Enabling Synthetic Data adoption in regulated domains

Giorgio Visani  
DISI Department, University of Bologna  
& CRIF S.p.A., R&D department  
Bologna, Italy  
giorgio.visani@unibo.it

Giacomo Graffi  
CRIF S.p.A., R&D department  
Bologna, Italy

Mattia Alfero  
CRIF S.p.A., R&D Department  
Bologna, Italy

Enrico Bagli  
CRIF S.p.A., R&D department  
Bologna, Italy

Federico Chesani  
DISI Department  
University of Bologna  
Bologna, Italy

Davide Capuzzo  
CRIF S.p.A., R&D department  
Bologna, Italy

Abstract—The switch from a Model-Centric to a Data-Centric mindset is putting emphasis on data and its quality rather than algorithms, bringing forward new challenges. In particular, the sensitive nature of the information in highly regulated scenarios needs to be accounted for. Specific approaches to address the privacy issue have been developed, as Privacy Enhancing Technologies. However, they frequently cause loss of information, putting forward a crucial trade-off among data quality and privacy. A clever way to bypass such a conundrum relies on Synthetic Data: data obtained from a generative process, learning the real data properties. Both Academia and Industry realized the importance of evaluating synthetic data quality: without all-round reliable metrics, the innovative data generation task has no proper objective function to maximize. Despite that, the topic remains under-explored. For this reason, we systematically catalog the important traits of synthetic data quality and privacy, and devise a specific methodology to test them. The result is DAISYnt (aDoption of Artificial Intelligence SYnthesis): a comprehensive suite of advanced tests, which sets a de facto standard for synthetic data evaluation. As a practical use-case, a variety of generative algorithms have been trained on real-world Credit Bureau Data. The best model has been assessed, using DAISYnt on the different synthetic replicas. Further potential uses, among others, entail auditing and fine-tuning of generative models or ensuring high quality of a given synthetic dataset. From a prescriptive viewpoint, eventually, DAISYnt may pave the way to synthetic data adoption in highly regulated domains, ranging from Finance to Healthcare, through Insurance and Education.

Index Terms—synthetic data, benchmarks, goodness evaluation, data utility, privacy, finance, data sharing

I. INTRODUCTION

Critical aspects of a valuable dataset are data quality and privacy. The former is stressed in the Data-Centric mindset pioneered by Andrew Ng, while the latter is required by novel regulations such as the GDPR [1] and the U.S. FERPA [2] and HIPAA [3], educational and medical data privacy respectively. Privacy Enhancing Technologies [4] already help protecting sensitive data, at the cost of an information loss. In fact, privacy and data quality behave as two antagonistic features. A clever way to potentially avoid such conflict relies on Synthetic Data: data obtained from a generative process, learning real data properties [5]. The quest for valuable synthetic data is highly relevant in regulated domains such as Finance [5] and Healthcare [6], where they may enable several use-cases such as: i) enforcing privacy protection, ii) facilitating data sharing among companies and towards the research community, iii) tackling class imbalance (eg. fraud detection), iv) increasing the amount of data for prediction models. Despite that, the assessment of synthetic data quality and privacy remains an under-explored, although vital, topic. Whilst few taxonomies and tests have been proposed, we feel the need for a decisive improvement.

In this paper we tackle the open question of how to evaluate the quality and privacy of tabular synthetic data. Firstly, we systematically catalog their most important features into three concepts: Statistical Similarity, Data Utility and Privacy. To measure these notions, we devise appropriate state-of-the-art tests yielding a numeric value in the range [0, 1], where higher metrics imply better performance. The final result is DAISYnt (aDoption of Artificial Intelligence SYnthesis): a comprehensive and easy to use test suite, that sets a de facto standard for synthetic data evaluation. As a practical use-case, a variety of generative algorithms have been trained on real-world Credit Bureau Data. The best model has been assessed, using DAISYnt on the different synthetic replicas. Further possible DAISYnt applications entail auditing and fine-tuning of the models or ensuring high quality of a given synthetic dataset. Moreover, from a prescriptive viewpoint, DAISYnt may pave the way to synthetic data adoption in real-world applications concerning highly regulated scenarios.

The present paper does not aim at improving the state-of-the-art of generative models, rather i) it reviews the most important properties of synthetic data, ii) proposes a structured methodology to test them in detail, on any tabular generated dataset, iii) provides an implementation of the test suite, by means of the DAISYnt python package.
In the following, Section II contains taxonomy and literature review. Section III is dedicated to general purpose tests, while Sections IV, V and VI respectively concern with distribution similarity, data utility and privacy tests. Section VII incorporates DAISYnt application on Credit Scoring data, while Section VIII includes a discussion on its implications and future perspectives. Methodological sections contain DAISYnt graphs and results on the Adult dataset from the UCI repository [7]. In it, the task is to predict whether a given adult makes more than 50.000$ a year based on attributes such as education, weekly working hours, etc.

II. RELATED WORK

Consider to have two different datasets, \( D_{train} \) and \( D_{test} \) coming from the same Data Generating Process (DGP). \( D_{train} \) is used to fit a generative model \( G \) with the capability of producing a new dataset, \( D_{synth} \). To avoid any bias, \( D_{test} \) will act as the benchmark dataset. Previous approaches to synthetic data evaluation are well summarized into three concepts:

- \( D_{synth} \) shall retain all the statistical properties of the original data, i.e. \( D_{test} \).
- Data Utility: Prediction models built respectively on \( D_{train} \) and \( D_{synth} \) shall be the most similar as possible. **Model comparison is carried out on \( D_{test} \).**
- Privacy of the \( D_{train} \) individuals and new ones (emulated by \( D_{test} \)) shall be guaranteed.

