Utility and Privacy Assessments of Synthetic Data for Regression Tasks

Markus Hittmeir
SBA Research
Vienna, Austria
mhittmeir@sba-research.org

Andreas Ekelhart
SBA Research
Vienna, Austria
aekelhart@sba-research.org

Rudolf Mayer
SBA Research
Vienna, Austria
rmayer@sba-research.org

***Abstract*—With ever increasing capacity for collecting, storing, and processing of data, there is also a high demand for intelligent data analysis methods. While there have been impressive advances in machine learning and similar domains in recent years, this also gives rise to concerns regarding the protection of personal and otherwise sensitive data, especially if it is to be analysed by third parties. Besides anonymisation, which becomes challenging with high dimensional data, one approach for privacy-preserving data mining lies in the usage of synthetic data, which comes with the promise of protecting the users' data and producing analysis results close to those achieved by using real data. In this paper, we analyse a number of different approaches for creating synthetic data, and study the utility of the created datasets for regression tasks, i.e. the prediction of a numeric value. We further investigate the similarity of real and synthetic data samples. Finally, we contribute to privacy assessments and measurements of the risk of attribute disclosure on synthetic data by extending an approach developed for categorical data.**

***Index Terms*—Synthetic Data, Machine Learning, Privacy, Regression, Attribute Disclosure Assessment**

I. Introduction

Driven by the recent success of machine learning on challenging problems, organisations increasingly want to explore its potential and implement automated services. For this purpose, large amounts of data are collected, stored and processed by both public and private organisations, covering domains such as health care, employment, finance, or social media. While ever more approaches and tools are released to analyse data, the privacy of individuals has to be protected. Ethical and regulatory standards, such as the EU's General Directive on Data Protection (GDPR), oblige data holders and providers to implement technical and organisational measures to keep the personal data secure and to ensure its lawful processing.

Statistical disclosure control (SDC) refers to techniques to ensure that no person is identifiable from published data. In the case of micro-data, two possibilities of disclosure of sensitive information are considered: Identification disclosure happens when an attacker is able to conclude that a certain record in the dataset belongs to a certain individual. Attribute disclosure happens whenever the dataset allows the attacker to learn new information about the individual in question, e.g. the value of a certain attribute. In most cases, it does not suffice to remove primary identifiers like names or social security numbers from the data, due to potential re-identification via linkage attacks. To minimise disclosure risks, approaches like Differential Privacy [2] and k-Anonymity [16] have been developed. The reader may consult [5] for a general overview on privacy-preserving data publishing methods. However, these approaches still have shortcomings. For example, k-anonymity is still subject to re-identification attacks whenever sufficient, auxiliary background knowledge is available.

In this paper, we will thus consider the generation of *synthetic data* as the main disclosure control measure. Synthetic data refers to data samples generated from a model that is, in turn, obtained from a real dataset. By the synthesis process, global properties in the dataset are retained, while details on specific individuals are suppressed. However, organisations willing to apply this method lack information on (i) how to generate synthetic data, (ii) the utility of synthetic data for machine learning tasks, and (iii) information on how well the privacy of individuals is really protected, e.g. in terms of attribute disclosure risk. Since fully synthetic records do not relate to original records in terms of a 1-to-1 correspondence, the notion of identification disclosure is not in our focus. In our privacy assessment, we therefore consider attribute disclosure risks and assume that the attacker knows the values of certain attributes of their victim (called the *key* variables) and wants to learn the value of some sensitive attribute (called the *target* variable). Approaches for measuring the related risk have been proposed by Reiter et al. [13] and by Taub et al. [17]. The methods differ by the amount of the assumed background knowledge $\mathcal{B} = \{A, S\}$ of the attacker. A denotes the attacker's knowledge about records in the original dataset, and S comprises available information about the process of generating the synthetic data, like code for the synthesizer or a description of the used tools. Reiter et al.'s approach assumes a worst case attacker scenario, in which the adversary knows all entries in the original dataset except the target attribute value they want to learn. While the authors admit that this assumption may be viewed as overly conservative and unrealistic, they suggested that their measures offer a type of

This work was partially funded by the KIRAS program (No 860663) of the Austrian Research Promotion Agency (FFG), and the EU Horizon 2020 research and innovation programme under grant agreement No 732907 (Project "MyHealthMyData"). The competence center SBA Research (SBA-K1) is funded within the framework of COMET — Competence Centers for Excellent Technologies by BMVIT, BMDW, and the federal state of Vienna, managed by the FFG.

upper bound on the disclosure risks. Taub et al.'s approach, on the other hand, assume an attacker's behaviour that does not rely on $\mathcal{B}$ at all, and is feasible for $A = S = \emptyset$. Consequently, the risk estimates offered by their measure may be considered as a type of lower bound.

In [7], two synthetic data generation tools have been evaluated for their utility for the supervised machine learning task of *classification*, i.e. the automatic categorisation of unknown data samples into one of a predefined set of categories. In the present paper, we extend this evaluation to *regression* tasks, i.e. the supervised learning of a model for predicting a continuous output variable. We complement the choice of synthetic data generation tools (see Section II) and compare the performance of the evaluated approaches. In addition to the utility evaluation, we conduct an empirical privacy assessment of the generated synthetic datasets. For this purpose, we develop a procedure to measure the distance between original and synthetic data samples. Another contribution of this paper is the extension of the attribute disclosure method by Taub et al. This technique has only been formulated for categorical key attributes. We develop a novel method for continuous variables, and measure disclosure risks on the same datasets we used for the utility assessment. As a result, this allows us to compare both aspects for the investigated synthetic data generation tools.

The remainder of this paper is organised as follows. Section II gives on overview on related work about synthetic data generation. In Section III, the approach of Correct Attribution Probability is discussed. Furthermore, we present our novel extension to continuous variables. Section IV describes the datasets we used and the experimental setup of our empirical utility evaluation, the results of which are going to be discussed in Section V. Section VI contains our privacy assessment. Finally, we provide conclusions and an outlook on future work in Section VII.

