The Alan Turing Institute

Data Study Group Final Report: Department for Work and Pensions

12 – 30 Apr 2021

The assessment of utility and privacy of synthetic data

https://doi.org/10.5281/zenodo.5729466

Contents

1 Executive summary

1.1 Challenge overview

The Department for Work and Pensions (DWP) collects and administers a large quantity of sensitive personal data in the course of its day-to-day functioning. This includes, for example, information about demographic characteristics of individuals, whether they are in receipt of Universal Credit, and how much.

The DWP Innovation Lab have a particular interest in exploring how cutting-edge data analysis techniques might be applied to these datasets, and naturally this work is strengthened by collaborating with researchers and sharing data across services. However, the high degree of sensitivity of the data in question presents a great challenge to working collaboratively.

The difficulties and risks inherent in making such sensitive data more available to analysts and data scientists can mean long lead-times before data is made available, and work having to be conducted in restrictive computational environments. Anonymization and disclosure control techniques can permit a version of the data to be shared more widely: **Synthetic data** are a promising alternative or addition to standard anonymization procedures.

This challenge explored methods to gauge the suitability of synthetic data (including particular datasets provided by two commercial teams). The methods for synthesising data that we are considering start from an original, sensitive dataset. This raises two key questions. First: How well is the **privacy** of individuals present in the original dataset protected? (alternatively, how much can be inferred about the original dataset from the synthetic data?) Second: How suitable is the synthetic data as a substitute for the original data, for its intended uses? The latter we refer to as its **utility**.

This aim of this challenge is to explore these questions, with a focus on (but not limited to) several synthetic datasets provided by DWP, and how issues of privacy and utility trade off against one another.

1.2 Data overview

The DWP provided two datasets of Universal Credit claimant data. Both datasets were synthetic data, in fact; the challenge did not work with real sensitive personal data. These represent the original sensitive population, and are close in structure to existing DWP datasets: One of these datasets is cross-sectional and the other is longitudinal. Together they form the original data on which the synthetic data is based. DWP also provided a number of corresponding synthetic datasets generated by two commercial teams.

1.3 Main objectives

The objectives of the challenge were:

- To investigate synthetic data generation methods, and their suitability for use with the DWP data
- To better understand available measures of privacy when used to evaluate synthetic data, and under which settings they apply
- To use these measures to understand how well synthetic data techniques preserve the privacy of individuals in the original datasets
- To determine whether the synthetic datasets represent a reasonable trade-off between utility and privacy, by performing some representative analysis tasks on the synthetic datasets and comparing with the original data.

1.4 Approach

Initially, we explored the datasets (both original and synthetic) by plotting univariate distributions and correlations between features of the data. This served as a basic quality check of the data, and allowed simple comparisons to be made between the original and synthetic data.

To have additional 'baseline' datasets for comparison with the team synthetic datasets, we explored several techniques for generating synthetic data

from the literature. This also was a practical requirement to provide the quantity of synthetic data needed to evaluate privacy and utility metrics.

Given time and resource constraints, we focused on two approaches which appeared most suitable to our challenge: the Gaussian Copula method, which implements a simple statistical transformation, and CT-GAN, a deep learning based synthetic data generators (Generative Adversarial Network) model used specifically for single table data which are able to learn from real data and generate synthetic data. We used the implementations of these methods from the SDV (Synthetic Data Vault) framework [42].

The second component of the work was to develop an evaluation framework to assess both the privacy and utility of the synthetic data, and the trade-off between these two criteria. We explored a number of ways to define and measure privacy, implementing a set of metrics to evaluate the distributional similarity of the datasets.

To assess utility, we performed regression tests and investigated a set of modelling questions, proposed as part of the challenge, and considered representative of questions that the data might be used to investigate.

We augmented the evaluation framework to also consider the robustness, namely adversarial robustness, and fairness, in terms of equality of outcomes, of the synthetic data.

1.5 Main conclusions

Our evaluation of the team synthetic datasets for utility and the results of the modelling experiments are broadly in line with the results of the performance evaluation already conducted by DWP: We established that team B has produced superior distributional similarity when modelling dataset A, but that the similarity indices we establish are less convincing for this team with respect to dataset B. Indeed, team C produces superior results in some of our simulations of downstream tasks on their sets based on dataset B, which may indicate that team B's generation algorithm scales better to larger source datasets.

We also establish some intriguing initial results in fairness evaluation, which indicate that team C may have introduced additional sources of bias related to the gender of the respondent in their generation process. This finding requires additional research to validate, and tracking down the source of such additional bias would prove a fascinating field of work in and of itself.

We conclude from our limited exploration of synthetic generation that we obtain superior results from a tailored Gaussian Copula statistical process rather than a more general approach utilising neural network generation. We conclude that the CTGAN model we attempted to utilise produces more variation in the results and hence a lower level of distributional similarity with the original datasets. However, it is clear that fine tuning the hyperparameters of a GAN-based model such as CTGAN and increasing the size of the original dataset could prove beneficial to the observed performance level. For this reason, we would indicate that further experiments must be carried out on dataset A before we can confidently state the superiority of the Gaussian Copula method.

1.6 Limitations

We recognise a number of limitations in this work. Among them are the limited number of potential generative models we have assessed, as well as the textual features present in the original dataset that we were unable to synthesize.

We have explored only a very small number of the potential privacy metrics that have been proposed—there are certainly many more suitable candidate metrics discussed in the literature on this topic. Wagner and Eckhoff [59] provide an excellent overview of the field.

Notably excluded from the privacy evaluation was differential privacy, a general information privacy field which provides a measurable level of privacy by measuring the sensitivity of particular queries applied to the dataset and using noise to enforce single-record-level indistinguishability [10]. It would be

valuable future work to investigate the differential privacy properties of the synthetic generators (where these are known), or give empirical estimates of differential privacy from data, of which a few are documented in the literature. See section for more discussion.

1.7 Recommendations and future work

Future work on synthetic data evaluation could focus on the expansion of the techniques and metrics used to evaluate privacy provided by synthetic datasets—empirical estimates of differential privacy could be particularly interesting—as well as expanding the definition and scope of privacy risk. Future work should also include more diverse types of attack (e.g. membership inference). Finally, some novel evaluations techniques aiming to have a more easily-understood output deserve consideration in future work. Such is the case of table-to-text generations models, which emits easy-to-read structured text.

Generation of text fields based on conditional distribution of original data is a natural follow-up for this project, as it would increase the utility of the synthetic sets. We also strongly suggest incorporating differential privacy in future model-based synthetic data generations attempts, in order to establish strong row-level privacy from the model training step onwards. We suggest two promising approaches to include differential privacy: Local differential privacy by including perturbations in the sample training dataset and Private Attribute Adversarial Training, in which an adversarial model attempts to infer protected features from intermediate representations of the generative model, providing the latter with an objective measure to minimise.

2 Data overview

2.1 Dataset description

The data provided by DWP consisted of two 'real' datasets. These data were in fact generated by DWP using a microsimulation method for the purpose of the challenge.

Dataset A A household-level repeated cross-section survey of approximately 5 million observations. It consists of around 2.5 million observations each for the months of January and February 2020. This dataset is derived from Stat-Xplore, an open database of DWP benefit statistics (`https://stat-xplore.dwp.gov.uk`). In preparation for the challenge (both for the teams and DSG), this was augmented by DWP to include (synthetic) personal data, to add precision to binned variables, and to add deliberate errors (discussed below).

Dataset B A time series dataset of approximately 20 000 households, observed for 12 months from January to December 2020. No deliberate errors are included in this dataset.

Both datasets include the demographic characteristics of the household, including the address and family type (Single or Couple); the name, date of birth, age, National Insurance (NI) number, gender, employment status and occupation of each of up to two adult household members; number of children and, for each child, date of birth and an indicator for child disability; and quantity of benefits received, including the Standard Allowance, Carer Allowance, Housing Allowance and Child Allowance, and Total Deductions, Total Additions and the Total Allowance received. Dataset B also includes an additional variable of Disabled Child Allowance.

team data DWP also provided multiple synthetic datasets constructed by two external commercial teams (denoted team B and team C), who had been previously identified by the DWP as potential candidate institutions to synthesise DWP data. Each team provided two synthetic datasets

derived from Dataset A, and three synthetic datasets derived from Dataset B. These have the same columns as the original datasets.