DAISYnt pipeline and datasets used in each test class are highlighted in Fig. 1.

Hereafter, we review statistical similarity and data utility, while the privacy topic, has its own dedicated Section VI. In [8], the authors compare the two correlation matrices calculated respectively on \( D_{train} \) and \( D_{synth} \), in order to check whether the pairwise linear correlation of variables pairs is preserved. The two matrices are compared element-wise using an aggregate performance metric, namely MAE. In [6], the correlation matrix comparison is performed either, while the similarity of univariate distributions is assessed using the Kolmogorov-Smirnov (KS) two-sample test. In [9] correlation is calculated by means of a particular \( \phi \) coefficient (enabling its usage on both categorical and continuous variables).

More involved statistical similarity tests consist in comparing the distributions of the original and synthetic data. The simplest scenario is to compare the univariate distributions of each variable, as done in [10]. Unfortunately, the comparison relies on visualization techniques of the univariate histogram distributions, without providing any similarity metric. Multivariate distributions are taken into account in [11], where the authors devise three separate metrics to evaluate distribution similarity, analysing the support of the \( D_{train} \) and \( D_{synth} \) full multivariate distributions. In [12] the authors propose to employ the Total Variation Distances (TVD) as a more refined method to compare distributions, but in order to cope with the curse of dimensionality, they use TVD only on a number of low-dimensional multivariate distributions (considering just few variables).

Regarding the data utility property, a widely popular assessment method is “Train on Synthetic, Test on Real” (TSTR) method [13], whose intuition is to train supervised machine learning models on both \( D_{train} \) and \( D_{synth} \) and compare their accuracy on \( D_{test} \). In this vein, the Synthetic Ranking Agreement (SRA) [14] has been proposed for classification datasets, where different prediction models are tested in a TSTR fashion and the discrepancy in classification rankings is recorded. High discrepancy yields low SRA values. Eventually, the Synthetic Data Vault (SDV) [15] provides a sandbox for open-source generative models and related quality metrics. This is the widest open-source test suite to date, incorporating tests related to all the categories described above, although we feel that the underlying methodology needs to be updated with more recent testing methods.

III. GENERAL COMPARISON TESTS

In many business scenarios, data is solely used for descriptive statistics. The following tests check \( D_{synth} \) usefulness for these general-purpose, yet very popular, tasks.

A. Pairwise Correlation

Pairwise correlation is a well-known technique to measure the strength of the association between two variables. The test compares the pairwise Spearman correlation matrices \( R^T, R^S \), respectively obtained on the \( D_{test} \) and \( D_{synth} \) datasets. Spearman retrieves a good amount of non-linear correlations, while it behaves as Pearson coefficient for linear dependence. However, it requires numerical variables. To this end, we encode categorical variables using the CatBoost encoding [16], i.e. an improved version of the Target Encoding which provides guarantees about no information leakage. The technique retains the relationship among the variable to be encoded and the target, while it may distort associations with other independent variables. To compensate that, the Catboost Encoder is trained on the \( D_{test} \) and its mapping is applied to \( D_{synth} \), ensuring the same distortion on both datasets, i.e. preserving their similarity.

Matrix similarity is evaluated through a rescaled version of the Frobenius norm [17]:

\[
d_{corr}(R^T, R^S) = 1 - \frac{\text{tr}\{R^T R^S\}}{\|R^T\|_F \|R^S\|_F} = 1 - \frac{<R^T, R^S>_F}{\|R^T\|_F \|R^S\|_F}
\]
The Frobenius inner product concatenates the rows of the matrix and computes the euclidean inner product between the two vectors. Intuitively, it consists of element-wise comparison of the two matrices (represented in Fig. 2 for the Adult dataset).

B. Predictive Power comparison

The Information Value (IV) [18] measures the strength of the association between each variable and the target, commonly employed in Credit Scoring for feature selection. It requires categorical variables, so the continuous ones have been binned according to the deciles of their marginal distribution. We compute two IV vectors $IV^T, IV^S$ on $D_{test}$ and $D_{synth}$ respectively, and measure their similarity by means of Pearson linear correlation:

$$\rho(IV^T, IV^S)$$

High values ensure that $D_{synth}$ maintains the same variables ranking and relative distance between values, while specific IV numbers might differ.

An example is given in Fig. 3.

IV. DISTRIBUTIONS COMPARISON TESTS

The most powerful tool to describe a dataset is the multivariate distribution: it encodes all the variables information, such as moments, single feature marginal distribution and multivariate distribution for groups of features, pointing out correlations and interactions.

Having considered to estimate the $D_{test}, D_{synth}$ (pdf) and subsequently compare them, we recognize this is not a viable solution. In fact, parametric estimation is not easily generalizable since it requires domain knowledge for the distribution choice, while non-parametric methods do not provide pdf formulas for the comparison. Therefore, we opted for discrepancy tests between the two pdf, without explicitly estimating them. Moreover, since we deal with finite datasets, multivariate tests may fail. Thus, we detach univariate and multivariate distribution comparison. This ensures accurate results on the low-dimensional tests while it provides a theoretically solid framework to perform high-dimensional testing, even being aware they could fail due to a huge number of features.