II. Related Work

One of the earliest applications of synthetic data is described by Rubin in [14], where multiple imputation is used to synthetically generate certain columns of datasets. A comparative study of existing methods for synthetic data generation can be found in [15]. For our evaluation in this paper, we selected the following three data synthesizers: The *Synthetic Data Vault* (DV)[11] has been developed in 2016, and is provided as implementation for the Python programming language. It builds a model based on estimates for the distributions of each column. In order to preserve the correlation between attributes, the synthesizer applies a multivariate version of the Gaussian copula and, subsequently, computes the covariance matrix. For more details and a utility evaluation conducted by the developers, please refer to [11]. We further utilise the *DataSynthesizer* (DS) [12], which has also been developed in Python in 2017. The DataSynthesizer provides three approaches for learning a representation of the original dataset: a 'random mode', the 'independent attribute mode', and the 'correlated attribute mode'. In 'correlated attribute mode', dependencies between attributes are preserved in the model. The DataSynthesizer generates synthetic data based on a Bayesian network model learned from the original data. For SDC, the DataSynthesizer uses the framework of Differential Privacy, and offers the possibility to inject noise in the model and thus subsequently into the generated data, by a user-controlled parameter specifying the magnitude. More information can be found in [12]. Finally, we use the *synthpop* (SP) [10] package for the statistical analysis language *R*. In this case, the default synthesis method is the CART (Classification and Regression Trees) algorithm. However, the user is able to specify a large number of parameters and may apply a built-in function for disclosure control to the resulting synthetic dataset.

When sanitising a dataset via anonymisation, synthetisation or related approaches, some sensitive information at the level of individual records is invariably removed [1]. Utility evaluation of privacy-preserving methods, by means of a *utility metric*, can generally be done in two directions. The first approach is to measure certain properties on the modified (or created) dataset, as opposed of the original data set. Metrics could include various statistical moments such as mean or standard deviation, or more generally a comparison of two distributions. This utility evaluation has the advantage of being independent of the final task being carried out on the dataset, but is also generally less precise and more difficult to quantify. Another approach is to measure the utility on a task that the dataset is intended for, e.g. a supervised learning task in the form of regression analysis. In this approach, the metric measures the differences in regression effectiveness of the models on the original vs. the synthetic dataset, as e.g. for the effect of anonymisation via k-anonymity in [9]. For regression effectiveness, commonly used measures such as Mean Absolute Error (MAE) could be employed and then compared for differences on the two flavours of the data.

For our evaluation, we utilise both approaches, i.e. utility measured directly on the synthetic dataset by an analysis of attribute correlation, as well as on a specific task, by means of a regression analysis.

III. Disclosure Risk of Continuous Attributes

The concept of Correct Attribution Probability (CAP) has been introduced in [3] and elaborated on in [17] by J. Taub et al. For assessing attribute disclosure risk, CAP assumes that the attacker knows the values of a set of key attributes for an individual in the original dataset, and wants to learn the respective value of some target attribute. For now, we suppose that these attributes are categorical. Consider a dataset $\mathcal{O}$ consisting of micro-data with n records representing individuals and an unspecified number of attributes in the columns. For $j \in \{1, \ldots, n\}$, let $K_{o,j}$ be the vector representing the values of the key attributes of the j-th record in the original dataset, and let $T_{o,j}$ be the corresponding value of the target attribute. Let $\mathcal{S}$ be a fully synthetic version of $\mathcal{O}$. We define $K_{s,j}$ and $T_{s,j}$ for the synthetic dataset.

Suppose that an attacker knows $K_{o,j}$ of the j-th record in the original dataset, and has access to $\mathcal{S}$, the synthesised version of $\mathcal{O}$. The basic assumption is that they would search for all records in $\mathcal{S}$ with $K_{s,i} = K_{o,j}$. The resulting group of matching synthesised records is often referred to as *equivalence class* of $K_{o,j}$ in $\mathcal{S}$. As their prediction for the target $T_{o,j}$, the attacker now picks the value resulting from a majority vote among the $T_{s,i}$ for all records in the equivalence class. Correspondingly, the CAP score for record j in the original dataset is the empirical probability of its target value given its key attribute values, that is

$$\mathrm{CAP}_{s,j} := \frac{\sum_{i=1}^{n}[T_{s,i} = T_{o,j} \wedge K_{s,i} = K_{o,j}]}{\sum_{i=1}^{n}[K_{s,i} = K_{o,j}]}.$$

Note that the denominator is 0 if the combination of attribute values in $K_{o,j}$ does not occur in the synthetic dataset. In the original publications, it is suggested to either define the corresponding CAP scores as 0 or to treat them as undefined. Both approaches are based on the assumption that an attacker would not be able to form an equivalence class and, hence, a prediction for the target variable. However, we now show that this assumption is challenged by the attacker's possibility to use tools from machine learning and improve the proposed method to obtain a prediction.

If $K_{o,j}$ consists not only of categorical, but also of continuous variables, there will most likely be no *exact* match in the synthesised records. However, also near matches of continuous variables might be considered relevant, if the difference to a known value is either really small, or the exact continuous value is not known. Assume e.g. that we know from a specific person that she earns approximately 4,000$, then values very close to that should also be considered matches. We therefore need another approach for constructing the corresponding equivalence classes. In this context, it is natural to consider those records i in $\mathcal{S}$ for which $K_{s,i}$ is comparably close to $K_{o,j}$ relative to a certain metric Δ. Let $\mathcal{S}\big|_{K,T}$ denote the dataset that results from omitting all attributes but the target T and those in the key K. The attacker may form an equivalence class and a prediction by following the subsequent procedure.

Algorithm 3.1: Input: A synthetic data set $\mathcal{S}$, a target attribute T in $\mathcal{S}$ and an attribute key K together with a value vector $K_{o,j}$ of an original data's record. Furthermore, a metric Δ and a radius ρ.

Output: A prediction T^* for $T_{o,j}$

1: Set $N = \emptyset$.
2: $N \leftarrow \left\{a \in \mathcal{S}\big|_{K,T} : \Delta\left(K_{o,j}, a\big|_K\right) \leq \rho\right\}$, where $a\big|_K$ omits the value of T.
3: Choose T^* as the arithmetic mean of the values of T for the elements in N.

