THE NATIONAL UNIVERSITY
of SINGAPORE

School of Computing
Computing 1, 13 Computing Drive, Singapore 117417

TRA6/18

A comparative study of synthetic dataset generation techniques

Ashish Dandekar, Remmy A. M. Zen and Stéphane Bressan

June 2018

T e c h n i c a l R e p o r t

Foreword

This technical report contains a research paper, development or tutorial article, which has been submitted for publication in a journal or for consideration by the commissioning organization. The report represents the ideas of its author, and should not be taken as the official views of the School or the University. Any discussion of the content of the report should be sent to the author, at the address shown on the cover.

Mohan KANKANHALLI
Dean of School

A comparative study of synthetic dataset generation techniques

Ashish Dandekar[1], Remmy A. M. Zen[1], Stéphane Bressan[1]

National University of Singapore, Singapore
(ashishdandekar, remmy)@u.nus.edu, steph@nus.edu.sg

Abstract. Unrestricted availability of the datasets is important for the researchers to evaluate their strategies to solve the research problems. While publicly releasing the datasets, it is equally important to protect the privacy of the respective data owners. Synthetic datasets that preserve the utility while protecting the privacy of the data owners stands as a midway.
There are two ways to synthetically generate the data. Firstly, one can generate a fully synthetic dataset by subsampling it from a synthetically generated population. This technique is known as fully synthetic dataset generation. Secondly, one can generate a partially synthetic dataset by synthesizing the values of sensitive attributes. This technique is known as partially synthetic dataset generation. The datasets generated by these two techniques vary in their utilities as well as in their risks of disclosure. We perform a comparative study of these techniques with the use of different dataset synthesisers such as linear regression, decision tree, random forest and neural network. We evaluate the effectiveness of these techniques towards the amounts of utility that they preserve and the risks of disclosure that they suffer.

Keywords: synthetic data, random forest, decision tree, risk of disclosure, privacy

1 Introduction

On one hand, the philosophy of open data dictates that if the valuable datasets are made publicly available, the problems can be crowdsourced in the expectation to obtain the best possible solution. On the other hand, business organizations have their concerns regarding the public release of the datasets which may lead to the breach of private and sensitive information of stakeholders. In order to mitigate the risk of confidentiality breach, agencies employ different techniques such as reordering or recoding of sensitive variables, shuffling values among different records. In spite of these efforts by agencies, we have examples of confidentiality breaches in anonymised datasets. For instance, identification of the medical records of Massachusetts governor in an anonymised dataset [19], privacy breach at Netflix one million dollar challenge that led to litigations [10].

Analysts and researchers use datasets to validate their hypotheses. In order to have faithful analyses, publicly released datasets need to retain the utility

from the original dataset. By utility, we expect retention of the relationships among different attributes as well as retention of the distribution of values for each attribute. In order to achieve privacy, noise introduced by the privacy-preserving mechanisms deteriorates the utility of the dataset. Therefore, if one keeps privacy of the dataset as the sole objective, utility of the datasets is highly compromised. Therefore, there is a need of a way to generate datasets that can be made publicly available with minimum risk of disclosure and maximum utility.

Fully synthetic datasets proposed by Rubin [17] and partially synthetic datasets proposed by Little [8] bridge the gap between privacy and utility. They use multiple imputation, a technique used for repopulating the missing values in a dataset, to generate synthetic records which preserve relationships in the population. Following up on these works of multiple imputation, Reiter et al. [1, 6, 13, 15] use different machine learning tools to generate synthetic datasets. These works treat values of synthetically generated attributes as missing values that are generated using models such as Decision Trees, Random Forest, Support Vector Machine, etc.

We comparatively evaluate synthetic dataset generation techniques using different dataset synthesisers: namely Linear Regression, Decision Tree, Random Forest and Neural Network. We evaluate their effectiveness in terms of utility retention and risk of disclosure. We evaluate their efficiency in terms of time required to generate synthetic datasets. Given the tradeoff between the efficiency and effectiveness, we observe that Decision Trees are not only efficient but also competitively effective compared to other dataset synthesisers.

In Section 2, we present the related work. Section 3 introduces the formalism of synthetic dataset generation using multiple imputation. We present different dataset synthesisers in Section 4 followed by experiments and evaluation in Section 5. Section 6 concludes the work by discussing the insights and the extensions to the existing work.