A. Univariate Distributions

There are no proper techniques working well for both categorical and continuous features. Convenience solutions entail to convert continuous variables into categorical, through binning methods, at the cost of losing information (we lose granularity coercing continuous variables with many different values into a fixed number of bins). Conversely, it is difficult to convert categorical features to continuous without distorting the multivariate distribution. We keep the two comparisons separated, in order to employ more powerful tests.

1) Univariate Distribution - Continuous variables: Consider the generic continuous variable $X_{(k)}$: $X^T_{(k)}$ ($X^S_{(k)}$) values in $D_{test}$ ($D_{synth}$) are drawn from the $f^T_{(k)}$ marginal distribution, while $X^S_{(k)}$ ($X^T_{(k)}$) values in $D_{synth}$ ($D_{test}$) stem from $f^S_{(k)}$. The test goal is to assess whether the null hypothesis $H_0: f^T_{(k)} = f^S_{(k)}$ holds.

Distributions are fully characterized by the collection of their moments: abstract values that can be approximated using real-world data, obtaining quantities such as mean, variance, kurtosis etc. Hence, an intuitive approach is to search for differences in the $X^T_{(k)}, X^S_{(k)}$ moments. To aid the intuition, specific moments discrepancies between two sample distributions are shown in Fig. 4.

Regardless of which moment is involved, it is possible to distinguish the two distributions by choosing a proper $q$ function and evaluating its expectation. Two equal distributions shall obtain similar expected values for each valid $q$. As an example, difference in variance (Figure 4c) can be grasped using $q(x) = x^2$. 

![Fig. 2: $R^T, R^S$ matrices for the Adult dataset](image1)

![Fig. 3: $IV^T, IV^S$ vectors for the Adult dataset](image2)
Maximum Mean Discrepancy (MMD) [19] looks for the $q$ maximizing the difference between the two expectations:

$$\text{MMD}[\mathcal{Q}, X_T^{(k)}, X_S^{(k)}] := \sup_{q \in \mathcal{Q}} \left( \frac{1}{m} \sum_{i=1}^{m} q(x_T^{(k),i}) - \frac{1}{n} \sum_{j=1}^{n} q(x_S^{(k),j}) \right)$$

(1)

Here $m,n$ respectively stand for the $D_{test}, D_{synth}$ sample size, while expectations have been replaced by the sample means. Under mild regularity conditions (Q contains only continuous, bounded, smooth functions), MMD is guaranteed to be different from 0 only when $f_T^{(k)} \neq f_S^{(k)}$ are truly different. However, the superior operator requires to screen all the valid $q \in \mathcal{Q}$, making Equation 1 impractical to compute. An efficient MMD formulation involves kernel functions:

$$\text{MMD}[\mathcal{Q}, X_T^{(k)}, X_S^{(k)}] = \left[ \frac{1}{m^2} \sum_{i,j=1}^{m,n} k(x_T^{(k),i}, x_T^{(k),j}) + \frac{2}{mn} \sum_{i,j=1}^{m,n} k(x_T^{(k),i}, x_S^{(k),j}) + \frac{1}{n^2} \sum_{i,j=1}^{m,n} k(x_S^{(k),i}, x_S^{(k),j}) \right]^{\frac{1}{2}}$$

The usual choice for $k(\cdot, \cdot)$ is the Gaussian kernel, thanks to its ability to screen infinitely large function spaces. In our implementation of the test, we take advantage of the heuristic provided in [20] to choose the proper value for $\sigma$ (the only hyper-parameter for the Gaussian kernel).

Eventually, we estimate the distribution of MMD under the Null Hypothesis $\mathcal{H}_0$, using a permutation test [20]. It essentially consists of randomly partitioning the data $D_{test} \cup D_{synth}$ into $D_{test}$ and $D_{synth}$ (under $\mathcal{H}_0$ the two samples come from the same distribution) and computing the MMD. Repeating it many times, we obtain a numerical approximation of its distribution. With it, we calculate the acceptance interval (considering a 5% type I error), through which we establish whether the variable $X^{(k)}$ passed the univariate similarity test. In Fig. 4 the MMD test is applied to the “Age” variable of the Adult dataset: in 5a an illustrative histogram of the $D_{test}$ and $D_{synth}$ distributions for the chosen variable, along with the proper $q$ function, while in 5b confidence bounds for $\mathcal{H}_0$ acceptance and actual value of the MMD test are displayed.

While MMD is one of the most powerful tests for distribution similarity, it does not scale well for large datasets (due to kernels). In DAISYnt the test runs on $D_{test}$ and $D_{synth}$ sub-samples, whose size can be set by the user to match its computational and hardware requirements. An additional option for extremely large datasets, is to bin continuous variables using deciles and perform the faster categorical distribution testing, at the cost of a coarser result.