First, we have to choose a metric Δ and a suitable radius ρ for the neighbourhood defining the equivalence class. Depending on Δ, it may be necessary to prepare the data for these computations by applying label encoding and/or feature scaling. Instead of an arithmetic mean, we may also want to consider a weighted mean with regards to the distance $\Delta\left(K_{o,j}, a\big|_K\right)$. Such approach would render those elements a as highly relevant which have a combination of attribute values that is most similar to $K_{o,j}$. Moreover, it puts less emphasis on the difficult problem of finding a proper choice for ρ.

It is easy to see that the procedure discussed above is just a variation of a Radius-Nearest-Neighbour algorithm for regression. Likewise, the original CAP approach is equivalent to setting $\rho = 0$. The validity of the predictions produced by this procedure may be evaluated by the same methods used for the evaluation of other regression algorithms. These measures will be described in the subsequent section. The concrete choice of Δ and ρ for our privacy experiments will be discussed in Section VI. We now describe our setup for the utility evaluation.

IV. Experimental Setup

We evaluate both the utility and the privacy aspects of synthetic data. We therefore aim for utilising datasets that have attributes with personal, potentially sensitive data. To be able to discuss the results in detail, we utilise datasets that are publicly available and not restricted. Further, this enables the experiments to be repeatable. We thus base our experiments on datasets that are frequently used for benchmark evaluation, even if they are limited in size. The most important characteristics of these datasets are listed in Table I.

TABLE I: Dataset Characteristics

Dataset	# Features	# Instances	Target Variable
Boston Housing[1]	14	506	'MedianValue'
Bike Sharing[2]	16	731	'Count'
Social Network Ads[3]	5	400	'EstSalary'
Insurance[4]	7	1,338	'Charges'

The Social Network and the Insurance datasets contain personal identifying information and sensitive data (e.g., demographic, health and financial information), which are typical cases for the application of privacy-preserving techniques. However, we also included two additional benchmark datasets for further analysis and comparison. The intention was to cover a range of different domains. Let us briefly discuss the associated tasks.

The first dataset concerns housing values in suburbs of Boston. The dataset has been introduced in [6]. The goal is to predict the median value of owner-occupied homes in $1000's based on information like the average number of rooms per dwelling, per-capita crime rate by town, the accessibility to radial highways or the pupil-teacher ratio.

The Bike Sharing dataset, first presented in [4], comprises two variants, aggregating bike sharing counts either on an hourly or daily basis. We worked with the version for the

[1]https://archive.ics.uci.edu/ml/machine-learning-databases/housing/
[2]https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset
[3]https://www.kaggle.com/rakeshrau/social-network-ads
[4]https://github.com/stedy/Machine-Learning-with-R-datasets/blob/master/insurance.csv

daily records, which consists of 731 entries. Here, the goal is to predict the count of total rental bikes based on input like the weather situation, temperature, humidity, season and weekday.

The Social Network Ads dataset has the initial task to predict if a user bought a certain product based on their gender, age and salary. In the experiment of the present paper, we use the dataset to define a regression task, i.e. we use gender, age and information about the purchase for the prediction of the salary of the individuals in the dataset.

Finally, the task on the insurance dataset, which has been published in [8], is to predict the amount of insurance charges of a customer, based on personal attributes like the gender, bmi, the number of children and whether the person smokes or not.

For the generation of synthetic data, we used the Synthetic Data Vault, the DataSynthesizer and the synthpop package, all of which have been discussed in Section II. As our primary goal is an unbiased evaluation and not an optimisation towards a specific synthesizer or target evaluation, we performed only a limited parameter search starting with the standard settings of each synthesizer, and implemented them as described in the respective documentation. For each of the datasets discussed in Section IV, we performed the following procedure in order to synthesise and prepare the data for the utility evaluation.

1) We deleted columns in the context of standard feature cleaning, e.g. purely identifying attributes like the primary key 'UserID' in the Social Network Ads dataset.
2) To ensure reliable and statistically sound results, we performed a repeated holdout method, i.e. we randomly generate ten different splits of the table into training and test data, such that the size of the latter is 20% of the original table. In the evaluation, we then report averages of this repeated application.
3) For each split, we applied the three data synthesis methods discussed above. The input to each of the implementations is the training dataset after it has been split, and as an output, we generate new, synthetic training data of equal length.

In order to investigate different configurations regarding its Differential Privacy settings, the DataSynthesizer is applied twice in Step 3. For each of the splits generated in Step 2, we therefore obtain six data files: (i) the original training data, (ii) the training data synthesised by the Synthetic Data Vault, (iii) the training data synthesised by synthpop, (iv) the training data synthesised by the DataSynthesizer without applying Differential Privacy, and (v) the training data synthesised by the DataSynthesizer when applying Differential Privacy with the parameter $\varepsilon = 0.1$, and finally (vi) the test dataset, which is used to estimate the generalisation error of the machine learning models on all the training sets.

We applied several popular regression models, namely Linear Regression, Support Vector Regression (SVR, based on the Support Vector Machine classification model), and Multilayer Perceptron (MLP) Regression, a neural-network based model. For all techniques, we utilised the implementation provided in the scikit-learn package available for the Python programming language[5]. For each dataset, we performed the following procedure:

1) We applied label encoding and sklearn's Standard Scaler for feature scaling on both the training and the test dataset. The Standard Scaler (also referred to as z-score normalisation) first subtracts the mean value from the population, and then scales vectors to unit variance.
2) We fitted the models to the training data and predicted results for the test data.
3) We repeated Step 1 and 2 for the synthesised training data from the DV, DS and SP.

For the MLP Regression, we used the "Limited-Memory BFGS" (lbfgs) solver, an alternative to stochastic gradient descent, as this option tends to converge faster and performs better on small training datasets compared to the standard 'adam' solver. For the Support Vector Regression, we chose an 'Radial Basis Function' (rbf) kernel, and set the error penalty parameter to $C = 100$. Besides that, the three regression models were used with default settings.

