to preserve privacy for data sharing,” *JMIR Med. Informat.*, vol. 8, no. 7, Jul. 2020, Art. no. e18910, doi: [10.2196/18910](https://doi.org/10.2196/18910).
- [39] V. S. Chundawat, A. K. Tarun, M. Mandal, M. Lahoti, and P. Narang, “A universal metric for robust evaluation of synthetic tabular data,” *IEEE Trans. Artif. Intell.*, vol. 5, no. 1, pp. 300–309, Jan. 2024, doi: [10.1109/TAL.2022.3229289](https://doi.org/10.1109/TAL.2022.3229289).
- [40] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.
- [41] J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao, “PrivBayes: Private data release via Bayesian networks,” *ACM Trans. Database Syst.*, vol. 42, no. 4, pp. 1–41, Dec. 2017, doi: [10.1145/3134428](https://doi.org/10.1145/3134428).
- [42] J. P. Reiter, “Using CART to generate partially synthetic public use microdata,” *J. Off. Statist.*, vol. 21, pp. 209–235, Jan. 2005.
- [43] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, “Modeling tabular data using conditional GAN,” in *Adv. Neural Inf. Proc. Syst.*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, Eds., Örebro, Sweden: Vancouver, BC, Canada: Curran Associates, Dec. 2019, p. 7335. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/254ed7d2de3b23ab10936522dd547b78-Paper.pdf
- [44] J. Drechsler and J. P. Reiter, “An empirical evaluation of easily implemented, nonparametric methods for generating synthetic datasets,” *Comput. Statist. Data Anal.*, vol. 55, no. 12, pp. 3232–3243, Dec. 2011.
- [45] D.-Y. Park and I.-Y. Ko, “An empirical study of utility and disclosure risk for tabular data synthesis models: In-depth analysis and interesting findings,” in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2024, pp. 67–74, doi: [10.1109/bigcomp60711.2024.00020](https://doi.org/10.1109/bigcomp60711.2024.00020).



JULIAN HÖLLIG received the B.A. degree in technical communication from the University of Applied Sciences Karlsruhe, Germany, in 2018, and the M.S. degree in computational linguistics from LMU Munich, Germany, in 2020. Since 2020, he has been a Research Assistant with the Chair of Data Science, University of the Bundeswehr Munich, Germany. He has contributed to several studies on abusive language detection models and authored an article on semantic text segment classification in technical documents. His current work focuses on data synthesis for machine learning applications.



MICHAELA GEIERHOS is currently a Professor in data science with the Department of Computer Science, Institute for Data Security, and the Technical Director of the Research Institute CODE, University of the Bundeswehr Munich, Germany. Previously, she was a Professor in digital cultural studies and an Assistant Professor in information systems with Paderborn University, Germany. Her research interests include the practical application of natural language processing in business scenarios, operating at the intersection of computational linguistics, and computer science. Her key areas include developing algorithms for semantic information processing and enabling human-machine communication through interaction via information systems, such as natural language search and question-answering systems. She has received numerous awards for her research and teaching.

...