2 Related Work

Synthetic dataset generation work stems from the early works of data imputation to fill in the missing values in the surveys [16]. In [17], Rubin proposes a procedure to generate fully synthetic dataset that uses multiple imputation technique to synthetically generates values for a set of attributes for all datapoints in the dataset. Although it is advantageous to synthetically generate values for all datapoints, it is not always a necessity. Partially synthetic datasets, proposed by Little [8], are generated by synthetically generating the values of the attributes that are sensitive to public disclosure. Various dataset synthesisers such as decision tree [15], random forest [1], support vector machine [2] have been used to generate fully and partially synthetic datasets. Drechsler et al. [6] have performed an empirical comparative study between different dataset synthesisers. Comparison between fully and partially synthetic datasets can be found in [4]. Recently, Nowok et al. [11] have created an R package, *synthpop*, which pro-

vides basic functionalities to generate synthetic datasets and perform statistical evaluation.

The effectiveness of the synthetic dataset lies in the amount of utility it retains from the original dataset. Most of the works [1, 6, 13, 15] use statistical methods of estimation for the evaluation of utility. They use estimators of mean and variance to calculate confidence intervals. Regression analyses are used to test whether the relationships among different variables are preserved. Aside from these analysis specific measures, Woo et al. [20] and Karr et al. [7] have proposed global measures such as Kullback-Leibler (KL) divergence, extension of propensity score, cluster analysis measure.

One of the prime motivations behind publicly releasing synthetic dataset instead of original datasets is to maintain the privacy of the data owners. In [14], Reiter introduces formalism to calculate risk of disclosure in synthetically generated datasets using multiple imputation. The same formalism has been used in [6, 15] to evaluate the risk of disclosure. For further details, readers are requested to refer to [3].

In this work, we comparatively evaluate efficiency and effectiveness of fully and partially synthetic dataset generation techniques using different dataset synthesisers including neural networks.

3 Synthetic dataset Generation using Multiple Imputation

In this section, we describe the general procedure of synthetic dataset generation using multiple imputation. Firstly, we introduce the terminology used for multiple imputation which is further used to explain full and partial synthetic dataset generation.

3.1 Multiple Imputation

We follow the formalism proposed by Drechsler [4] to explain the idea of multiple imputation.

Consider a dataset of size n sampled from a population of size N. Let Y_{nobs} denote subset of attributes in the dataset whose values are either missing for some datapoints or sensitive towards the public disclosure. Rubin [16] proposes to synthetically generate values for Y_{nobs} given the knowledge of rest of the attributes in the dataset, say Y_{obs}.

Let, $\mathcal{M}$ be a dataset synthesiser that generates values for an attribute Y_i given the information about rest of the attributes, denoted as Y_{-i}. With the help of $\mathcal{M}$, an imputer independently synthesises values of Y_{nobs} m times and releases m synthetic datasets $\mathcal{D} = \{\mathcal{D}^1, \mathcal{D}^2, ..., \mathcal{D}^m\}$. Each dataset D^l comprises of datapoints with the values for the attributes in Y_{obs} same as the original dataset and datapoints with synthetically generated values for variables in Y_{nobs}.

In order to synthesise multiple sensitive attributes, we follow the procedure presented in [2]. Suppose that we want to synthetically generate values all of

the attributes except an attribute Y_0. The procedure to synthetically generate values of multiple attributes is as follows:

1. Generate values for Y_1 using known Y_0. Let, $Y_1^{(syn)}$ denote synthetically generated values for Y_1.
2. Generate values for Y_2 using Y_0 and Y_1. (Y_0, Y_1) is used to train the model $\mathcal{M}$ whereas $(Y_0, Y_1^{(syn)})$ is used to generate $Y_2^{(syn)}$.
3. Generate values for Y_i for $i = 3, 4, ...$ using $(Y_0, Y_1, ..., Y_{i-1})$.

This procedure is repeated m times to generate m synthetic datasets.

Suppose that we want to statistically estimate of an attribute Q. Let, q_l and v_l denote sample mean and sample variance of Q using the dataset D^l. Let, $\bar{q_m}$, b_m and $\bar{v_m}$ denote mean of the sample means, "between imputation variance" and "within imputation variance" over m datasets respectively. They are calculated as defined in Equation 1.

$$\begin{aligned} \bar{q_m} &= \frac{1}{m}\sum_{l=1}^{m} q_l \\ b_m &= \frac{1}{m-1}\sum_{l=1}^{m}(q_l - \bar{q_m})^2 \\ \bar{v_m} &= \frac{1}{m}\sum_{l=1}^{m} v_l \end{aligned} \tag{1}$$

The reason behind releasing m different datasets and combining estimators on each dataset is two folds. Firstly, there is randomness in the dataset due to sampling from the population. Secondly, there is randomness in the dataset due to imputed values. In order two capture these variabilities, framework of multiple imputation proposes the release of m datasets.