2) Univariate Distribution - Categorical variables: The MMD based test is no use for categorical variables, therefore DAISYnt incorporates a classic Chi-Square two-sample test. The null Hypothesis $\mathcal{H}_0$ “no significant difference in the $X^{(k)}$ class frequencies between $D_{test}$ and $D_{synth}$” has formula:

$$\sum_{k=1}^{C} \left( \frac{f_T^{(k),c} - f_S^{(k),c}}{f_T^{(k),c}} \right)^2 \sim \chi^2_{(n-1)\times(C-1)}$$

where $c$ shuffles through the different $X^{(k)}$ classes, $f_T^{(k),c}$, $f_S^{(k),c}$ are the class frequencies in $D_{test}$, $D_{synth}$. Under $\mathcal{H}_0$, the test exhibits a $\chi^2$ distribution with degrees of freedom $df = (n-1)\times(C-1)$, which allows to control the type I error (DAISYnt default is 5%). Eventually, it is important to check whether the assumptions are met, in particular the test is meaningful only if at least 80% of the classes have an expected frequency (in this case $f_T^{(k),c}$) greater than 5 and no class has 0 frequency in $f_S^{(k)}$ [21]. Following these prescriptions, DAISYnt do not test variables not complying with such criteria.

To obtain a unified metric, DAISYnt applies the proper test to categorical and continuous variables separately and calculates the fraction of accepted trials.

B. Multivariate Distributions

As already stated, high dimensionality and mixed data are notable challenges for two-sample distribution tests. To this end, feature selection is employed and categorical and continuous variables are considered separately. As a minor limitation, DAISYnt cannot discover any difference in distribution for groups of mixed variables.
Fig. 5: MMD Test on the Age variable. Here the test failed, meaning that Age distribution is different in $D_{test}$ and $D_{synth}$.

1) Multivariate Distribution - Continuous Variables: Based on empirical experiments, we deduce that MMD greatly suffers the curse of dimensionality. Hence, we restrict the similarity test to the subset of meaningful variables only, namely the relevant ones for a given prediction task. Boruta [22] is the method of choice, thanks to its stability, reliability and reasonable computational time.

Boruta improves the Tree-based feature importance, known to be particularly unreliable in presence of correlation and interactions among variables [23]. The method creates “shadow features” (variables with no relationship with the target variable) and iteratively generates Random Forest models, computing confidence intervals for the feature importance scores. Variables with significantly higher feature importance than any shadow feature are the relevant ones. Repeated trials ensure statistical stability of the procedure, while the choice of Random Forest models, which do not require any fine tuning, guarantees an acceptable computation time.

DAISYnt relies on Boruta to extract the set of numerical meaningful features and runs the MMD test on it. The test is much likely to fail whenever one of the important variables had already failed the univariate test. Moreover, even if all the single variables passed the univariate test, the multivariate one may fail due to differences in correlations and interactions. Substantially, the multivariate comparison is more challenging but, in case of acceptance, gives strong evidence of distribution similarity.

2) Multivariate Distribution - Categorical Variables: We propose to extend the Chi-Square test to groups of categorical variables by considering a new feature, whose classes are the cartesian product of the group variables’ classes, and frequency given by the contingency table. The test is carried out on the new feature. Notice that the more variables in the group, the more likely the test assumptions are not met.

DAISYnt considers the power set of the categorical variables. Starting from the 2-variables groups, it checks whether the test assumptions are fulfilled. If so, the Chi-Square test is carried out and the acceptance/rejection is recorded, otherwise the group is discarded. Bigger groups are tested only if all the sub-groups met the assumptions (otherwise the super-group would fail them either). The fraction of accepted tests is the DAISYnt similarity metric for multivariate categorical distributions.

C. Discriminator Model

Eventually, we propose to aggregate $D_{test}$ and $D_{synth}$ together, label each tuple based on the dataset it belongs to, and train a discriminator model to predict the label. Even if this test is not explicitly concerned with distributions, discriminators obtain good performance by implicitly learning distributional differences between the two datasets.

DAISYnt exploits a Gradient Boosting (xgboost) and a shallow Neural Network (nn) as discriminators (more on model specifics in the Data Utility section), and evaluate their performance using Gini Index. Since we strive for synthetic data to be not distinguishable from the real ones, the discriminator test metric is rescaled as follows:

$$1 - \frac{Gini(d_{xgb}) + Gini(d_{nn})}{2}$$

V. DATA UTILITY TESTS

Synthetic data have many different applications, not least to replace real data in prediction tasks. Conceptually, two datasets stemming from the same multivariate distribution shall yield the same insights, but this is not always the case when advanced models are involved.

To ensure synthetic data maintain data utility, DAISYnt trains prediction models respectively on $D_{train}$ and $D_{synth}$ and devises three tests with an increasing level of detail.

The models employed are Gradient Boosting (Xgboost [24] implementation) and Neural Network. Xgboost is trained with
Cosine Similarity is computed on centered prediction vectors (vector difference (corresponding to the ranking concept for predictions). The metric is rescaled as follows:

$$\frac{1 - A(Y_{GB}^T) - A(Y_{GB}^S) + A(Y_{NN}^T) - A(Y_{NN}^S)}{2}$$

where $A$ is the AUC value computed between observed $Y$ values on $D_{test}$ and model predictions. An example is given in Fig. 6.

**B. Single Prediction Comparison**

Woefully we acknowledge that AUC, due to its aggregate nature, do not guarantee the similarity of single predictions. Conversely, obtaining the same prediction probability rankings (on the $D_{test}$ individuals) is important to produce the same insights.

DAISYnt therefore compares prediction vectors using the Cosine Similarity, since it takes into account solely vector direction (corresponding to the ranking concept for predictions). The metric is computed on centered prediction vectors (vector mean subtracted element-wise), in order to achieve values all over the interval $[0, 1]$. In fact, consider two randomly generated prediction vectors, i.e. two models which learnt completely different patterns, basic Cosine Similarity achieves average values of 0.75, while the centered version values are close to 0.