DWP deliberated introduced errors into Dataset A, with the purpose of testing the robustness of the teams' data generation processes. These errors include:

- 10 000 duplicate records
- Mismatch between certain individuals' DOB and age

2.2 Data quality issues

We identified a number of issues that may affect the outcome of this piece of research:

- **Post Code**/**District**: Through discussions with the data provider, we determined that the post codes within the dataset were generated from a real stem value, but were not necessarily genuine postcodes, corresponding to real habitations. It was therefore not possible to validate the relationship between `District` as a local authority or regional designation and the post code.
- **Generated data columns**: Several fields within the dataset, including the post code and name columns, were generated using the Python Faker package [26]. We therefore would not expect to find realistic results when basing the results of some tests on these attributes. Consider for instance `Name`, which may encode useful information about socio-economic or cultural background which will not be accounted for in a synthetic set that is composed of semi-randomised results.
- **Occupations**: Occupations are produced by Faker, but are more typically a free-text field in the real DWP data. To group these by occupation, we performed string matches to match to the ILO occupational groupings[1].

[1] `https://www.ilo.org/public/english/bureau/stat/isco/isco08/`

- **NaN values**: Dataset B contains some NaN[2] values, mostly in the optional columns. e.g. `Gender 2`. For data generation, either dropping any row containing a NaN, or imputing the missing data might be considered. Imputation might affect the quality of synthetic data.

3 Data exploration and visualisation

The exploratory data analysis included building feature histograms and computing correlations between features. We also considered the number of unique values present for each feature.

The marginal distributions for the real and synthetic versions of both datasets were found to be similar in many cases (by observation of the distribution plots), presented in Figures 1-4. The largest difference between real and synthetic data is observed for some categorical variables, and for team C in general (as shown in Figure 1). Largely the same observations can be made between Dataset A and Dataset B for these purposes. The full analysis, including a large number of additional plots, is presented in the notebook `eda/EDA.ipynb` included in the code artefact, and reproduces several figures found in the team reports on the data.

Figure 2 shows a surprising anomaly in the distribution of dates of birth in the original dataset. These dates are clustered into tight groups at five-year intervals, leading to a distribution with many separate peaks. The same anomaly seems to have been well captured by both synthetic datasets (as is desirable). Overall, the range of ages represented is as would be expected—the youngest are 16 years old, and the frequently mostly drops to zero after retirement age. There are a number of outlying individuals older than this, however, including some much older.

Figure 3 shows the distribution of the most frequent Surnames found in the dataset. The most frequent names in the original dataset and team B's synthetic dataset are similar (with the exception of Martinez/Thomas), but

[2]Not A Number, a Python language convention for a missing numeric value

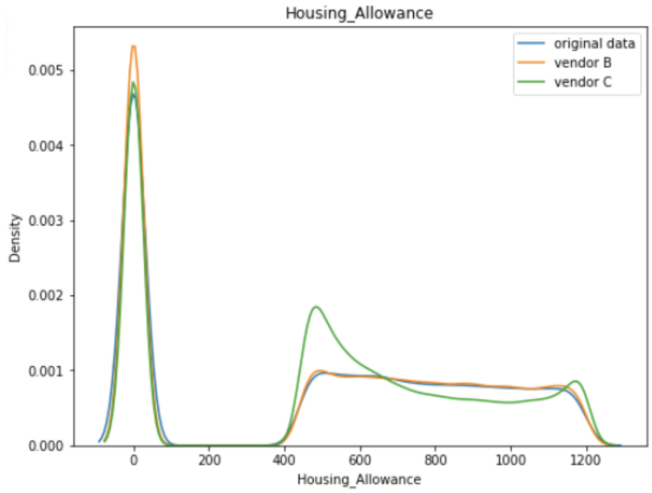

Figure 1: Distribution of `Housing_Allowance` for the original and synthetic data from both teams, for dataset A. Note the somewhat unrealistic distribution for team C. The apparent density present for negative values in the distribution of `Housing_Allowance` is a plotting artefact.

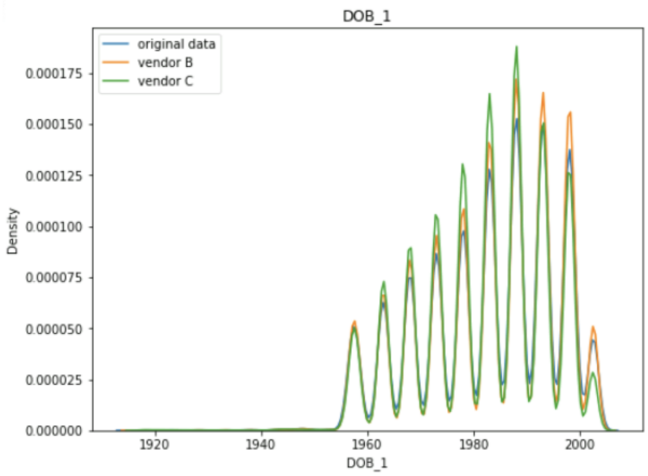

Figure 2: Distribution of `DOB_1` (date of birth) for the original and synthetic data from both teams, for Dataset A. Note the unusual and unrealistic multimodal distribution, with peaks every five years (but not completely discretized). This seems to be a genuine feature of the original underlying data (and not a plotting artefact), and is perhaps caused by the mechanism used for the initial synthesis. This anomaly is reproduced by both synthetic datasets.

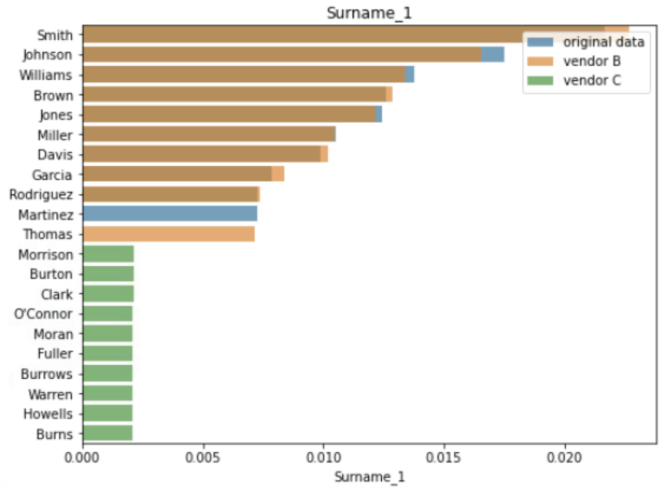

Figure 3: Distribution of the `Surname_1` field of the original and synthetic data, for Dataset A, with the ten most frequent names shown for each. The horizontal axis shows the proportion of rows with a particular surname. The most frequent names in the original dataset and team B's synthetic dataset are similar (with the exception of Martinez/Thomas), but those for team C are completely disjoint from the original.

those for team C are completely disjoint from the original, and follow a distribution close to uniform, which is rather different from the original.

These figures used the columns `Surname_1`, `DOB_1` and so on, rather than `Surname_2`, `DOB_2`. The latter represent a second adult (perhaps a partner) for claims in multi-adult households. These fields were null (NaN) in for $>$85% of the records.

A correlation analysis of numeric columns carried out on Dataset B, as shown in Figure 5, shows a low level of linkage between most attributes, while exposing the key insight that the number of children in a household is unsurprisingly correlated both with the amount of child-related allowance and the total amount awarded.

4 Generation

Several generation methods were considered, including PATE-GAN [27], Rob-GAN [33], RDP-GAN [34], or DP-CGAN [55], since these models in-

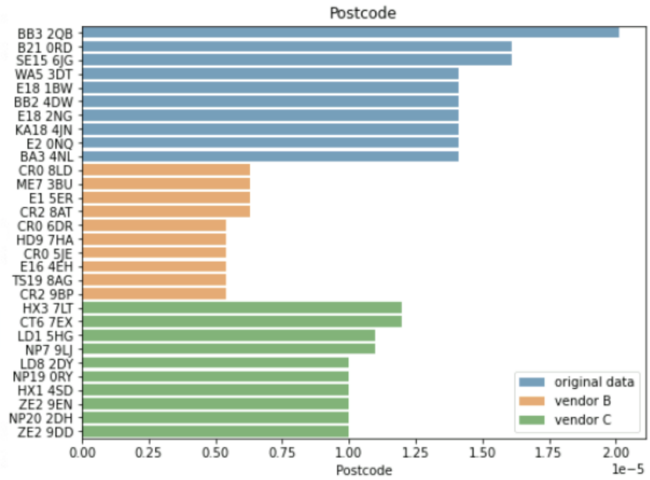

Figure 4: Distribution of the `Postcode` field of the original and synthetic data, for Dataset A, with the ten most frequent postcodes shown for each. The horizontal axis shows the proportion of rows with a particular postcode. Notice that the most frequent postcodes for each team are disjoint.