We present the results in aggregated tables in Section V. These tables consist of three sub-columns for each regression model, which contain the scores of the considered evaluation measures. In the following, we briefly describe the three selected, common measures: The Mean Absolute Error (MAE) is defined as

$$\left(\sum_{i=1}^{n} |y_i - x_i|\right) / n,$$

where n is the number of samples in the test dataset, y_i is the true and x_i is the predicted target value of the i-th record. Similarly, the Mean Absolute Percentage Error (MAPE) is defined as

$$\frac{100}{n} \sum_{i=1}^{n} \frac{|y_i - x_i|}{y_i}.$$

The idea is to scale MAE to a percentage error. This enables an easier comparison of values across different datasets, as other measures such as the MAE are generally influenced by the range of the predicted values, and a direct comparison of the values is thus not possible. Finally, we also use the R^2 score, which is defined by

$$1 - \frac{\sum_{i=1}^{n}(y_i - x_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2},$$

where $\bar{y}$ is the mean of the y_i. The R^2 score is generally in the value range of $-\infty$ to +1, with the latter being the optimal value. Our choice of evaluation metrics can be roughly summarised as follows. Both MAE and MAPE are easy to understand and appropriate for the comparison of model performance between the real data and synthetic data, as they provide a linear score. The MAPE, though it has some shortcomings, is chosen as it is based on percentages, and thus further enables comparability between different data

[5]https://scikit-learn.org/stable (we used package version 0.20.3)

sets. The R^2 score is another frequently used measure, and further contains information captured by the Mean Squared Error (MSE).

V. Utility Evaluation

In this section, we discuss the results of the three synthesizers and analyse the differences in their performances. We first describe the effects on the data itself, where we utilise the information on the preserved correlation between attributes in the dataset. We have summarised these relationships in a pairwise correlation plot, depicted in Figure 1 for the Bike Sharing dataset as representative example. Here, darker red colours indicated a stronger correlation, while darker blue colours indicate stronger indirect correlations; white indicates no correlation at all.

There are a couple of notable differences in the correlation preserving power of the synthesizers. On the one hand, the flavour of the DataSynthesizer without Differential Privacy (DS 0) seems to specifically preserve the correlation between the last two attributes, namely the number of registered users, and the actual count of rentals. The synthpop package seems to be generally similar, but rather weak in preserving many of the relations of the attribute "holiday". However, it can be observed that synthpop is better in preserving the "non-correlations" (i.e. the white cells in the correlation plot), while DS 0 seems to generate some additional correlations that did not manifest in the original dataset.

For the Synthetic Data Vault (DV), it can be seen that many more cells are white than for the original dataset and the two previously discussed synthesizers, meaning that DV specifically does not capture some of the correlations at all. Correlations are generally weaker and more random for the DataSynthesizer with Differential Privacy enabled (DS 0.1).

As second step of our evaluation, we will discuss the results of the regression task. In the following tables, each of the three cells shows the mean of the evaluation measures over the ten single splits, and also the standard deviation. In order to improve readability of the tables, we rounded all scores to integers, except for MAE in Boston Housing, due to its low value range and rather similar results. We start by considering Table II for Boston Housing. The DataSynthesizer without Differential Privacy (DS 0) and synthpop (SP) perform rather good for Linear Regression and SVR, but not for MLP Regression. The Synthetic Data Vault (DV) is best for Linear Regression, average for Support Vector Regression, and bad for MLP Regression. Compared to the scores achieved on the original data, the DataSynthesizer with enabled Differential Privacy (DS 0.1) appears to be rather useless for regression on the Boston Housing dataset, with the evaluation metric values being up to three times worse than on the original dataset.

On the Bike Sharing dataset, the results of which are reported in Table III, we observe that both synthpop and Synthetic Data Vault perform well for all three regression models. Especially noteworthy is that the Synthetic Data Vault achieves the lowest errors of all synthetisation approaches for the MLP Regression. Regarding the DataSynthesizer with Differential Privacy disabled (DS 0), it does fine for Linear Regression and Support Vector Regression, but shows a comparably large loss of performance for MLP Regression. Again, with Differential Privacy enabled (DS 0.1), this dataset leads to high MAPE percentages and a negative R^2 score for MLP.

We now consider Table IV for the Social Network dataset. Here we observed the weakest baseline performance with relatively high MAPE percentages and low R^2 scores on the original dataset. However, the performance of the regression models is comparably stable on the synthetic datasets, and even the DataSynthesizer with Differential Privacy (DS 0.1) shows results close to real data. Particularly interesting is its exceptional performance for the Support Vector Regression, which is very close to the best results.

Finally, Table V presents the results for the Insurance dataset. It shows good results for the DataSynthesizer without Differential Privacy (DS 0) and synthpop, mediocre results for the DataSynthesizer with Differential Privacy (DS 0.1), and the worst results for the Synthetic Data Vault. For the latter, the utility metric on the regression task is approximately 50% worse than on the other synthetic datasets. The difference seems the largest for the MLP regression, while it is not so pronounced on the Support Vector Regression.