Cosine similarity is evaluated on the centered prediction vectors, for Gradient Boosting and Neural Network separately. Results are then aggregated into a final test metric:

$$\frac{CS_{cent}(\hat{Y}_{GB}^S, \hat{Y}_{GB}^T) + CS_{cent}(\hat{Y}_{NN}^S, \hat{Y}_{NN}^T)}{2}$$

**C. Compare model internals**

The most refined and challenging comparison step is to check whether the model internals are the same. In a Neural Network, data flows through the network and gets transformed according to the neurons weights. We define layer activations as the values obtained by passing a data matrix through the network up to the chosen layer. The objective is to compare the activations of the $NN^T$ and $NN^S$ hidden layers, when passing $D_{test}$ as the data matrix. However, Neural Networks employ a huge number of parameters that frequently cause an over-representation, i.e. there are possibly infinite ways of achieving the same prediction, via different intermediate layer values. We cannot compare activations directly: they would always be different. Rather, we consider two activations to be equal when they are isotropic scaling or orthogonally invariant.

The HSIC quantity [26] captures orthogonal transformations but cannot handle the Isotropic Scaling invariance, which can be achieved by rescaling it into the Centered Kernel Alignment (CKA) [25]. Both CKA and HSIC employ kernels to achieve the required transformations. Popular options are the linear and gaussian kernels, where the difference consists in a trade-off between flexibility and computation time.

DAISYnt default consists in linear CKA between the $NN^T$ and $NN^S$ activations matrices, since the linear kernel usually achieves very similar results to the gaussian one [25].

**VI. Privacy Tests**

Since there exist no privacy regulation yet on synthetic data, we consider the taxonomy of privacy risks related to personal data, issued in the Article 29 Working Party Opinion 05/2014. Only the Linkability risk definition is slightly modified, to emphasize that building a generative model implies privacy obligations towards the $D_{train}$ individuals. The privacy dangers are grouped as:
• **Singling out risk**: The attacker single out an individual, using \( D_{\text{synth}} \).

• **Linkability risk**: The attacker learns whether a given \( D_{\text{train}} \) record has been used to train the generative model.

• **Inference risk**: The attacker infer the value of a sensitive attribute for new records, using the relationships contained in the \( D_{\text{synth}} \).

Hereafter, the focus is on black-box privacy attacks (the attacker has access to synthetic data only) and how to defend against them, rather than on formal privacy frameworks such as k-anonymity, l-diversity or differential privacy [27], which are difficult to evaluate post-hoc. Membership Inference attacks usually exploit an overfitted generative model which creates \( D_{\text{synth}} \) records too close too similar to the \( D_{\text{train}} \) ones. Closeness is used in [28], while the similarity in [29], to retrieve which records belong to \( D_{\text{train}} \). A different attack can be devised using the same procedure as [30], i.e. training a model to predict a \( D_{\text{synth}} \) sensitive variable and exploit it to infer confidential information on new records.

In the following we will present tests useful either to ensure that privacy is not at risk or to spot potential vulnerabilities.

### A. Singling Out Tests

1) **Cloned rows test**: The first test checks the presence of equal records in \( D_{\text{train}} \), \( D_{\text{synth}} \) and measures their percentage. Cloned rows can be removed from the synthetic data, at the cost of a having less data and modifying the statistical properties of the dataset.

2) **Close rows test**: The second test looks for very similar rows. DAISYnt converts continuous variables into categorical through binning, on both \( D_{\text{train}} \) and \( D_{\text{synth}} \), and use the new categorical versions to calculate the Hamming distance for each original-synthetic pair of records. Pairs with Hamming distance \( < 2 \) are considered close. The test metric is the fraction of \( D_{\text{synth}} \) records which are close to no \( D_{\text{train}} \) records.

An example of cloned and close rows between \( D_{\text{train}} \) and \( D_{\text{synth}} \) on the Adult dataset is given in Fig. 8.

### B. Linkability Tests

An overfitted \( G \) model may potentially leak information about the dataset used for training. In practice, an attacker may re-identify the \( D_{\text{train}} \) records starting from \( D_{\text{synth}} \), if she has access to a database containing some of the \( D_{\text{train}} \) individuals.

1) **Linkability distance test**: The assumption here is that an overfitted \( G \) model shall produce \( D_{\text{synth}} \) units much closer to the \( D_{\text{train}} \) ones than to \( D_{\text{test}} \) [28], i.e. choosing a fixed tiny size \( \epsilon \), the \( \epsilon \)-neighbourhood of a generic \( D_{\text{train}} \) individual \( U_{\epsilon}(x \in D_{\text{train}}) \) will presumably contain more \( x' \in D_{\text{synth}} \) than \( U_{\epsilon}(x \in D_{\text{test}}) \). Since distance notions suffer the curse of dimensionality, we perform Factor Analysis of Mixed Data (FAMD) [31] to retrieve a restricted set of orthogonal variables. On the new coordinates system, we compute the euclidean distance matrix between \( D_{\text{train}} \cup D_{\text{test}} \) and \( D_{\text{synth}} \).

The “median heuristic” [28] supports the choice of \( \epsilon \). Per each \( x \in D_{\text{train}} \cup D_{\text{test}} \) we count the fraction of \( D_{\text{synth}} \) units belonging to its neighbourhood, and use it as a ranking measure.

\[
\text{rank}(x) = \frac{1}{n_{\text{synth}}} \sum_{i=1}^{n_{\text{synth}}} 1 \{ x' \in U_{\epsilon}(x) \} \quad \text{where } x' \in D_{\text{synth}}
\]

where \( 1 \) is the indicator function.