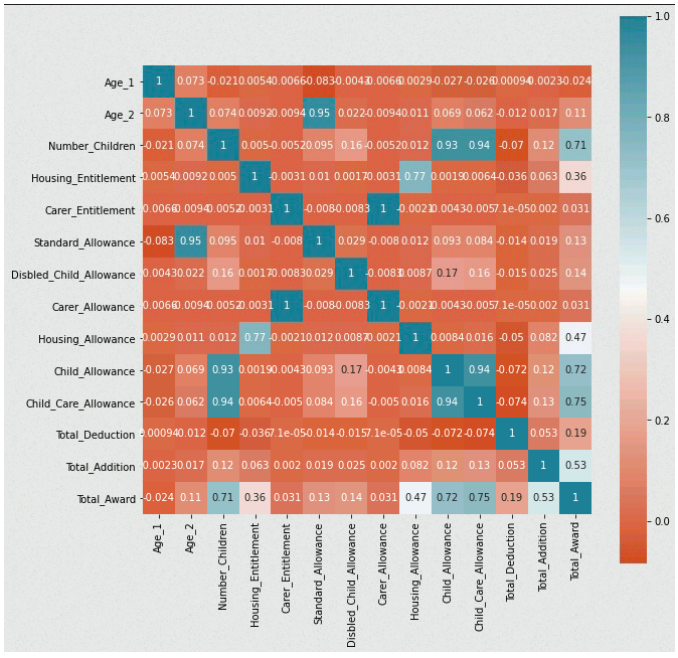

Figure 5: Heatmap showing the correlation between the numerical columns of Dataset B. Most pairs of columns are completely uncorrelated, but several are closely correlated. Many of these are unsurprising (e.g. `Number_Children` with `Child_Allowance` and with `Child_Care_Allowance`, both types of child benefit). See the text for further discussion.

clude various approaches towards implementing privacy mechanics within the generative mechanism, including achieving a measurable privacy guarantee via the implementation of a differential privacy system.

However, these methods proved too time-consuming to implement with the resources available to us. We refer to some of these methods in the further work section as possible avenues for continuing research.

4.1 Gaussian Copula method

The Gaussian copula provides a particularly simple method to synthesize data. It is implemented by SDV [42].

Copulae capture the dependence between random variables. An important theorem, attributed to Sklar, is that the joint cumulative distribution function of a multivariate distribution may be represented in terms of the cdfs of its marginals, and a function linking them—this latter function is the copula. The marginal cdfs are functions of one variable alone, containing no information about the dependence of the variables on one another, and the copula

After an overview of copulae, we describe how we used them to produce synthetic data: In short, we determine the marginal distributions from the data, and make the assumption of a Gaussian copula, with covariance also fit to the data.

Under some weak conditions, a distribution whose cdf is F can be expressed as

$$F(x_1, \dots, x_K) = C(F_1(x_1), \dots, F_k(x_K)) \tag{1}$$

where F_k is the marginal distribution of X_k, and where $C : [0,1]^K \to [0,1]$ is the copula.

The Gaussian copula is given by

$$C_{\mathcal{N}}(u_1, \dots, u_K; \mu, \Sigma) = \Phi_K \left(\Phi^{-1}(u_1), \dots, \Phi^{-1}(u_K); \mu, \Sigma \right), \tag{2}$$

where Φ is the (univariate) standard normal cumulative distribution function, and Φ_K is the multivariate cumulative distribution in K variables, with the specified mean μ and covariance Σ.

If the marginals X_k have standard normal distributions, it can be seen that equation 1 recovers a multivariate normal distribution, since

$$F(x_1, \dots, x_K) = C_{\mathcal{N}}(F_1(x_1), \dots, F_k(x_K); \mu, \Sigma) \tag{3}$$
$$= C_{\mathcal{N}}(\Phi(x_1), \dots, \Phi(x_K); \mu, \Sigma) \tag{4}$$
$$= \Phi_K(x_1, \dots, x_K; \mu, \Sigma). \tag{5}$$

In general, a particular joint distribution may not have Gaussian marginals, and yet may still have a Gaussian copula.

Suppose now that we want to produce synthetic data based on N observations of K features. Let $(X_1, ..., X_K)$ be random variables corresponding to a particular observation, and suppose that the distribution underlying the observed data is given by

$$F(x_1, \dots, x_K) = C(F_1(x_1), \dots, F_k(x_K)) \tag{6}$$

where C is the copula, and F_k is the cdf of the marginal distribution of the kth feature, and these are all unknown.

For each k, we determine an approximate marginal, $\widetilde{F}_k$, from the observations $(x_k^i)_{i=1}^N$, and approximate the joint distribution as

$$\widetilde{F}(x_1, \dots, x_K) = C_{\mathcal{N}}(\widetilde{F}_1(x_1), \dots, \widetilde{F}_k(x_K); \widetilde{\mu}, \widetilde{\Sigma}), \tag{7}$$

where $\widetilde{\mu}$ and $\widetilde{\Sigma}$ are estimates of the mean and covariance matrix.

Synthesizing new data is now a matter of sampling from the distribution with joint cdf $\widetilde{F}$.

4.1.1 Applying the Gaussian copula to Dataset B

As noted earlier in the report, Dataset B is monthly time-series data. We can see that for a given UC_Ref, most of the data except the last two columns for the Total_Addition and Total_Deduction remain constant over time, unless Family_Type or Number_Children change. Therefore, we will omit the generation for those two exceptional columns and also the Total_Award column.

There are two main difficulties with directly applying a Gaussian copula to this dataset. The first difficulty is the appearance of 'NaN' values, and the second is the change in award amounts for a household when particular events occur (such as having children or getting married).

To overcome these difficulties and apply the Gaussian copula method, we split the data set into 10 groups according to their Family_Type and Number_Children. We then generate data corresponding to each household (with a given UC_Ref, taking this to be the row of the first month where this appears in our dataset).

We describe the steps of our synthesis process below.

1: Summarise the original dataset

We first summarise the procedure used to synthesise data corresponding to the month that the household appears in the data set.

1. Remove the Total_Addition, Total_Deduction and Total_Award columns from the original data set.
2. Remove columns that are unlikely to contribute information to further inference. This includes National Insurance numbers, Names, Postcode (although we keep District) and ROW_ID.
3. Remove the columns that can be computed deterministically from other columns. In our generation, we remove the following columns: Count_Date, Age, Employment_Status, Child_Disabled (which

can be inferred from `Disabled_Child_Allowance`, when this field is non-zero).

4. Construct the new dataset with 40 000 rows, including two rows corresponding to the first and the last months that a given `UC_Ref` appears.
5. For each `UC_Ref`, generate extra columns: `Family_Type_Change` and `Number_Children_Change` to indicate the month that those events occur.
6. Generate extra columns to indicate the last month that each `UC_Ref` appears in the data set.
7. Split the data set (with 40 000 rows) into ten groups, corresponding to family type and number of children.
8. For each group, fit a Gaussian copula, and use it to generate the dataset for each group. The total count of each group corresponds to the considered dataset, restricted to the month that it first appears. This results in 20 000 rows being synthesized.

Note: We can see that all of the values in each column are either discrete number or categorical data. We thus allocate an intervals of a continuous variate to each category, and with the width of each interval corresponding to the proportion of each. We then randomly assign a continuous value in the interval to each data point—in our simulation, we use a truncated Gaussian distribution. For date of birth data, we convert these to integers. We then take this transformed data as the raw input to the Gaussian copula and construct the marginal cumulative density function by using their empirical distribution.

2: Map family type updates

Suppose that (X, Y) has a multivariate centred Normal distribution, and that $Var(X) = \Sigma_X$, $Var(Y) = \Sigma_Y$ and $Cov(X, Y) = \Sigma_{XY}$. In this case,

$$X|Y \sim \mathcal{N}\left(\Sigma_{XY}\Sigma_Y^{-1}Y,\ \Sigma_X - \Sigma_{XY}\Sigma_Y^{-1}\Sigma_{XY}^{\top}\right). \tag{8}$$

That is, $X|Y$ is also normally distributed, with the given mean and variance. We use this fact to generate new observations.

For the 20 000 synthesized data where a particular event occurs (either they have have a new child or get married), update the `Family_Type` or `Number_Children` as appropriate. After the update, transform the column to be normally distributed again. If the family type changes, regenerate `Standard_Allowance`, `DOB_2`, `Occupation_2`. If `Number_Children` changes, it can only have increased by one: Randomly assign a date within the month to the next available `DOB` column. In this case, also recompute `Child_Allowance` and `Child_Care_Allowance`.

3: Complete generation

1. We construct the complete dataset of 240 000 rows, including every month included in the synthesis. We also take into account the events that were identified by the contents of the additional columns `Family_Type_Change` and `Number_Children_Change`.
2. We drop some rows according to the time that the household participates in UC.
3. This leaves us with data corresponding to each `UC_Ref`. Each `UC_Ref` has records from January to the month that they leave UC, which includes some `UC_Refs` that have fewer than twelve months of records (the smallest in the original dataset had seven months). In our simulation, we see roughly 50–60 rows with missing data caused by an individual leaving UC in this way. To match the original data, we randomly select particular values of `UC_Ref` that have twelve months of records, and drop the initial months of data, such that there are exactly 100 values of `UC_Ref` that have fewer than twelve months of data, as in the original dataset.
4. The columns that can be directly determined from the generated data columns are completed.