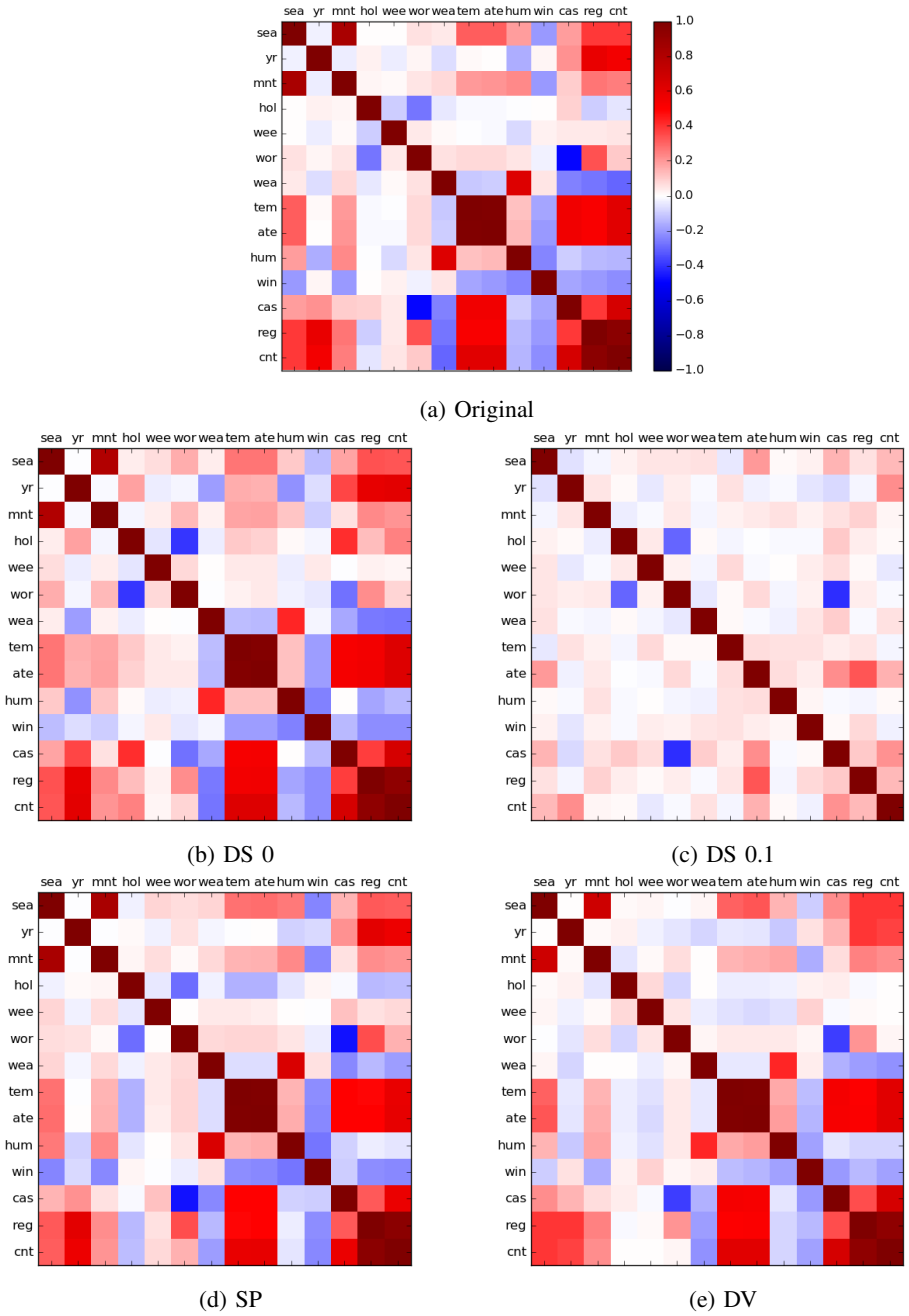
VI. Privacy Assessment

The three tools we used for synthetic data generation offer a variety of privacy-preserving measures. The conclusion in [7] was that there appears to be a trade-off between utility for machine learning and the degree of privacy. For this reason, the scores of different tools should not be compared directly. We therefore complemented the utility evaluation by assessing the privacy guaranteed by the synthetic datasets.

The first step of our experiment is the measurement of the distance between synthetic data samples and original samples, which has also been conducted in the evaluation of the DataSynthesizer and the Synthetic Data Vault for classification tasks in [7]. Our basic assumption is that privacy is endangered if individuals like the ones in the original dataset appear also in the synthetic dataset. We thus want to identify whether there are any individuals (i.e. rows) in the original dataset that also appear in the synthesised datasets. We are interested in both exact fits as well as similarities, as those might still contain some information that could lead to a privacy breach. For each row in the synthesised datasets, we therefore computed the nearest neighbour in the original dataset, that is, the row to which the distance is minimal. We chose the well-known euclidean distance as metric. To transform categorical attributes into comparable numerical values, we used label encoding followed by a step of feature scaling.

In Table VI, we consider the mean minimal distance for each synthetic dataset. This is computed as the mean over the distances between each synthetic sample and the respective nearest neighbour in the original dataset. The cells also show mean and standard deviation for the ten different splits.

In three cases, namely on Boston Housing, Bike Sharing and the Insurance dataset, synthpop produced the lowest score,



(a) Original

(b) DS 0

(c) DS 0.1

(d) SP

(e) DV

Fig. 1: Heat maps of attribute correlations for the Bike Sharing dataset

while for Social Network Ads, DS 0 produced the dataset with the lowest distance to the original. We can further see that, compared to the DataSynthesizer without Differential Privacy (DS 0) and synthpop, the DataSynthesizer with Differential Privacy (DS 0.1) and the Synthetic Data Vault produce samples with larger differences to the original. In particular, the mean minimal distance of the Synthetic Data Vault on the Insurance dataset, being roughly double as high as for the DataSynthesizer and synthpop, may explain its particularly weak performance on this dataset. On the other hand, the Synthetic Data Vault also exhibits a large distance for the Social Network Ads dataset, approximately double as high as for the other methods, while the utility evaluation on that dataset showed the performance on the Synthetic Data Vault dataset to be almost on par with the best techniques, except for a few cases.

In addition to the tabular representation, we also plotted histograms for one of the splits. The following figures show the minimal distance of synthetic samples on the x-axis, and the number of samples on the y-axis. As representative example, we present the graphs for the Boston Housing dataset in Figure 2, where we utilised a bin size of 40 for generating

TABLE II: Boston Housing

Method	Linear Regression			SV Regression			MLP Regression		
Measure	MAE	MAPE	R^2	MAE	MAPE	R^2	MAE	MAPE	R^2
Real	3.5±0.4	18±2	.71±.08	2.3±0.3	12±2	.86±.05	2.7±0.4	15±2	.81±.07
DS 0	3.9±0.3	19±3	.65±.08	3.6±0.4	18±2	.68±.08	5.2±0.9	25±3	.29±.19
DS 0.1	9.3±0.8	57±7	-.44±.30	11.0±2.3	66±15	-1.26±.78	29.3±4.1	163±24	-22.63±9.95
DV	3.6±0.3	18±2	.69±.07	4.7±0.4	24±3	.50±.08	6.0±0.5	33±3	.26±.16
SP	3.8±0.5	19±4	.64±.10	3.4±0.3	17±2	.71±.09	5.0±0.7	25±4	.43±.18