Under the initially stated assumption, \( D_{\text{train}} \) data should have higher \( \text{rank} \) values, i.e. the \( \text{rank} \) variable should discriminate well between \( D_{\text{train}} \) and \( D_{\text{test}} \) data. Its discrimination power is measured through AUC, obtaining the test metric:

\[
1 - \text{AUC}_{\text{rank}}
\]

From a different standpoint, we create a classification model exploiting the closeness of synthetic data to classify the units as \( D_{\text{train}} \) or \( D_{\text{test}} \).

2) **Linkability ML test**: Information contained in \( D_{\text{synth}} \) should be more valuable for re-estimating the \( D_{\text{train}} \) data rather than \( D_{\text{test}} \), when \( G \) is an overfitted model [29]. To test such assumption, we build a Neural Network on the \( D_{\text{synth}} \) target variable and employ the model to obtain target prediction \( \hat{y}|x \) on the \( x \) units of \( D_{\text{train}} \) and \( D_{\text{test}} \). Considering a classification task, the cross-entropy error \( \text{Err}(\hat{y}|x) \) per each single prediction is computed (yet the generality of the framework allows for any other error metric to be used).

As before, we test the \( \text{Err} \) variable discrimination power, i.e. how well it separates \( D_{\text{train}} \) from \( D_{\text{test}} \) units. Considering the worst-case scenario, the threshold \( \tau \) is chosen to maximize the proportion differences of \( D_{\text{train}} \), \( D_{\text{test}} \) over and below \( \tau \) (an attacker usually does not have enough information to choose the best \( \tau \)). The test metric is computed as follows:

\[
1 - \left( \frac{\sum_{x \in D_{\text{train}}} 1\{ \text{Err}(\hat{y}|x) < \tau \}}{n_{\text{train}}} - \frac{\sum_{x \in D_{\text{test}}} 1\{ \text{Err}(\hat{y}|x) < \tau \}}{n_{\text{test}}} \right)
\]
C. Inference Risk test

We assume that an attacker has access to a set of public variables \( X_{pub} \), but only an aggregate knowledge (average) of a sensitive variable \( \bar{y}_{sens} \), i.e. \( \bar{y}_{sens} \). We measure the \( \bar{y}_{sens} \) increased disclosure the attacker gains, by having access to \( D_{synth} \). The attacker may build a prediction model \( f : X_{pub} \rightarrow \bar{y}_{sens} \) to predict the sensitive attribute on any individual, rather than predicting \( y_{sens} \).

Recall that \( R^2 \) metric calculates the prediction improvement using the \( f \) model, against the basic average \( \bar{y}_{sens} \) prediction. We employ a slightly modified \( R^2 \) version to measure the inference risk on the \( D_{train} \) units:

\[
\frac{\sum_{i=1}^{n_{train}} (y_i - f(X_{pub},i))^2}{\sum_{i=1}^{n_{train}} (y_i - \bar{y}_{sens})^2}
\]

Such test is valuable for the data owner to understand the dangers of sharing \( D_{synth} \) and to decide how to protect against them.

VII. APPLICATIONS

Various financial applications resort to prediction models to gain important insights. A prominent example is Credit Scoring, which consists in estimating the probability that a debtor will not repay the due amount [32]. Several demographic, economic and financial variables concur to classify each individual as good or bad payer. The current application employs anonymized real-world Credit Bureau data, composed of 16 variables and 1.49% of “bad payers”, split into \( D_{train} \) and \( D_{test} \) of size 138,273 and 59,261 respectively. The scope of the application is to create a reliable synthetic replica to be used for data sharing with model development purposes. To this end, four generative models have been trained to produce \( D_{synth} \) datasets of the same \( D_{train} \) size, and DAISYnt is used to determine the best replica.

Different open-source [15] model implementations have been employed: i) Gaussian Copula (GC), ii) Conditional Tabular GAN (CTGAN), iii) CopulaGAN (CopGAN), iv) Tabular Variational Auto-Encoder (TVAE). Out-of-the-box SDV models were used with no fine-tuning of the hyper-parameters. Concerning the test suite, DAISYnt package\(^1\) requires only few essential parameters, namely the \( D_{train}, D_{test}, D_{synth} \) datasets, knowledge of the categorical features and the target variable. These information are used for a shallow preprocessing step, in which DAISYnt takes care of missing and special values (if specified). On categorical variables they are handled using ad-hoc classes, while on continuous variables they are treated through mean imputation and a flag indicating which records held the missing/special fields. The target variable is instead required for data utility. Additionally, we may specify the list of tests to perform, by default DAISYnt runs the entire set.

DAISYnt results are summarised in Table I. All the tests described in the methodological sections have been performed, apart from the Inference Risk test. The latter is especially valuable to understand privacy implications of sharing synthetic data, although it requires to know which features are available to the adversary to carry out an Inference attack.