5. The name and NI fields can be randomly generate (e.g. with Faker), and assigned to each `UC_Ref`.

6. `Total_Addition` and `Total_Deduction` were generated using a linear regression model[3]. We observe that for the majority of records, `Total_Addition` and `Total_Deduction` are zero—we only apply the regression to the rows where `Total_Addition` and `Total_Deduction` are non-zero. To obtain `Total_Addition` and `Total_Deduction` for our generated data, we first choose rows at random that will have a non-zero value for these columns, and then apply the regression model. All other rows will have a value of zero for these fields.

4.1.2 Comment on the code and generated data

Near the end of the challenge, and after generating data in this way, we identified some issues with the code that we used to generate our data, which may have an impact on the quality of the generated data.

1. When transforming categorical data to a continuous normal distribution (as described in the note to the step 'Determine when an individual household first appears in the original data set'), immediately transforming the output back results in a change in category for a few percent of the rows, when they should be identical. Given the time constraint, we have chosen to ignore this error for the course of the challenge, which will affect the similarity of the distribution of categorical features slightly. It is likely caused by a bug in the function used to transform categorical data to continuous data with a normal distribution.

2. In our generator, children born during the time period of interest (that is, when a row has `Number_Children_Change` of 1) never have

[3] These columns (`Total_Addition` and `Total_Deduction`) could also be generated using a Gaussian copula applied to the dataset, conditional on the synthetic data generated using the above procedure. However, given the time constraint, we opted for a simple regression model

Child_Disabled_* set (that is, they are never disabled). This will have a clear effect on the utility of the data, compared to the original dataset, for related research questions. We also restrict the date of birth of the new child to be from the 1st to the 14th of the month. This is to make sure that the date of birth is consistent with the Count_Date column.

3. In the original dataset, the surnames of members of a household are not independent, and have a high probability of being the same. This property is not reflected in our generated data, where each surname is generated independently.

4. As noted, we used a simple regression model to predict Total_Deduction and Total_Addition. Useful further work would be to consider other models, and whether more sophisticated models confer an advantage to the utility of the data.

4.2 CT-GAN

Conditional Tabular Generative Adversarial Networks, or CT-GANs, is a GAN-based approach to data synthesis, developed to handle a mixture of continuous and discrete features [62].

For generating synthetic data using CTGAN, we again used the framework developed by SDV. The Synthetic Data Vault (SDV) enables end users to easily generate Synthetic Data for different data modalities, including single table, multi-table and time series data. They also have an implementation of CTGAN for tabular data.

1: Baseline generation

We generated synthetic data using a baseline model of SDV's CTGAN with default values of its hyper-parameters. These are:

- Epochs and batch size: ts default values are 300 and 500 respectively, and batch size needs to always be a value which is multiple of 10.

- Log frequency: Whether to use log frequency of categorical levels in conditional sampling. It defaults to True.
- Embedding dim (int): Size of the random sample passed to the Generator. Defaults to 128.
- Generator dim (tuple or list of int): Size of the output samples for each one of the Residuals. Defaults to (256, 256).
- Discriminator dim (tuple or list of int): Size of the output samples for each one of the Discriminator Layers. Defaults to (256, 256).
- Generator lr (float): Learning rate for the generator. Defaults to 2×10^{-4}.
- Generator decay (float): Generator weight decay for the Adam Optimiser. Defaults to 1e-6.
- Discriminator lr (float): Learning rate for the discriminator. Defaults to 2×10^{-4}.
- Discriminator decay (float): Discriminator weight decay for the Adam Optimiser. Defaults to 1×10^{-6}.
- Discriminator steps (int): Number of discriminator updates to do for each generator update. Default used is 1 to match the original CTGAN implementation. *source: SDV*

The synthetic data generated from the baseline model resulted with numerical columns were not highly accurate but similar to the original data.

2: Hyper-parameter tuning We tuned the hyperparameters of the CTGAN model by reducing them 10 times to the default values, initially for generating only numeric columns. The similarity between real and synthetic data was about 0.77

Here we present loss function of different model setups. As you can see we we got good improvement in process of model training. We sure that

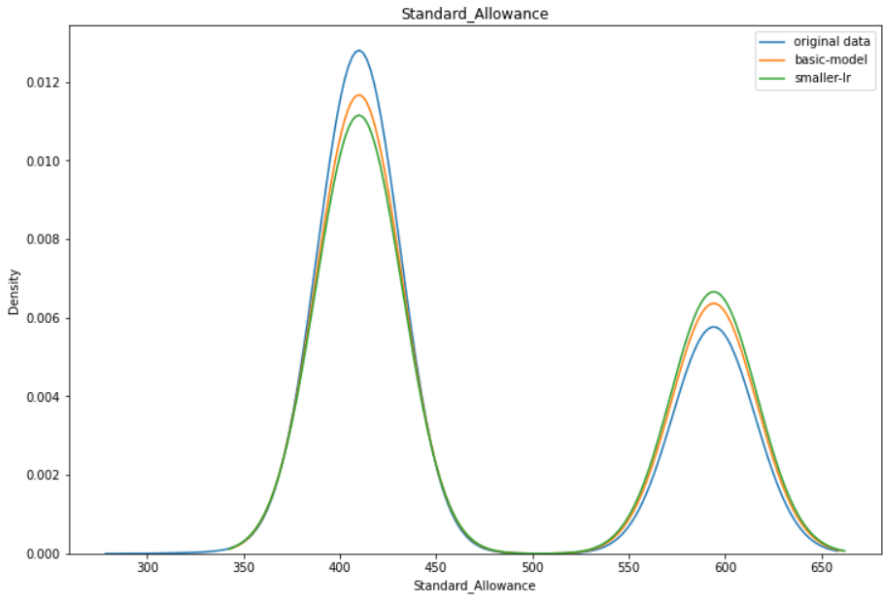

Figure 6: Distribution of `Standard_Allowance` in the original (blue line) and two synthetic data sets produced by CTGAN: The baseline model (orange line), and a tuned model, after adjusting the hyperparameters (green line).

model's hyperparameters can be improved significantly, but it needs much more times and experiments.

However, generating text or categorical columns using this model does not produce results close to the original data. We suspect that the use of Faker in SDV's implementation of CTGAN could be the reason for the dataset's low similarity.

4.3 Text Generation

Ideally, we would jointly generate the textual information in our dataset alongside the numerical and categorical fields to ensure consistency between the source dataset and the generated data. Our initial research indicated the best way to achieve this would be to train a model to learn the joint distribution of the numerical fields from the source set and predict the contents of the textual fields, thereby learning both the plausible content of the field as well as the format constraints.

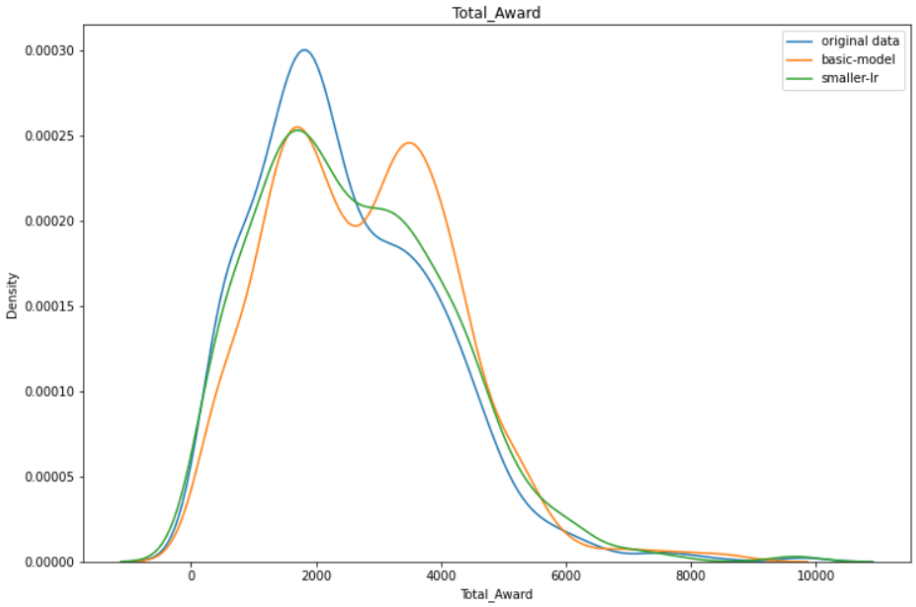

Figure 7: Distribution of `Total_Award` in the original (blue line) and two synthetic data sets produced by CTGAN: The baseline model (orange line), and a tuned model, after adjusting the hyper-parameters (green line).