TABLE III: Bike Sharing

Method	Linear Regression			SV Regression			MLP Regression		
Measure	MAE	MAPE	R^2	MAE	MAPE	R^2	MAE	MAPE	R^2
Real	196±134	6±4	.98±.02	453±73	20±4	.88±.02	196±133	6±4	.98±.02
DS 0	364±96	11±3	.94±.03	594±67	24±5	.81±.03	610±89	21±4	.76±.11
DS 0.1	1460±100	49±7	.14±.11	1484±48	51±6	.13±.05	2870±174	111±14	-3.72±1.36
DV	255±114	8±4	.97±.02	616±71	27±6	.80±.03	257±112	8±4	.97±.02
SP	205±134	6±4	.98±.02	517±80	22±5	.85±.03	320±99	11±3	.95±.02

TABLE IV: Social Network

Method	Linear Regression			SV Regression			MLP Regression		
Measure	MAE	MAPE	R^2	MAE	MAPE	R^2	MAE	MAPE	R^2
Real	25481±1852	59±4	.12±.06	27216±2121	60±5	.01±.02	26396±1925	56±5	-.02±.29
DS 0	25645±1950	58±4	.11±.05	27160±2083	61±5	.01±.01	31370±6035	60±9	-.36±.37
DS 0.1	25703±1516	60±4	.08±.09	27119±2039	62±5	.02±.01	31688±5044	69±7	-.46±.39
DV	26082±1975	58±3	.09±.03	27273±2222	60±4	.01±.02	32437±8610	87±62	-.32±.41
SP	25578±1876	59±4	.11±.06	27206±2027	60±5	.01±.02	27917±3649	61±5	-.21±.38

TABLE V: Insurance

Method	Linear Regression			SV Regression			MLP Regression		
Measure	MAE	MAPE	R^2	MAE	MAPE	R^2	MAE	MAPE	R^2
Real	4135±189	44±4	.76±.03	5280±239	40±4	.31±.03	3191±394	38±6	.82±.04
DS 0	4414±199	51±6	.75±.03	5718±237	59±7	.32±.03	4057±429	60±10	.75±.06
DS 0.1	5662±325	90±11	.62±.04	6936±244	96±10	.24±.02	6730±532	101±13	.38±.11
DV	6369±277	64±8	.34±.04	7475±256	85±9	.02±.02	12586±2765	132±29	-1.08±1.12
SP	4168±199	44±6	.75±.03	5441±211	46±6	.31±.03	3833±721	44±9	.78±.10

TABLE VI: Mean Minimal Distances

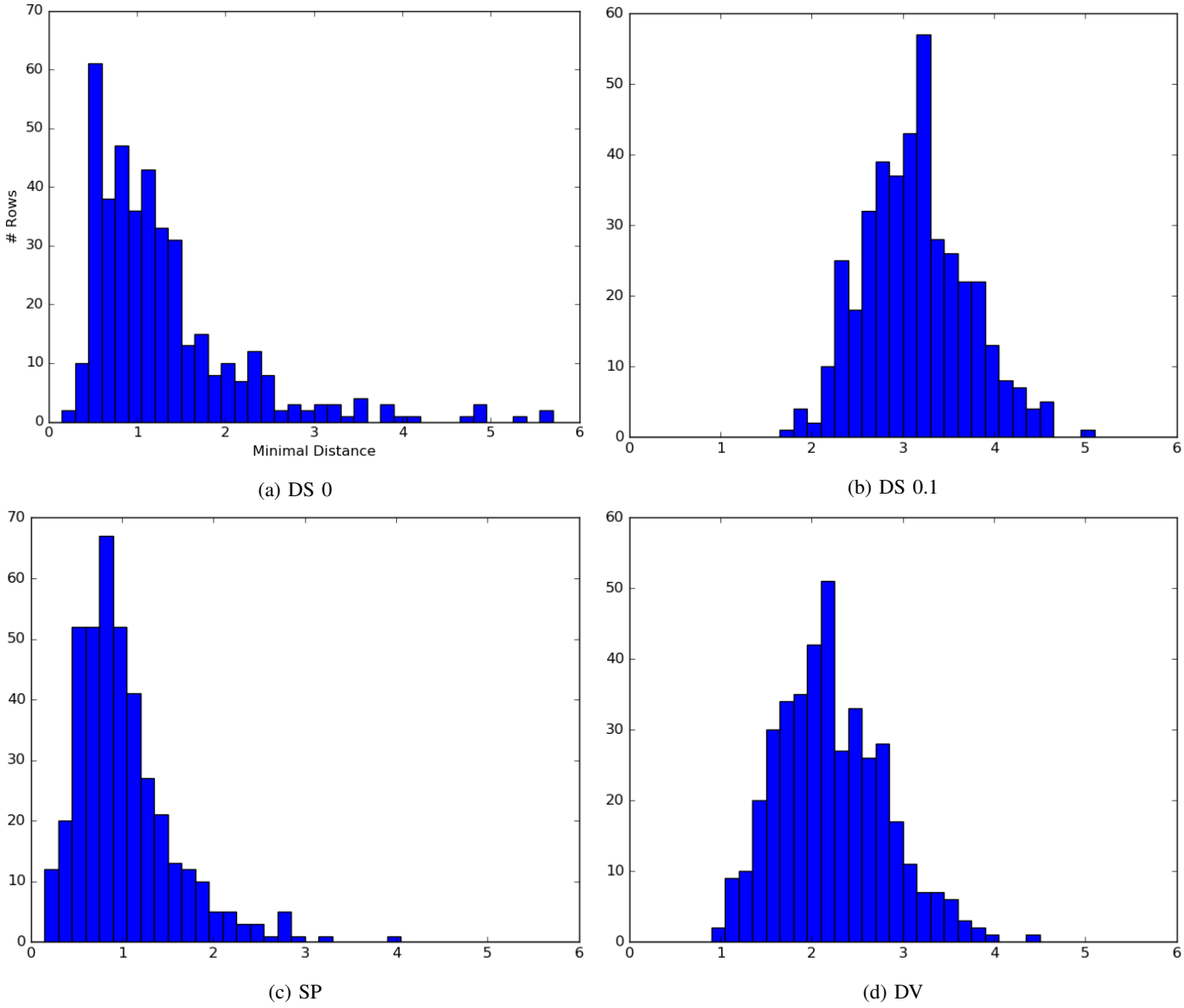
Distance	Boston	Bike	Network	Insurance
DS 0	1.22±0.14	1.28±0.02	**0.14±0.02**	0.47±0.01
DS 0.1	3.13±0.03	2.64±0.02	0.21±0.01	0.62±0.01
DV	2.27±0.01	1.96±0.02	0.39±0.03	0.89±0.01
SP	**1.04±0.04**	**1.18±0.02**	0.16±0.02	**0.45±0.02**

the histogram. From a visual inspection, the graphs for the DataSynthesizer without Differential Privacy (DS 0) and for synthpop are rather similar, especially on the very close samples, with the former having a slightly longer tail on the right side. On the other hand, the samples in the Synthetic Data Vault are on average only marginally closer to the real dataset than for the DataSynthesizer with Differential Privacy enabled (DS 0.1).