General purpose tests guarantee same correlation and predictive power patterns, proving that the datasets can be used for descriptive analysis. Concerning the distribution related tests, we notice satisfactory results for the univariate distributions (with binned variables) and multivariate categorical distributions. However, more in-depth tests, such as the MMD univariate, continuous multivariate distributions and the discriminator test, show quite poor values. Since we tested just the vanilla implementations, distribution discrepancies were expected. However, these results suggest the replicas cannot be employed for advanced statistical analyses, such as rebalancing good-bad payers [33]. About data utility, aggregate predictions testify similar AUC values, but this is not enough to enable model development on the synthetic data. In fact, the more in-depth single prediction test shows that models on \( D_{train} \) and \( D_{synth} \) achieve substantially different insights. Eventually, high privacy metrics guarantee that privacy is preserved and the models do not overfit.

In conclusion, GAN models perform visibly better with respect to Gaussian Copula and TVAE, especially in terms of data utility. The generated datasets may be used for data sharing.

---

Table I: DAISYnt results on the \( D_{synth} \) datasets, obtained through generative models

| Test                        | Group     | Detail | GC | CTGAN | CopGAN | TVAE |
|-----------------------------|-----------|--------|----|-------|--------|------|
| Correlations                | General   | basic  | 0.93| 0.96  | 0.97   | 0.94 |
| Predictive Power           | General   | basic  | 0   | 0.97  | 0.92   | 0.38 |
| Univariate Distributions (bins) | Distrib     | basic  | 0.81| 0.94  | 0.88   | 0.94 |
| Univariate Distributions (MMD) | Distrib     | in-depth | 0.13| 0.25  | 0.13   | 0.25 |
| Multivariate Categorical Distributions | Distrib | basic  | 0.99| 1     | 0.99   | 1    |
| Multivariate Continuous Distributions | Distrib | in-depth | 0   | 0     | 0     | 0    |
| Discriminator              | Distrib   | in-depth | 0.01| 0.07  | 0.08   | 0.05 |
| Aggregated Predictions     | Utility   | in-depth | 0.68| 0.95  | 0.92   | 0.77 |
| Single Predictions         | Utility   | in-depth | 0.92| 0.46  | 0.32   | 0.12 |
| Model Internals            | Utility   | in-depth | 0.70| 0.79  | 0.66   | 0.61 |
| Closed Rows                | Privacy   | basic  | 1   | 0.99  | 0.99   | 0.99 |
| Close Rows                 | Privacy   | basic  | 1   | 0.99  | 0.99   | 0.99 |
| Linkability Distance       | Privacy   | basic  | 0.95| 1     | 0.98   | 1    |
| Linkability ML             | Privacy   | basic  | 1   | 0.99  | 0.99   | 0.99 |

\(^1\)https://pypi.org/project/daisyn/
purposes, thanks to good privacy results. For model development purposes, fine-tuning of the hyper-parameters or more powerful generative algorithms are viable solutions to achieve satisfying results.

VIII. CONCLUSIONS

Recent advances in the generative models field seem to solve the emerging need of high quality and private data. But the adoption of this new technology require consistent and detailed metrics to evaluate salient properties of the generated datasets. The present paper fills the gap, providing a taxonomy of the important aspects and powerful tests to assess them. The DAISYnt test suite is an easy-to-use python package, characterized by modularity and flexibility. The tests are grouped by different properties and level of detail. Based on the data generation purpose, we may choose the most appropriate tests to run. In fact, different use-cases require different quality levels of each concept. As an example, data sharing use-cases usually require high privacy levels, model development needs good data utility, while data augmentation compels strong distribution similarity. In this regard, DAISYnt is particularly versatile and business oriented.

In general, the tests are suitable also for big data sizes. Although few of them take long to compute and may require considerable storage space, to keep track of distance matrices. This is the case for the “Univariate Distributions Test” using MMD, the “Model Internals Test”, the “Close rows test” and the “Linkability distance test”. To tackle this issue, we provided an alternative implementation of the “Univariate Distributions Test” based on binning which drastically reduce the computation time. In addition, DAISYnt presents a class parameter to perform data subsampling for these particular tests: reducing the dataset size to match computational and hardware requirements of the users. Although, using the sub-sampling parameter lends itself to two shortcomings: i) it is not suggested for the Privacy tests, since they require each unit to be tested to properly assess privacy risks and, in general, ii) the results are coarser since we consider only a small part of the data.

Future directions therefore entail tests refinement, as well as exploring new ways for obtaining highly accurate results with a reduced computation time (for the above-mentioned tests). Another interesting research direction consists in extending DAISYnt to regression and multi-class classification prediction tasks. Nonetheless, we consider the current ease of use and the applied use-cases focus as essential aspects of DAISYnt and we will strive to maintain them under future improvements.