3.2 Fully Synthetic Dataset Generation

We follow the formalism proposed by Rubin [17] to generate fully synthetic datasets.

Consider a dataset of size n sampled from a population of size N. Suppose that an imputer knows the values of a set of variables X for the entire population and values for rest of the variables, Y, only for a selected small sample. Let, Y_{inc} and Y_{exc} denote values of variables which are included in the sample and excluded from the sample respectively. The imputer synthetically generates values of Y_{exc} using a dataset synthesizer $\mathcal{M}$ trained on Y_{inc} and X. This synthesis is equivalent to performing multiple imputation with Y_{exc} as Y_{nobs} and Y_{inc} as Y_{obs}. Publicly released datasets, $\mathcal{D}$, comprise of m samples selected synthetically generated population.

Suppose that we want to statistically estimate of an attribute Q. We use estimator of the mean and estimator of the variance presented in [13]. The sample mean, $\bar{q_m}$, is the estimator of the mean. For smaller values of $m(< 30)$, the

estimator of variance of Q, T_f, follows Student's t-distribution with ν_f degrees of freedom. The estimator is calculated using Equation 2.

$$\begin{aligned} T_f &= \frac{b(m+1)}{m} - \bar{v_m} \\ \nu_f &= (m-1) * (1 - r^{-1})^2 \\ r &= \frac{b_m(1+m^{-1})}{v_m} \end{aligned} \tag{2}$$

Theoretically, fully synthetic datasets provides 100% guarantee against the disclosure of value of sensitive attribute [17]. Since $n << N$, it is less probable to have record from the original sample in the final dataset. Final datasets are sampled from synthetic population datasets in which $N-n$ records are synthetically generated.

3.3 Partially Synthetic Dataset Generation

We follow the formalism proposed by Reiter [13] to generate partially synthetic datasets.

Let S be a dataset of size n sampled from a population of size N. In order to protect the sensitive information, an imputer decides to alter values of a set of attributes, Y, for a subset of datapoints in S. Let Z be a binary vector of size n. Z_i takes value one if Y values of the i^{th} datapoint are to be synthetically generated and Z_i takes value zero if values of Y attributes are not be altered.

Let $Y_{syn} = \{Y_i | \forall i, Z_i = 1\}$ and $Y_{org} = \{Y_i | \forall i Z_i = 0\}$. We generate m partially synthetic datasets by multiple imputation. In this case, Y_{syn} are the datapoints, with missing values, that we synthetically generate by training a dataset synthesiser on the available data, i.e Y_{org}. This synthesis is equivalent to performing multiple imputation with Y_{syn} as Y_{nobs} and Y_{org} as Y_{obs}. Publicly released datasets, $\mathcal{D}$, comprise of m datasets sampled from the population wherein values of attributes in Y are synthetically generated, as specified by Z, for each of the dataset.

Suppose that we want to statistically estimate of an attribute Q. We use estimator of the mean and estimator of the variance presented in [12]. The sample mean, $\bar{q_m}$, is the estimator of the mean. For smaller values of $m(< 30)$, the estimator of variance of Q, T_p, follows Student's t-distribution with ν_f degrees of freedom. The estimator is calculated using Equation 3.

$$\begin{aligned} T_p &= \bar{v_m} + \frac{b}{m} \\ \nu_p &= (m-1) * \left(1 + \frac{m\bar{v}_m}{b}\right)^2 \end{aligned} \tag{3}$$

4 Dataset synthesisers

As described in Section 3, method of multiple imputation generates values for an attribute Y_i by using a dataset synthesiser $\mathcal{M}$ trained on the information about

rest of attributes Y_{-i}. In this section, we discuss different dataset synthesisers namely Linear Regression, Decision Tree, Random Forest and Neural Network.