We determined that a classification system using the Transformer architecture [57] would be eminently capable of generating the range of outputs we require, using a pre-computed mapping between the output representation and a corresponding high-dimension vector. This would have the benefit of allowing us to compute an intermediate representation—what one might call a row embedding—for any synthetic set in a consistent format, which we could use to predict the appropriate text field values.

This proved too time-consuming to realistically effect however, and plans to implement the system were shelved in favour of generating field content through a semi-randomised process using the Python package Faker. We would intuitively expect this system to generate content that did not appear in the original dataset, leading to a less rigorous match to the underlying distribution of values. It is our belief that a fine-tuned generation process would produce superior results in terms of utility.

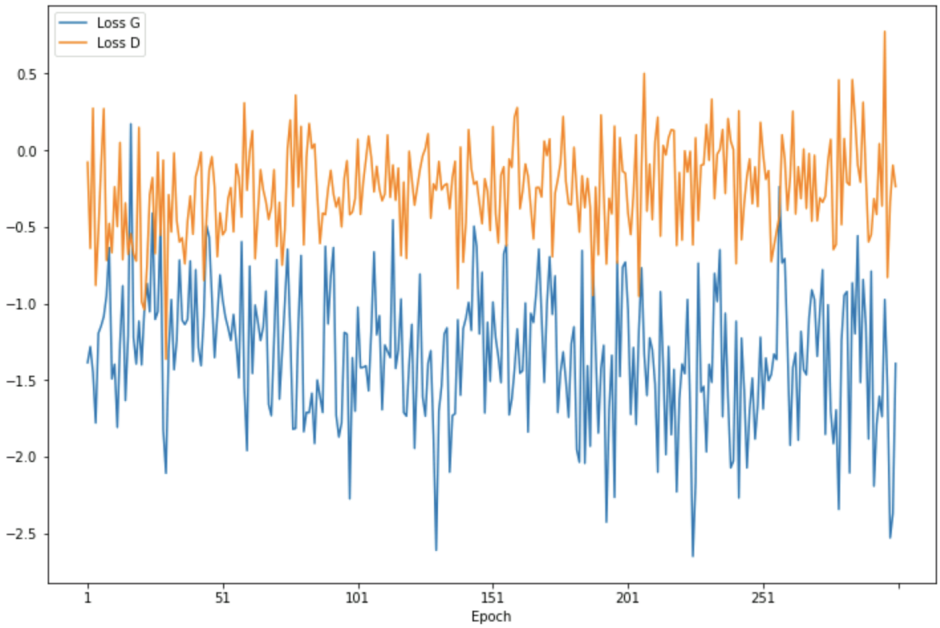

Figure 8: Loss during training of the Generator (Loss G) and Discriminator (Loss D), for the baseline model, with parameters described in Step 1 of the CTGAN section of the text.

5 Evaluation

This section will deal with the question of the quality of synthetic data in comparison to the original dataset. We will examine in turn:

1. The potential level of privacy embodied in the generated set,
2. The effective utility when using the synthetic data on potential downstream applications,
3. Potential effects on the robustness of generated sets,
4. The addition of new sources of bias that may affect the fairness of generated sets.

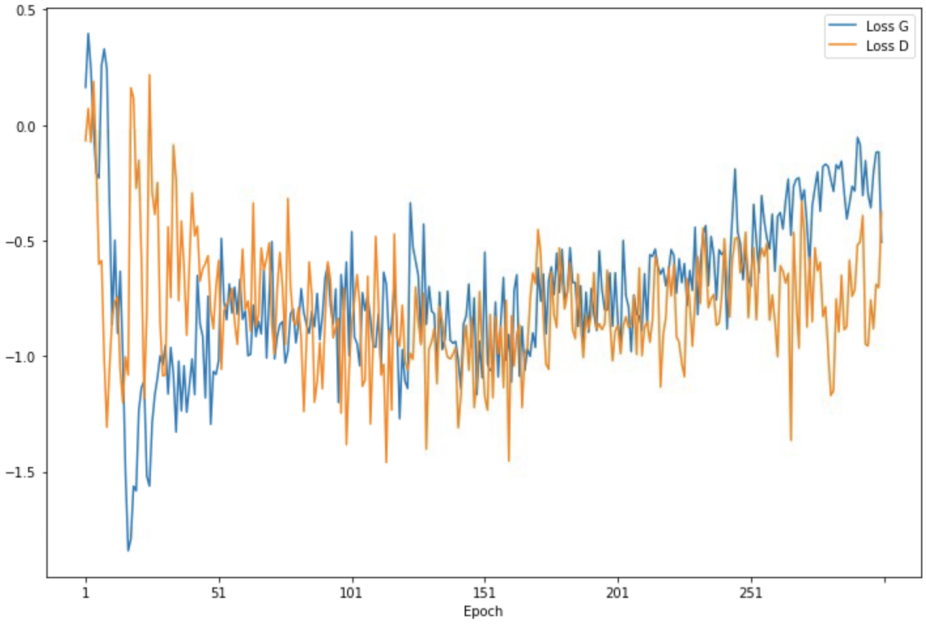

Figure 9: Loss during training of the Generator (Loss G) and Discriminator (Loss D), for model with smaller learning rate described in Step 2 of the CTGAN section of the text.

5.1 Privacy

There are a number of ways to evaluate the privacy of a dataset. Wagner and Eckhoff [59] present an authoritative taxonomy based on the output measure desired, which we reproduce as Figure 10.

Given the scope of the challenge, we were unable to investigate all of the potential measures included here, and so we chose to focus on some fundamental measures that we believe give a good measure of the potential risk of data release, specifically for synthetic data in comparison to the original set. We encourage further research in the application of additional metrics to the problem, and have referred to several potential approaches in the Further Work section of this report.

At the most fundamental level, we wish for the synthetic data to be similar to the real data: Not only would this make it harder for attackers to determine

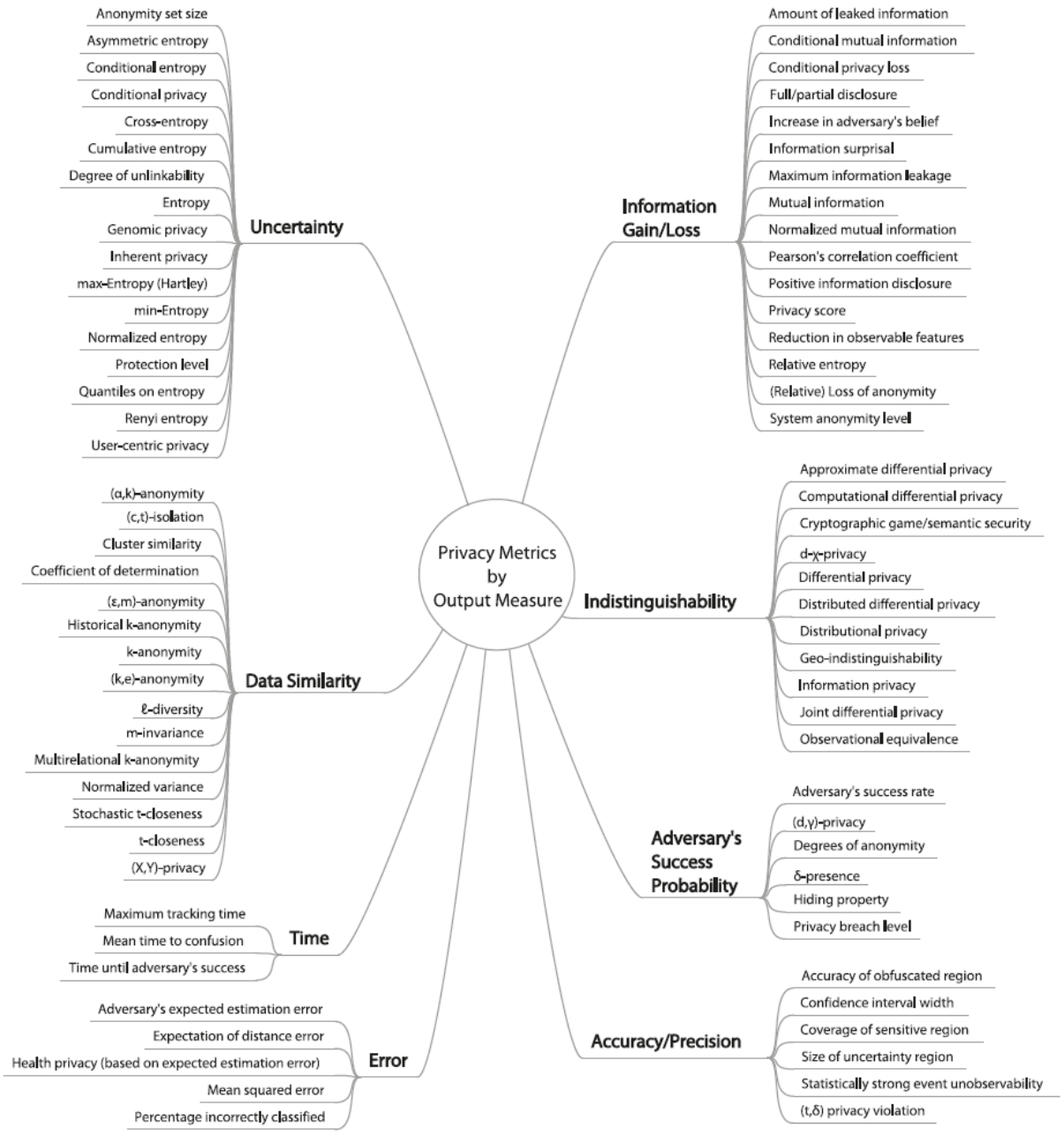

Figure 10: Privacy metric taxonomy in Wagner and Eckhoff (2018).

whether obtained data is real or fake, but it would also improve the quality of any analyses done with the synthetic data [47, 15].