REFERENCES

[1] C. J. Hoofnagle, B. van der Soot, and F. Z. Borgesius, “The European Union general data protection regulation: What it is and what it means,” Information & Communications Technology Law, vol. 28, no. 1, pp. 65–98, 2019.
[2] T. Gilbert, “Family Educational Rights and Privacy Act (FERPA),” 2007.
[3] I. G. Cohen and M. M. Mello, “HIPAA and protecting health information in the 21st century,” Jama, vol. 320, no. 3, pp. 231–232, 2018.
[4] Y. Wang, “Privacy-enhancing technologies,” in Handbook of Research on Social Liabilities in Information Security. IGI Global, 2009, pp. 205–227.
[5] S. Assefa, “Generating synthetic data in finance: Opportunities, challenges and pitfalls,” Challenges and Pitfalls (June 23, 2020), 2020.
[6] Z. Wang, P. Myles, and A. Tucker, “Generating and Evaluating Synthetic UK Primary Care Data: Preserving Data Utility & Patient Privacy,” in 32nd International Symposium on Computer-Based Medical Systems (CBMS). IEEE, 2019, pp. 126–131.
[7] D. Dua and C. Graff, “UCI machine learning repository.” 2017. [Online]. Available: http://archive.ics.uci.edu/ml
[8] M. Hittmeir, A. Ekelhart, and R. Mayer, “Utility and Privacy Assessments of Synthetic Data for Regression Tasks,” in 2019 IEEE International Conference on Big Data (Big Data). IEEE, 2019, pp. 5763–5772.
[9] A. Galloni, I. Lendáč, and T. Horváth, “A Novel Evaluation Metric for Synthetic Data Generation,” in International Conference on Intelligent Data Engineering and Automated Learning. Springer, 2020, pp. 25–34.
[10] M. Hittmeir, A. Ekelhart, and R. Mayer, “On the utility of synthetic data: An empirical evaluation on machine learning tasks,” in Proceedings of the 14th International Conference on Availability, Reliability and Security, 2019, pp. 1–6.
[11] A. M. Alaa, B. van Breugel, E. Savelev, and M. van der Schaar, “How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models,” arXiv preprint arXiv:2102.08921, 2021.
[12] M. Platzer and T. Reutterer, “Holdout-based empirical assessment of mixed-type synthetic data,” Frontiers in Big Data, p. 43. 2021.
[13] C. Esteban, S. L. Hyland, and G. Ratsch, “Real-valued (medical) time series generation with recurrent conditional gans,” arXiv preprint arXiv:1706.02633, 2017.
[14] J. Jordon, J. Yoon, and M. van der Schaar, “Measuring the quality of synthetic data for use in competitions,” arXiv preprint arXiv:1806.11345, 2018.
[15] N. Patki, R. Wedge, and K. Veeramachaneni, “The synthetic data vault,” in 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA). IEEE, 2016, pp. 399–410.
[16] A. V. Dorogush, V. Ershov, and A. Gulin, “CatBoost: Gradient boosting with categorical features support,” arXiv preprint arXiv:1810.11363, 2018.
[17] M. Herdin, N. Czink, H. Ozcetik, and E. Bonek, “Correlation matrix distance, a meaningful measure for evaluation of non-stationary MIMO channels,” in 2005 IEEE 61st Vehicular Technology Conference, vol. 1. IEEE, 2005, pp. 136–140.
[18] E. Zdravevski, P. Lameski, and A. Kulakov, “Weight of evidence as a tool for attribute transformation in the preprocessing stage of supervised learning algorithms,” in International Joint Conference on Neural Networks. IEEE, 2011, pp. 181–188.
[19] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, “A kernel method for the two-sample-problem,” Advances in neural information processing systems, vol. 19, pp. 513–520, 2006.
[20] D. J. Sutherland, H.-Y. Tung, H. Strathmann, S. De, A. Ramdas, A. Smola, and A. Gretton, “Generative models and model criticism via optimized maximum mean discrepancy,” arXiv preprint arXiv:1611.04498, 2016.
[21] M. J. Fisher, A. P. Marshall, and M. Mitchell, “Testing differences in proportions,” Australian Critical Care, vol. 24, no. 2, pp. 133–138, 2011.
[22] M. B. Kursa, A. Jankowski, and W. R. Rudnicki, “Boruta—a system for feature selection,” Fundamenta Informaticae, vol. 101, no. 4, pp. 271–285, 2010.
[23] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, “Conditional variable importance for random forests,” BMC bioinformatics, vol. 9, no. 1, pp. 1–11, 2008.
[24] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[25] S. Korblith, M. Norouzi, H. Lee, and G. Hinton, “Similarity of neural network representations revisited,” in International Conference on Machine Learning. PMLR, 2019, pp. 3519–3529.
[26] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, “Measuring statistical dependence with Hilbert-Schmidt norms,” in International Conference on Algorithmic Learning Theory. Springer, 2005, pp. 65–77.
[27] S. M. Bellovino, P. K. Dutta, and N. Reitinger, “Privacy and synthetic datasets,” Stan. Tech. L. Rev., vol. 22, p. 1, 2019.
[28] B. Hilprecht, M. Härterich, and D. Bernau, “Monte carlo and reconstruction membership inference attacks against generative models,” *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 4, pp. 232–249, 2019.

[29] L. Song, R. Shokri, and P. Mittal, “Privacy risks of securing machine learning models against adversarial examples,” in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 241–257.

[30] A. Kassem, G. Acs, C. Castelluccia, and C. Palamidessi, “Differential inference testing: A practical approach to evaluate sanitizations of datasets,” in *2019 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2019, pp. 72–79.

[31] G. Saporta, “Simultaneous analysis of qualitative and quantitative data,” in *Società Italiana Di Statistica*, vol. 1. CEDAM, 1990, pp. 62–72.

[32] G. Visani, E. Bagli, F. Chesani, A. Poluzzi, and D. Capuzzo, “Statistical stability indices for LIME: Obtaining reliable explanations for machine learning models,” *Journal of the Operational Research Society*, vol. 73, no. 1, pp. 91–101, 2022.

[33] G. Cascarino, M. Moscatelli, and F. Parlapiano, “ Explainable Artificial Intelligence: Interpreting default forecasting models based on Machine Learning,” Bank of Italy, Economic Research and International Relations Area, Tech. Rep., 2022.