4.1 Linear Regression

Linear regression [9] models relationships between a dependent attribute and one or more independent attributes. Linear regression models datapoints as samples from a Gaussian distribution, as specified in Equation 4, where β are the parameters that we learn using the training data. Please refer to [9] for detailed derivation.

$$P(Y_i|Y_{-i}) = \mathcal{N}(Y_i|\beta^T Y_{-i}, \sigma^2) \tag{4}$$

In order to generate synthetic data, for every dataset and for every attribute Y_i to be synthetically generated, we learn the parameters of the regression model using the dataset with attributes in Y_{-i}. We generate values by sampling from Gaussian distribution as given in Equation 4.

4.2 Decision Tree

Decision tree or Classification and Regression Tree (CART) models [9] are predictive models which work by recursively splitting the feature space, the space spanned by the values attributes in the dataset, into smaller spaces. In the beginning, all of the training examples belong to one feature space. CART chooses an attribute value on which the dataset can be split using metrics such as Gini index or information gain. Partitioning is repeated on every smaller feature subspace until there are no more than k datpoints left in the subspace. This process is represented using a tree structure wherein each node defines a conditional distribution of its members given the criteria that defines the partition.

We adopt Reiter [15] who proposes the use of CART to generate partially synthetic datasets. The procedure starts with building a decision tree using the values of the attributes that are available in the dataset Y_{-i}. In order to synthesise the value of an attribute Y_i for a datapoint j, we trace down the tree using the known attributes of j until we reach the leaf node. Let L_j be the set of values of Y_i in the leaf node. For a categorical attribute Y_i, Reiter proposes Bayesian bootstrap sampling to choose m different values. For a continuous attribute Y_i, we fit a kernel density estimator over the values in L_j and sample m values from the estimate.

4.3 Random Forest

Random forest is a kind of Ensemble learning technique. It uses multiple decision trees that are constructed on the samples of the training dataset and the final output is given by aggregating the result from individual tree. Each decision tree is constructed on a datapoints using a set of randomly selected attributes.

Caiola et al. [1] use random forest to generate partially synthetic datasets. In order to synthesise values for a certain attribute Y_i, they train a fixed number decision trees on random samples of training dataset Y_{-i}. For a categorical

attribute, the collection of results from constituent decision tree forms a multinomial distribution. m values are sampled from this distribution as the synthetic values for Y_i. For a continuous attribute, they propose use of a kernel density estimator over the results from decision trees and sample values from the estimator.

4.4 Neural Network

Neural network [9] is a machine learning model that learns an abstract function mapping an input to the corresponding output. Schematically, it consists of three layers namely an input layer, one or more hidden layers and an output layer. Each layer comprises of neurons wherein every neuron acts a signal emitter. Every neuron receives input from the previous layer and emits a single output that is weighted sum of inputs with an additive bias. These interconnection builds a network of neurons. The weights and biases for every neuron are learned using backpropagation algorithm. Details of the algorithm can be found in [9].

For K-class classification, there are K nodes in the output layer, with value at k-th neuron representing the probability of class k. We train a neural network using features in Y_{-i}. In order to synthesise value of an attribute Y_i, we sample a class value using the output layer neuron values as a multinomial distribution.

5 Empirical Evaluation

5.1 Dataset and Experimental Setup

We conduct experiments on a microdata sample of US Census in 2000 provided by IPUMS International [18]. The dataset consists of 1% sample of the original census data. It spans over 1.23 million households with records of 2.8 million people. It has several attributes of which not every single attribute is reported by all of the people. In order to avoid these discrepancies in the data, we follow the approach presented in [6] to consider the records of the heads of households. We treat this collection of the records of 316,276 households as the population. Table 1 shows the set of features which we consider for experiments.

All programs are run on Linux machine with quad core 2.40GHz Intel® Core i7™processor with 8GB memory. The machine is equipped with two Nvidia GTX 1080 GPUs. Python® 2.7.6 is used as the scripting language.

5.2 Evaluation of utility

The utility of generated dataset needs to be evaluated at two different levels. Firstly, we need to evaluate differences between the distribution of values of original attribute and synthetically generated attributes. Secondly, we need to evaluate the difference between the quality of statistical estimation of a certain attribute for synthetic dataset and generated data.