To this end, we conducted tests based on established distributional metrics to quantify the similarity of the original real to the generated synthetic data. These metrics are statistical in nature and, as the goal is to identify the possible differences between samples from the real and synthetic data, are appropriately termed *two-sample tests*. In short, these metrics calculate different functions (formally, *statistics*) of the data distributions and compare them to particular reference distributions. From there, a p-value is calculated. Generally, the lower the p-value, the easier it is to tell the real and synthetic data apart.

In the DWP datasets, there were multiple data fields, some categorical, some numerical, others in a gray zone (e.g. First Name, NI). As a first pass, we restricted ourselves to single columns of categorical or numerical data, such as **District** (categorical) or **Age** (numerical). The two-sample test we applied differed based on the type of data under consideration. For categorical data, we applied *chi-squared* tests to compare the relative frequencies of categories (Table 1). For numerical data, we applied a suite of tests (Kolmogorov-Smirnov [KS] test, Student's t-test, Levene's test: Tables 2, 3) that each tested different aspects of the numerical distributions.

Although these metrics can be successfully applied to the data, they have one glaring weakness: they are generally univariate in nature and cannot be easily extended to multivariate fields, especially when dealing with a combination of numerical, categorical, and other data types. Therefore, we turned to some of the more modern literature, from which we identified the Kernel MMD (maximum mean discrepancy) two-sample test [20] as a promising approach. The primary advantage of this more modern approach is that it effectively subsumes most earlier tests. It does this by projecting the data onto a high-dimensional space (called an *RKHS*, or reproducing kernel Hilbert space) and utilising what they call a *witness function* to identify differences of any arbitrary nature.

The downside of such a method is that it requires the user to choose a *kernel*, which is not a straightforward task and is in practice often done by training a deep neural network. Choosing an appropriate kernel should allow the model to analyse any type of data, whether categorical or numerical in nature. Unfortunately, we were not able to implement this successfully over the course of the DSG, but we recommend this task for future work.

Stepping back a bit, however, it is important to note that although we have investigated a number of metrics for **distributional similarity** between real and synthetic datasets, this says nothing about the privacy of the synthetic datasets themselves. In fact, simply reproducing the original data would produce "synthetic" data from the same distribution, yet that data would obviously not be private. In the following sections, we tackle approaches to more private data.

	District	Type	Gender1	Status1	Gender2	Status2
Dataset A						
team B	2772.17	0.43	N/A	206.49	N/A	13.90
team C	75725.3	307618	18838.5	246099	N/A	157578
Dataset B						
team B	N/A	12110.2	258.97	39.25	9867.14	10181.6
team C	10719.6	398.29	34.55	1310.85	298.4	333.81
CTGAN	N/A	1056.48	6.48051	2.10716	1235.31	1472.72
GC	251.911	477.675	19.7176	36.345	27090.7	5356.56

Table 1: Similarity test with categorical features (χ^2). Type = Family Type, Status = Employment Status.

	Age 1	Age 2	No. Children	St. Allowance
Dataset A				
team B	ks: 0.00837	ks: 0.01221	ks: 0.00918	ks: 0.00489
	lv: 32.23	lv: 101.66	lv: 496.56	lv: 58.64
team C	ks: 0.04075	ks: 0.1046	ks: 0.04151	ks: 0.08594
	lv: 7317.38	lv: 63897.88	lv: 1172.90	lv: 87751.33
Dataset B				
team B	ks: 0.03028	ks: 0.08809	ks: 0.04472	ks: 0.11105
	lv: 38.73	lv: 132.13	lv: 983.70	lv: 8074.56
team C	ks: 0.04075	ks: 0.1046	ks: 0.04151	ks: 0.08594
	lv: 7317.38	lv: 63897.88	lv: 1172.90	lv: 87751.33
CTGAN	ks: 0.04983	ks: 0.5086	ks: 0.05559	ks: 0.49918
	lv: 13.65	lv: 11511.20	lv: 37.01	lv: 1707.77
GC	ks: 0.00389	ks: 0.16188	ks: 0.01896	ks: 0.02089
	lv: 11.39	lv: 1848.66	lv: 56.46	lv: 247.75

Table 2: Similarity test with numerical features, part 1. KS = Kolmogorov-Smirnoff test LV = Levene's test.

	Allowance Type			
	Carer	Housing	Child	Total
Dataset A				
Team B	ks: 0.00281	ks: 0.01134	ks: 0.00918	ks: 0.02774
	lv: 277.95	lv: 142.97	lv: 594.82	lv: 0.04
Team C	ks: 0.05718	ks: 0.05234	ks: 0.01376	ks: 0.04289
	lv: 88492.68	lv: 61216.24	lv: 18.92	lv: 15668.14
Dataset B				
Team B	ks: 0.00239	ks: 0.12753	ks: 0.02733	ks: 0.05823
	lv: 7.59	lv: 669.10	lv: 368.33	lv: 244.70
Team C	ks: 0.00264	ks: 0.07896	ks: 0.02261	ks: 0.05570
	lv: 9.27	lv: 2054.84	lv: 1.29	lv: 144.23
CTGAN	ks: 0.5405	ks: 0.13412	ks: 0.2269	ks: 0.10066
	lv: 0.01	lv: 15.20	lv: 201.60	lv: 133.30
GC	ks: 0.00482	ks: 0.00424	ks: 0.02102	ks: 0.01624
	lv: 29.80	lv: 5.65	lv: 249.25	lv: 77.20

Table 3: Similarity test with numerical features, part 2. KS = Kolmogorov-Smirnoff test LV = Levene's test.

5.1.1 Private Attribute Prediction

One of the primary risks in publishing a dataset is that it may expose the private information of participants to unwanted scrutiny; it is for this reason that most published data is usually anonymised in some fashion [25]. However, even removing identifiers such as Social Security numbers or names does not guarantee privacy. Re-identification attacks can reliably uncover this hidden information by linking the published set with external datasets and background knowledge [64, 30, 3, 52]. Narayanan and Shmatikov (2008) [37] proposed an effective methodology for achieving this de-anonymisation of personal data under which an attacker with moderate assumptions about a sparse pseudo-anonymous dataset can make correlations with existing public data to uniquely identify individuals, which they proved in spectacular fashion by successfully identifying 99% of subscribers included in the Netflix Prize dataset.

Such an attack can take the form of inferring the value of a private attribute from publicly available data that has either been anonymised or obfuscated in some form by the application of machine learning techniques [53, 41, 36, 18]. In the case of synthetic data release, we consider the possibility that an attacker may be capable of training a model with the released data to predict the value of a private demographic attribute, such as gender, age, racial background or employment status, that can then be used with a secondary dataset gathered from public sources to infer these protected characteristics of real individuals.

We obtain empirical estimates of this risk factor by training a set of classifiers (including Naive Bayes, Random Forest, Multi-Layer Perceptron, and Logistic Regression) to predict the gender of a record using other demographic data from the synthetic dataset. Then, we test the predictions of these classifiers against the original dataset, and score the performance of the models. If the synthetic dataset does not encode significant information about the real individuals, then we would expect the performance outcomes to resemble simple chance (50%).

We use as a success metric the F1 score (defined as the harmonic mean of the recall[4] and precision[5]). Note that precision, and hence F1, can be indeterminate if no predictions are made for a particular class, and in this case the score for all classes is set to zero. Results can be seen in Table 4.

	NB	RF	MLP	LR
Dataset A				
team B	0.53008	0.52002	0.46797	0.43854
team C	0.51378	0.44137	0.50853	0.44151
Dataset B				
team B	0.48382	0.40411	0.27028	0.27144
team C	0.51945	0.48407	0.47928	0.43533
CTGAN	0.49703	0.40516	0.46028	0.40402
GC	0.43136	0.27408	0.40402	0.27046

Table 4: Private attribute prediction. NB = Naive Bayes, RF = Random Forest, MLP = Perceptron, LR = Logistic Regression.