The second step of our empirical privacy evaluation consists of an application of the procedure discussed in Section III. We have seen that the attacker's approach assumed in the concept of Correct Attribution Probability can be extended to also handle continuous attributes. In these cases, the procedure then corresponds to the application of a Radius-Nearest-Neighbour Regression algorithm (RNNR). In the following, we used the scikit-learn implementation[6] of RNNR. We chose the euclidean distance as metric Δ, and considered a weighted mean for prediction, where the weight points are given by the inverse of the distance between $K_{o,j}$ and $a\big|_K$. This way, those a in the equivalence class that are closer to $K_{o,j}$ have a much greater influence on the resulting prediction, and the choice of the radius ρ is not as important. In fact, we chose $\rho = \infty$ in our experiments and hereby considered all the rows in the tables.

We conducted our first experiment on the Insurance dataset. The goal is to investigate the attacker's possibility of discovering the value of some (continuous) target attribute, given their knowledge about the values of several other variables, often

[6]https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.RadiusNeighborsRegressor.html



(a) DS 0

(b) DS 0.1

(c) SP

(d) DV

Fig. 2: Histogram of minimal distances of samples to the closest neighbour in the original data, for the Boston Housing dataset

referred to as *quasi-identifiers*. For the sake of this demonstration, we considered the following scenario, subsequently referred to as Scenario 1:

- QI = {'age', 'sex', 'children', 'smoker', 'region'}
- Target T='bmi'
- Keylength $k = 3$

An attribute key length of 3 means that, in the following procedure, we assume that the attacker knows the victim's values of three of the five variables in QI. We will consider all $C(5, 3) = 10$ resulting combinations.

Let $\mathcal{D}$ be the Insurance dataset. For the set QI, the target 'bmi' and key length 3, we performed the following steps.

1) Generate four synthesised versions of $\mathcal{D}$ of equal length:
   - The DataSynthesizer without Differential Privacy
   - The DataSynthesizer with Differential Privacy ($\varepsilon = 0.1$)
   - The Synthetic Data Vault
   - The synthpop package
2) Compute all 3-element subsets of the quasi identifiers QI of the respective scenario. Each subset corresponds to an attribute key used in the following step.
3) For each attribute key K:
   - For each record r in $\mathcal{D}$, compute the prediction for 'bmi' by applying RNNR to $r\big|_K$ on $\mathcal{D}\big|_{K,T}$ and on $\mathcal{S}\big|_{K,T}$ for each synthetic dataset $\mathcal{S}$.
   - Compute the MAE, the MAPE and the R^2 score of the results of all records r.
4) For each dataset, compute the mean and the standard deviation of these scores over all attribute keys.

The results for Scenario 1 are summarised in Table VII. Un-

surprisingly, the best scores and, hence, the highest disclosure risks may be observed on the original dataset. For MAE and MAPE, however, the difference between the original and the synthetic datasets is not as large as one might expect. Compared to DS0 and SP, the attribute disclosure risk in the considered scenario seems to be slightly smaller on DS 0.1 and DV.

TABLE VII: RNNR on Insurance / Scenario 1

DisclosureRisk	MAE	MAPE	R2
Original	4.23±0.5	14.59±1.73	.21±.14
DS 0	4.79±0.06	16.6±0.18	.05±.03
DS 0.1	4.83±0.07	16.98±0.24	.04±.03
DV	4.88±0.01	16.92±0.04	.01±.0
SP	4.77±0.07	16.45±0.21	.05±.03

Our application of RNNR resulted from the endeavour to generalise the concept of CAP to continuous variables. The attacker's problem, namely using the known values of the attribute key to obtain a prediction for the target variable, may also be solved by other regression algorithms. On the one hand, the MAE, MAPE and R^2 scores obtained on the synthetic datasets should be compared to the scores obtained on the original data, which establish an upper bound for the disclosure risk. On the other hand, any regression algorithm is only really useful if it beats the *DummyRegressor*[7], which always predicts the mean of the target variable in the respective dataset. In Table VIII, we summarised the scores of this approach for Scenario 1. Note that the procedure of the dummy regressor is independent of the attribute key K.

TABLE VIII: DummyRegressor on Insurance / Scenario 1

DisclosureRisk	MAE	MAPE	R2
Original	4.9	16.99	.0
DS 0	4.9	16.99	.0
DS 0.1	4.93	17.35	-.0069
DV	4.9	17.0	-.0001
SP	4.89	16.87	-.0005

We can see that RNNR beats the DummyRegressor in any single cell of the tables and, hence, constitutes an improvement from the attacker's perspective.

We complement this evaluation by conducting a second experiment on the SocialNetwork dataset. Let us consider Scenario 2:

- QI = {'Gender', 'EstimatedSalary', 'Purchased'}
- Target T='Age'
- Keylength $k = 2$

The results of RNNR are summarised in Table IX. We can see that the disclosure risk on DS 0 is even higher than on the original data. While SP is also very close to the scores obtained on the original data, the disclosure risk on DS 0.1 and DV is significantly smaller.

[7]https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyRegressor.html

TABLE IX: RNNR on SocialNetwork / Scenario 2

DisclosureRisk	MAE	MAPE	R2
Original	7.08±0.9	21.56±2.69	.27±.17
DS 0	6.98±0.91	21.4±2.73	.29±.17
DS 0.1	7.88±0.37	24.54±0.99	.12±.07
DV	7.91±0.4	23.74±1.25	.11±.09
SP	7.21±0.99	21.77±2.95	.25±.19

Considering the DummyRegressor for Scenario 2, we observe that the attacker's achieved improvement by the application of RNNR is much higher than it was in Scenario 1.