Let $y \in Y$ be any attribute that we synthetically generate from an original dataset. We calculate the similarity between the overall distribution of values

Attribute Name	Variable Type	Notes
House Type	Categorical	
Family Size	Ordinal	
Sex	Categorical	
Age	Ordinal	5.9% have age less than 26
Marital Status	Categorical	
Race	Categorical	
Educational Status	Categorical	
Employment Status	Categorical	
Income	Ordinal	7.13% have income more than 70000
Birth Place	Categorical	

Table 1. Dataset Description

of y by calculating **normalised KL-divergence** between the distribution of values of y in population and the distribution of synthetically generated values. For m synthetic datasets, we consider mean of the normalised KL-divergence over individual datasets. Closer the value to 1, more similar the synthetically generated values are to the original values.

Karr et al. [7] develop a mechanism based on **overlap** between confidence intervals to evaluate the effectiveness of a statistical estimator. We estimate mean and variance of y using the point estimators described Section 3. We construct a 95% confidence interval around the estimator. Let (L_s, U_s) be confidence interval for synthetically generate y and (L_o, U_o) be interval from original data. We compute intersection of these intervals denoted as (L_i, U_i). The overlap utility measure is calculated using Equation 5.

$$I = \frac{(U_i - L_i)}{2(U_o - L_o)} + \frac{(U_i - L_i)}{2(U_s - L_s)} \tag{5}$$

If the intervals are similar to each other, we say that the synthetic dataset generation procedure preserves the utility. The value of I is close to one if the utility is preserved and $I = 0$ refers to the dissimilar confidence intervals.

5.3 Evaluation of risk of disclosure

We follow Reiter [5, 13] to estimate the **risk of disclosure** in the synthetically generated dataset. Let $\mathbf{t}$ be a vector of information possessed by an intruder. We assume that the intruder has complete information about an auxiliary variable, say region of birth, which is not a sensitive variable. For instance, the intruder might be interested in an individual who is born in Nevada earning more than $70,000$\$ salary. For every datapoint j in the dataset, the intruder calculates the probability of the datapoint j being the record of interest by comparing the respective attributes as given in Equation 6. In Equation 6, $N_{(\mathbf{t},i)}$ denotes the number of records in dataset D^i that match target. $\mathbb{I}(Y_j^i = \mathbf{t})$ is the identity function that equals to 1 if j^{th} datapoint in the dataset D^i matches $\mathbf{t}$ otherwise 0.

$$Pr(J = j|D, \mathbf{t}) \;=\; \frac{1}{m}\sum_{i=1}^{m}\frac{1}{N_{(\mathbf{t},i)}}\mathbb{I}(Y_j^i = \mathbf{t}) \tag{6}$$

In the end, the intruder selects datapoints with maximum probability value. This process is repeated for every target datapoint in $\mathbf{t}$. For a datapoint $j \in \mathbf{t}$, an intruder may find multiple datapoints with the same value of maximum probability. Let, R denotes the set of datapoints in $\mathbf{t}$ for which only one datapoint in the dataset is matched with highest probability. Set R can be decomposed into two mutually exhaustive sets T and F that denote the set of datapoints with true matches and false matches respectively. In order to evaluate the risk of disclosure, we calculate *true match rate* and *false match rate*. They are calculated using Equation 7. The smaller the true match rate, better is the performance of a dataset synthesiser.

$$\begin{aligned} \text{true match rate} &= \frac{|T|}{|\mathbf{t}|} \\ \text{false match rate} &= \frac{|F|}{|R|} \end{aligned} \tag{7}$$

For further details about the calculation please refer to [5, 13].

5.4 Evaluation

The process starts by drawing 1% sample from the population, which we treat as the original dataset. We synthetically generate values for two attributes: income and age, in the same order. Interested readers can refer to [15] for a detailed discussion on choosing the order of synthesis. We generate 5 synthetic datasets for each original dataset. We repeat this procedure for 500 original datasets and mean of various metrics over 500 iterations is reported.

In order to generate partially synthetic datasets, we need to define the cutoffs for the values of attribute that determine quantify the sensitivity of the attribute towards disclosure. We consider datapoints that have more than 70000$ income value and less than 26 age value to be the ones with sensitive information. So, we synthetically generate values for age and income for the datapoints that fit these criteria. Utility evaluation results are presented in Table 2. In order to generate fully synthetic datasets, we generate values of age and income for all the records in the original dataset. Utility evaluation results are presented in Table 3. We observe that although two techniques show comparable values of synthetic means, the technique of partially synthetic dataset generation shows greater extent of the overlap with attribute distribution in the original dataset. Partially synthetic dataset generation does not replace all values of the attributes in the sample. Released datasets contains datapoints from the original dataset. Therefore, we observe higher overlap for partially synthetic datasets. In Table 2 and Table 3, we observe a large deviation in the sample mean of *age* from its original mean in case of linear regression. Linear regression learns parameters

such that squared loss on the training dataset is minimised and without any regularization it suffers from overfitting. As specified in Section 3, linear regression model is fit on the synthetically generated values of *income* while synthesising value for *age*. Linear regression fails to capture exact distribution of values in the original dataset. Thus, linear regression suffers from the order of in which attributes are synthesised. Decision tree and other models are not prone to overfitting the training dataset. Therefore, we do not observe such a degradation in utility.