We can see from these results that the attack has been relatively unsuccessful across the board, achieving no more than 3% performance improvement above random chance in the best case. We note the anomalously low results provided by the MLP and LR classifiers when applied to the data generated by team B derived from dataset B. This may be a misleading result, perhaps caused by a the classification algorithm failing to converge, a possibility that could be investigated by performing multiple cross-validation runs.

[4]Recall: The number of true positive predictions divided by the number of results that should have been predicted positive.

[5]Precision: The number of true positive predictions divided by the total number of positive predictions.

5.2 Utility

A key observation of previous research into privacy-preserving techniques is that there is an inherent trade-off between the level of personal risk reduction and the amount of utility that is preserved for tasks relying on the privatised dataset [44, 29, 17, 31]. We present a set of experiments here into a range of simulated tasks that a data consumer may wish to accomplish given a privatised synthetic dataset, providing some indicative results for how much additional error is introduced via the process.

It is important to note however that this task set should not be considered comprehensive, and indeed there may be particular use cases that exhibit markedly different dynamics in response to the privacy regimes under consideration. We recommend investigation of a wide range of possible tasks [22, 45] before any determination of data quality and robustness is made.

5.2.1 Metrics

The primary tests in this section involve regression towards a continuous variable, and hence we focus on metrics appropriate for that case. Those we have used are defined below.

Definition 5.1 (Mean Squared Error)**.** *Given examples $n \in N$ with real target value y and prediction $\hat{y}$*

$$MSE = \frac{1}{N}\sum_{n}^{N}(y_n - \hat{y}_n)^2.$$

Definition 5.2 (Mean Absolute Error)**.** *Given examples $n \in N$ with real target value y and prediction $\hat{y}$*

$$MAE = \frac{1}{N}\sum_{n}^{N}|(y_n - \hat{y}_n)|.$$

MAE scales linearly with changes in the underlying prediction error. Both MAE and MSE share the useful property that the units in which they are denominated are the same as the units of our target variable. In this sense, the scale of the error is straightforward to conceptualise.

Definition 5.3 (Coefficient of Determination, R^2)**.** *Given examples $n \in N$ with real target value y and prediction $\hat{y}$*

$$R^2 = 1 - \frac{\sum_n^N (y_n - \hat{y}_n)^2}{\sum_n^N (y_n - \bar{y}_n)^2}$$

where

$$\bar{y} = \sum_n^N y_n.$$

The R^2 metric quantifies the proportion of the variance in the target variable that has been explained by the independent variables used in the model. Values of R^2 can be negative, if the mean of the data provides a better fit to the outcomes than the predicted values in this particular test.

5.2.2 Tests of regression

These tests attempt to fit a regression model for a target variable to a selected group of categorical or numerical variables. Higher error indicates that the model is less accurate in predicting the target, and hence we would expect that the target is less strongly determined by those variables. In the case of synthetic data analysis, we would expect the original dataset to exhibit the lowest error and highest evidence of correlation, while sets that are generated from it will experience consequently higher error and lower correlation the less they resemble the original set. For this reason, we consider these measures good proxies for the general utility of the generated sets [22].

These experiments were conducted by extracting the `Total_Award` or `Allowance` columns from the dataset as the target variable which should be strongly determined by the demographic columns of the record. These

columns, including age, gender, job, employment status, number of children, and district were processed into categorical or numerical format as appropriate, and split into training (60%) and test sets (40%). The training set was used to fit a series of regression models, which then generated predictions for our unseen test set.

We developed models using Linear Regression, Random Forest Regression, Support Vector Regression, and Dense Neural Networks [13] for our test battery. Error metrics for each model can be seen in Tables 5 and 6. Results for dataset A can be found in Figure 11 and dataset B in Figure 12.

	LR			RF		
	MSE	MAE	R^2	MSE	MAE	R^2
Dataset A	13940	534.01	0.08	13336	549.56	0.12
Team B	**13606**	**530.58**	**0.08**	**13660**	**561.40**	**0.07**
Team C	18660	674.93	0.05	21247	760.66	-0.08
Dataset B	94070	714.72	0.52	60637	444.43	0.69
Team B	94800	**712.22**	0.47	75497	566.34	0.58
Team C	**93671**	750.61	**0.51**	**52088**	**450.96**	**0.73**
CTGAN	1400643	1011.6	-0.002	1501293	1027.35	-0.07
GC	817335	699.123	0.55487	502017	465.81	0.7266

Table 5: Regression results for Linear Regression (LR) and Random Forest (RF)

While both team B and C provide relatively good results, it should be noted that team B provides significantly better results in our synthetic utility test on Dataset A. Results with Dataset B are more mixed, perhaps indicating that team B's generation process scales in performance along with the size of the input dataset. This may also indicate an issue with team C's process in dealing with outliers in the data, as with the intentional errors added to Dataset A.

Our generative models provide some useful context here: The performance of the purely statistical Gaussian Copula model remains relatively stable across most tests in our battery, indicating a high level of reliability, while the

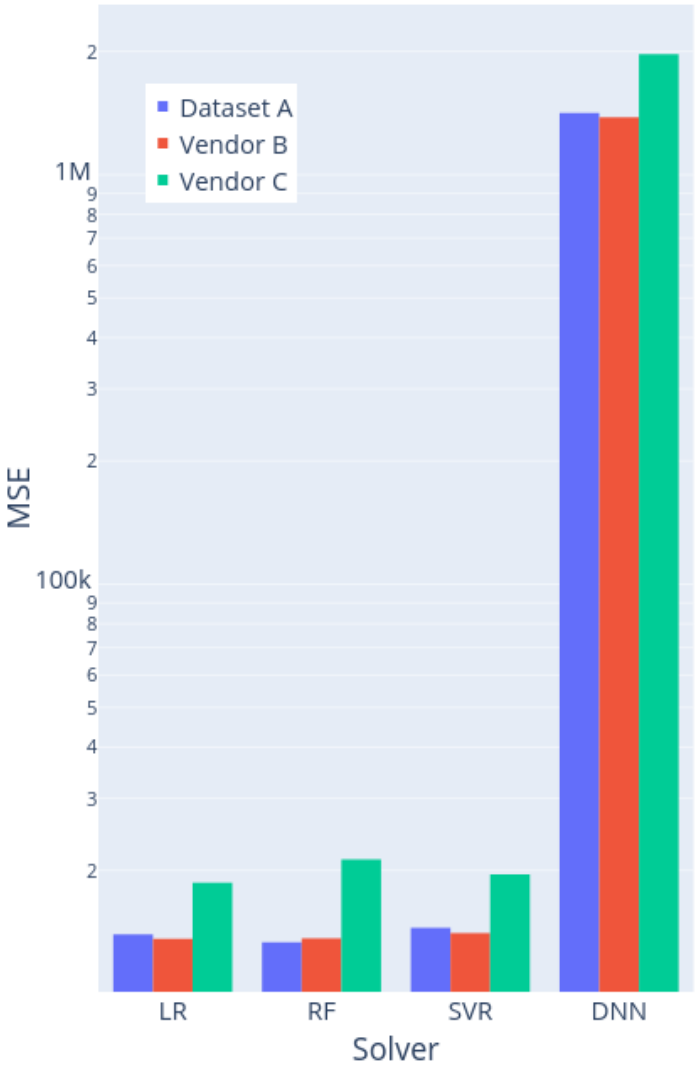

Figure 11: Results of various regressions carried out on Dataset A and the synthetic datasets. Shown are Linear Regression (LR), Random Forest (RF), Support Vector (SVR), and Neural Network (DNN) algorithms.