TABLE X: DummyRegressor on SocialNetwork / Scenario 2

DisclosureRisk	MAE	MAPE	R2
Original	8.45	25.63	-.0011
DS 0	8.47	25.9	-.0042
DS 0.1	8.51	26.42	-.0155
DV	8.42	25.24	-.0002
SP	8.44	25.45	-.0001

VII. Conclusions and Future Work

Comparing the experiment of the present paper to the analysis of the utility of synthetic data for classification tasks in [7], we observe two interesting differences. First, the DataSynthesizer with Differential Privacy appears to be less suitable for regression than it is for classification. Second, the Synthetic Data Vault seems to be best suitable for specific situations. We refer to the Bike Sharing dataset and to Linear Regression on Boston Housing. This is noteworthy because of the Synthetic Data Vault's tendency to create data with large differences to the original, which is demonstrated in Table VI. On average, the performance scores for the DataSynthesizer without Differential Privacy and synthpop appear to be closest to the original, and both tools are recommended for Linear and Support Vector Regression. Still, Table VI and Table IX raise privacy concerns for these synthesizers.

Concerning our privacy assessment using the Radius Nearest Neighbour Regression algorithm, we conclude that the attacker is able to obtain predictions that, on average, may be much closer to the true target value than predictions obtained by standard statistics such as the DummyRegressor. It follows that the task of estimating attribute disclosure on fully synthetic data (or on corresponding models) is particularly relevant whenever the comprised information and the correlations in the original data are *not* publicly known. Comparing the utility and privacy results of DS 0 and SP to the results of DS 0.1 and DV, we are able to confirm the general conclusion of [7], namely that there is a trade-off between utility and privacy. Obviously, our empirical analysis is restricted to the considered datasets and scenarios. For this reason, this paper does not intend to provide general benchmarks for the utility or privacy that warrant the publication of a synthetic dataset. It is rather a presentation of techniques any data provider can apply to check if their synthetic data fits specific, but often varying requirements. For example, RNNR might be used to estimate

the disclosure risk of a particularly sensitive attribute among the predictors; either for single records or, as demonstrated in Section VI, for the whole dataset.

It has to be mentioned that synthpop has several parameters for adding noise to the produced synthetic samples and even comes with its own function for statistical disclosure control. We have not used these options in our experiments, and their influence on the utility of the resulting data for both classification and regression is certainly an interesting subject for further investigations. Another interesting observation concerns the MLP Regression model. Even for the DataSynthesizer without Differential Privacy and synthpop, the scores are rather far from the original data. Likewise, the Synthetic Data Vault and the DataSynthesizer with Differential Privacy show the same pattern and, with few exceptions, perform worse on MLP than on Linear or Support Vector Regression. The reasons for this decrease of utility on more complex regression algorithms will also be a topic for future research.

REFERENCES

[1] B.-C. Chen, D. Kifer, K. LeFevre, and A. Machanavajjhala, "Privacy-preserving data publishing," *Foundations and Trends in Databases*, vol. 2, no. 1–2, pp. 1–167, January 2009.

[2] C. Dwork, "Differential privacy," in *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming ICALP*, ser. Lecture Notes in Computer Science, vol. 4052. Venice, Italy: Springer, July 10-14 2006, pp. 1–12.

[3] M. Elliot, "Final report on the disclosure risk associated with the synthetic data produced by the sylls team," University of Manchester, Tech. Rep., 2014. [Online]. Available: http://hummedia.manchester.ac.uk/institutes/cmist/archive-publications/reports

[4] H. Fanaee-T and J. Gama, "Event labeling combining ensemble detectors and background knowledge," *Progress in Artificial Intelligence*, vol. 2, no. 2, pp. 113–127, Jun 2014.

[5] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Computing Surveys*, vol. 42, no. 4, pp. 14:1–14:53, June 2010.

[6] D. Harrison Jr and D. L. Rubinfeld, "Hedonic housing prices and the demand for clean air," *Journal of environmental economics and management*, vol. 5, no. 1, pp. 81–102, 1978.

[7] M. Hittmeir, A. Ekelhart, and R. Mayer, "On the utility of synthetic data: An empirical evaluation on machine learning tasks," in *14th International Conference on Availability, Reliability and Security (ARES 2019)*, Canterbury, United Kingdom, August 26–29 2019.

[8] B. Lantz, *Machine learning with R*. Packt Publishing Ltd, 2013.

[9] B. Malle, P. Kieseberg, and A. Holzinger, "Do not disturb? classifier behavior on perturbed datasets," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, 8 2017.

[10] B. Nowok, G. Raab, and C. Dibben, "synthpop: Bespoke creation of synthetic data in r," *Journal of Statistical Software, Articles*, vol. 74, no. 11, pp. 1–26, 2016. [Online]. Available: https://www.jstatsoft.org/v074/i11

[11] N. Patki, R. Wedge, and K. Veeramachaneni, "The synthetic data vault," in *Proceedings of the IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, Montreal, QC, Canada, October 17–19 2016, pp. 399–410.

[12] H. Ping, J. Stoyanovich, and B. Howe, "Datasynthesizer: Privacy-preserving synthetic datasets," in *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, Chicago, IL, USA, June 27–29 2017.

[13] J. P. Reiter, Q. Wang, and B. Zhang, "Bayesian estimation of disclosure risks for multiply imputed, synthetic data," *Journal of Privacy and Confidentiality*, vol. 6, no. 1, 2014.

[14] D. B. Rubin, *Multiple Imputation for Nonresponse in Surveys*. Wiley-Interscience, Jue 2004.

[15] H. Surendra and H. S. Mohan, "A review of synthetic data generation methods for privacy preserving data publishing," *International Journal of Scientific & Technology Research*, vol. 6, no. 3, pp. 95–101, March 2017.

[16] L. Sweeney, "K-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, Oct. 2002.

[17] J. Taub, M. Elliot, M. Pampaka, and D. Smith, "Differential correct attribution probability for synthetic data: An exploration," in *International Conference on Privacy in Statistical Databases*. Valencia, Spain: Springer, September 26–28 2018, pp. 122–137.