Feature	**Data synthesisers**	**Original Sample Mean**	**Partially Synthetic Data**		
			Synthetic Mean	**Overlap**	**Norm KL Div.**
Income	Linear Regression	27112.61	27117.99	0.98	0.54
	Decision Tree	27143.93	27131.14	0.94	0.53
	Random Forest	27107.04	27254.38	0.95	0.58
	Neural Network	27069.95	27370.99	0.81	0.54
Age	Linear Regression	49.83	24.69	0.50	0.55
	Decision Tree	49.83	49.83	0.90	0.56
	Random Forest	49.82	49.74	0.95	0.56
	Neural Network	49.87	49.78	0.90	0.56

Table 2. Evaluation of utility for partially synthetic datasets generated using different dataset synthesisers.

Feature	**Data synthesisers**	**Original Sample Mean**	**Fully Synthetic Data**		
			Synthetic Mean	**Overlap**	**Norm KL Div.**
Income	Linear Regression	27112.61	27074.80	0.52	0.55
	Decision Tree	27081.45	27091.02	0.55	0.58
	Random Forest	27107.04	28720.93	0.54	0.64
	Neural Network	27185.26	26694.54	0.54	0.94
Age	Linear Regression	49.83	-192.21	0.50	0.56
	Decision Tree	49.83	49.83	0.56	0.56
	Random Forest	49.82	46.25	0.68	0.57
	Neural Network	49.76	54.32	0.75	0.99

Table 3. Evaluation of utility for fully synthetic datasets generate using different dataset synthesisers.

In order to evaluate the risk of disclosure, we require a scenario. We select the scenario by doing some exploratory analysis on the population. The datapoints that sparsely occur in the population, for instance people who are born in middle east with a certain income threshold, equally sparsely occur in a small sample. In order to statistically evaluate risk of disclosure, we need to have at least a handful of targets for evaluation.

Taking into account these requirements, we suppose that an intruder is interested in people who are born in US and have income more than $250,000$. All these people are the targets of the intruder. Intruder tries to match every single target with the records in the released datasets. We consider two records perfectly match if the people representing the records are born in US, they have income more than $250,000$ and the age of the person in dataset in within the tolerance of 2 compared to target person.

Two cases arise in the evaluation. For a given target, the intruder may or may not know if the target person is included in the released sample. If the target person is not included, matching probability is calculated in a different way. In such a case, instead of using $N_{(\mathbf{t},i)}$ in Equation 6, we use the number of datapoints in population that match with target t. Results are presented in Table 4. We observe that, in the case when the intruder does not have certainty about inclusion of target in the sample, risk of disclosure is the least. In most of the cases, the targets might not be present in the released sample which leads to true match rate of 0. Observing the results for the case when a target is present in the sample, we see that neural networks are comparatively giving better performance than rest of the dataset synthesisers.

Dataset synthesisers	**Target is in the sample**		**Target may be in the sample**	
	True MR	**False MR**	**True MR**	**False MR**
Linear Regression	0.06	0.82	0.00	0.00
Decision Tree	0.18	0.68	0.00	0.99
Random Forest	0.35	0.50	0.00	0.99
Neural Network	0.03	0.92	0.00	0.99

Table 4. Evaluation of risk of disclosure for different dataset synthesisers

We comparatively analyse efficiency of both these techniques using different dataset synthesisers. The running time, in seconds, for generating 5 synthetic datasets is reported in Table 5. We observe that the neural networks achieve the low risk of disclosure at the cost of a higher running time than the time taken by linear regression or decision trees.

dataset synthesiser	**Partially Synthetic Dataset Generation**	**Fully Synthetic Dataset Generation**
Linear Regression	0.040	0.068
Decision Tree	0.048	0.533
Random Forest	3.350	103.543
Neural Network	0.510	55.26

Table 5. Efficiency: Each cell shows the running time required, in seconds, to generate 5 synthetic datasets.

6 Discussion and Future Works

In this work, we comparatively evaluate fully and partially synthetic dataset generation techniques using different dataset synthesisers, namely linear regression, decision tree, random forest and neural network. We analyse the effectiveness using the overlap of generated values of attributes, such as estimators of mean and variance, with values of those attributes from original data. We also evaluate the distribution level similarity that captures the dataset statistics at global level. We address privacy concerns by calculating the risk of disclosure for synthetically generated datasets. The analysis shows that decision trees stand as a good dataset synthesiser given its high effectiveness compared to other data synthesisers. This observation agrees with the result in [6].

We use a well-structured dataset in this work. Many real-world datasets do not have a well defined structure. For instance, the social network datasets or the datasets generated from the readings collected by sensors. As a future work, we want to explore how synthetic dataset generation techniques can be adopted for such non-structured or semi-structured datasets.

References

1. Caiola, G., Reiter, J.P.: Random forests for generating partially synthetic, categorical data. Trans. Data Privacy 3(1), 27–42 (2010)
2. Drechsler, J.: Using support vector machines for generating synthetic datasets. In: Privacy in Statistical Databases. pp. 148–161. No. 6344, Springer (2010)
3. Drechsler, J.: Synthetic datasets for statistical disclosure control: theory and implementation, vol. 201. Springer Science & Business Media (2011)
4. Drechsler, J., Bender, S., Rässler, S.: Comparing fully and partially synthetic datasets for statistical disclosure control in the german iab establishment panel. Trans. Data Privacy 1(3), 105–130 (2008)
5. Drechsler, J., Reiter, J.P.: Accounting for intruder uncertainty due to sampling when estimating identification disclosure risks in partially synthetic data. In: International Conference on Privacy in Statistical Databases. pp. 227–238. Springer (2008)
6. Drechsler, J., Reiter, J.P.: An empirical evaluation of easily implemented, nonparametric methods for generating synthetic datasets. Computational Statistics & Data Analysis 55(12), 3232–3243 (2011)
7. Karr, A.F., Kohnen, C.N., Oganian, A., Reiter, J.P., Sanil, A.P.: A framework for evaluating the utility of data altered to protect confidentiality. The American Statistician 60(3), 224–232 (2006)
8. Little, R.J.: Statistical analysis of masked data. Journal of Official statistics 9(2), 407 (1993)
9. Murphy, K.P.: Machine learning: a probabilistic perspective. MIT press (2012)
10. Narayanan, A., Shmatikov, V.: Robust de-anonymization of large sparse datasets. In: Security and Privacy, 2008. SP 2008. IEEE Symposium on. pp. 111–125. IEEE (2008)
11. Nowok, B., Raab, G., Dibben, C.: synthpop: Bespoke creation of synthetic data in r. Journal of Statistical Software, Articles 74(11), 1–26 (2016)

12. Raghunathan, T.E., Reiter, J.P., Rubin, D.B.: Multiple imputation for statistical disclosure limitation. Journal of official statistics 19(1), 1 (2003)
13. Reiter, J.P.: Inference for partially synthetic, public use microdata sets. Survey Methodology 29(2), 181–188 (2003)
14. Reiter, J.P.: Estimating risks of identification disclosure in microdata. Journal of the American Statistical Association 100(472), 1103–1112 (2005)
15. Reiter, J.P.: Using cart to generate partially synthetic public use microdata. Journal of Official Statistics 21(3), 441 (2005)
16. Rubin, D.B.: Basic ideas of multiple imputation for nonresponse. Survey Methodology 12(1), 37–47 (1986)
17. Rubin, D.B.: Discussion statistical disclosure limitation. Journal of official Statistics 9(2), 461 (1993)
18. Ruggles, S., Genadek, K., Goeken, R., Grover, J., Sobek, M.: Integrated public use microdata series: Version 6.0 [dataset] (2015), http://doi.org/10.18128/D010.V6.0
19. Sweeney, L.: Computational disclosure control for medical microdata: the datafly system. In: Record Linkage Techniques 1997: Proceedings of an International Workshop and Exposition. pp. 442–453 (1997)
20. Woo, M.J., Reiter, J.P., Oganian, A., Karr, A.F.: Global measures of data utility for microdata masked for disclosure limitation. Journal of Privacy and Confidentiality 1(1), 7 (2009)