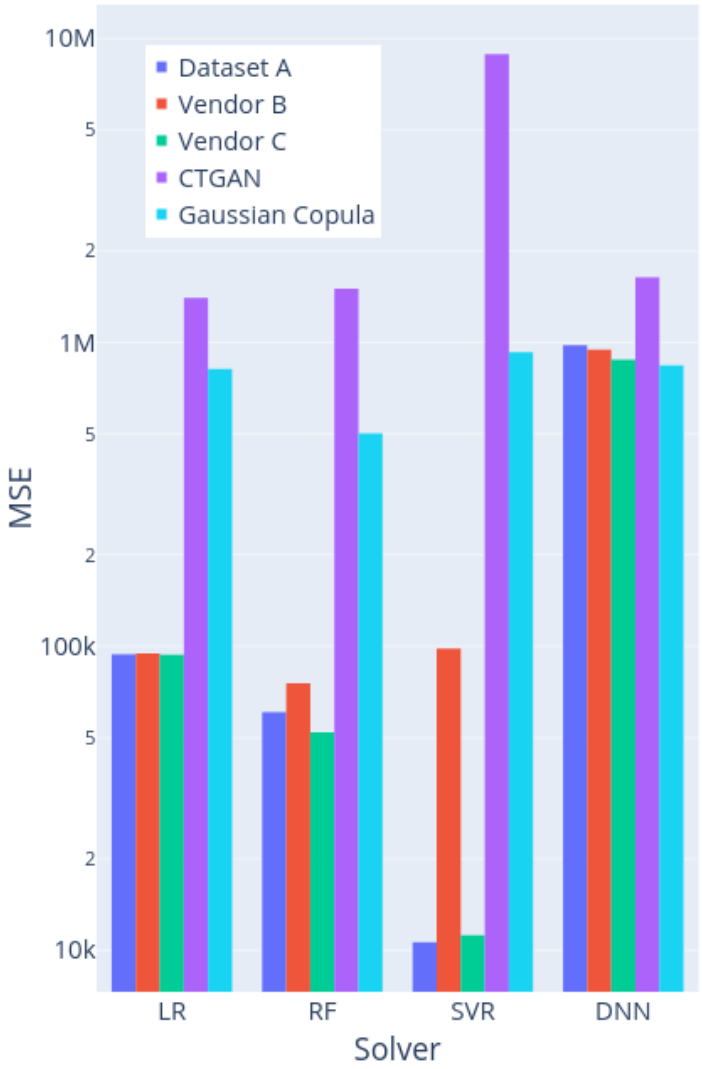

Figure 12: Results of various regressions carried out on Dataset B and the synthetic datasets. Shown are Linear Regression (LR), Random Forest (RF), Support Vector (SVR), and Neural Network (DNN) algorithms.

	SVR			DNN		
	MSE	MAE	R^2	MSE	MAE	R^2
Dataset A	14476	492.10	0.04	1413547	477.25	0.06421
team B	**14058**	**492.91**	**0.04**	**1380424**	**477.19**	**0.06153**
team C	19570	607.90	0.00	1970770	605.10	-0.00319
Dataset B	10601	764.08	0.45	979405	731.07	0.49516
team B	**98313**	**727.11**	**0.45**	**946318**	709.53	**0.47529**
team C	11179	809.57	0.42	876526	**726.99**	0.54265
CTGAN	8884567	2501.64	-5.35323	1635526	1062.21	-0.16954
GC	926765	743.125	0.49527	838386	708.446	0.54340

Table 6: Regression results for Support Vector Regression (SVR) and Dense Neural Network (DNN)

CTGAN process demonstrates clear signs of degraded utility. We speculate here that the reliability of results obtained from CTGAN would improve with additional fine-tuning, as well as training with a larger subject dataset.

5.2.3 Error in the marginal distribution

We illustrate error appearing in the categorical data in Table 7.

Table 7 illustrates the average MSE between the synthetic data and the original data. We can see from these results that the Gaussian copula method that we apply to Dataset B gives a similar error in the marginal distribution of `Total_Allowance` to the team synthetic data, with no particular method clearly outperforming the others.

	GC	B v1	B v2	B v3	C v1	C v2
District	3.57	5.21	6.46	5.31	16.71	20.21
Family_Type	14.20	72.51	20.22	6.79	13.25	65.68
Gender_1	3.12	11.75	6.68	1.16	4.26	31.83
Occupation_1	3.57	1.27	3.07	10.26	109.09	70.92
Occupation_2	17.85	58.27	14.12	4.36	19.01	78.69
Surname_1	3.28	4.47	4.67	3.84	17.06	17.21
Surname_2	10.20	55.39	14.16	4.63	11.17	62.37
Number_Children	1.68	1.60	1.68	1.68	1.66	1.64
Gender_2	12.31	68.45	17.56	6.00	11.71	64.01
First_name_1	3.56	4.56	4.66	4.01	31.55	32.36
First_name_2	90.75	51.40	14.53	4.80	13.96	62.91
Month	0.04	1.10	0.25	0.39	0.00	0.00
Child_Disabled_1	13.49	16.66	13.24	3.06	5.41	74.79
Child_Disabled_2	2.82	6.98	9.54	1.15	1.28	51.51
Child_Disabled_3	0.03	36.24	3.77	3.17	1.47	29.49
Child_Disabled_4	0.03	23.46	1.33	2.07	3.93	39.74
Housing_Entitlement	1.03	4.05	13.86	5.33	19.09	99.09
Carer_Entitlement	3.33	6.40	3.06	12.64	6.43	91.21
Count_Date	0.04	1.10	0.25	0.39	0.00	0.00
Employment_Status_1	4.10	4.24	4.06	13.77	24.50	41.86
Employment_Status_2	30.40	72.25	19.84	6.26	11.67	69.17
Age_1	15.48	15.34	15.38	15.37	15.24	12.86
Standard_Allowance	118.58	113.42	121.32	120.38	115.71	120.85
Disbled_Child_Allowance	29.21	25.22	26.34	27.85	29.98	47.25
Carer_Allowance	70.27	69.12	70.12	66.90	69.02	60.38
Housing_Allowance	868.82	846.13	858.59	869.50	897.95	919.94
Child_Allowance	318.64	319.39	318.22	318.26	317.19	318.18
Child_Care_Allowance	1089.87	1079.30	1092.28	1086.24	1141.11	1224.97
Total_Deduction	428.39	477.79	486.53	487.40	438.46	339.43
Total_Addition	715.23	753.98	748.89	767.88	728.28	574.26
Total_Award	1947.93	1945.44	1948.89	1976.40	1964.80	1925.44

Table 7: Error in categorical features for the synthetic datasets: GC=Gaussian cupola (our work), B=team B, C=team C, v1=version 1 of the provided synthetic data etc. The model with the smallest error is shaded for each feature.

5.2.4 Modelling

One potentially important application of synthetic data, is to public policy analysis. This could include research by staff within the department, but without access to the original data, by other UK government departments, or by the wider academic community. To evaluate the suitability of synthetic data for this purpose, we consider three modelling questions that are representative of some of the analysis of interest to DWP. This is an applied utility metric: Ideally, the synthetic data will replicate the distribution of the original data so that empirical policy analysis leads to the same conclusions, regardless of which dataset is used.

Question 1: Analysing the impact of family type on Universal Credit awards

We first investigated how awards of Universal Credit vary across family characteristics, including whether the household is a single or couple, and the number of children. Of particular interest is the impact over time, and so we applied two models to Dataset B, which is a longitudinal (household-month level) dataset over 12 months:

- Survival regression analysis
- Pooled OLS

Approach 1: Survival regression analysis We implemented Cox's proportional hazard model to study the uptake of Universal Credit over time, conditional on family structure. In this context, the hazard function $h(t)$ is the probability ("risk") of receiving a positive award of Child Credit:

$$h(t) = h_0(t) \exp(X\beta) \tag{9}$$

where X is the set of covariates, including family type (Single or Couple) and number of children.

The regression coefficients for the original data, and teams B and C are shown in Table 8 and survival conditional on family type and number of children are plotted in Figure 13. The regression output shows that the estimated impact of being a Couple on survival is underestimated by team B and overestimated by team C, by a large degree, relative to using the original dataset. The estimated impact of number of children is more similar across the original and synthetic datasets, with both teams B and C underestimating the impact of a marginal child. It can be seen from the plot that, independent of the team, the more children the family have, the less time passes before receiving Child Allowance, hence *ceteris paribus* increasing the total UC award of the household. We can also observe that single families tend to get the allowance faster. team B seems to overestimate the speed in which the families get the allowance, while team C seem to underestimate it.

	coef	exp(coef)	se(coef)	coef lower 95%	coef upper 95%	exp(coef) lower 95%	exp(coef) upper 95%	z	p	-log2(p)
Original dataset										
Family Type (Single = 0, Couple = 1)	-1.20	0.30	0.02	-1.23	-1.16	0.29	0.31	-63.45	<0.005	inf
Number of Children	0.80	2.22	0.01	0.78	0.81	2.19	2.25	120.93	<0.005	inf
team B synthetic data										
Family Type (Single = 0, Couple = 1)	-0.86	0.42	0.02	-0.90	-0.83	0.41	0.44	-51.92	<0.005	inf
Number of Children	0.67	1.96	0.01	0.66	0.68	1.94	1.98			
team C synthetic data										
Family Type (Single = 0, Couple = 1)	-1.32	0.27	0.02	-1.35	-1.28	0.26	0.28	-73.12	<0.005	inf
Number of Children	0.74	2.10	0.01	0.73	0.76	2.08	2.13			

Table 8: Impact of Family Structure: Cox Proportional Hazards Model Output

Approach 2: Pooled OLS Secondly, we estimated a pooled OLS regression model, which is a mixed model, controlling for the nested repeated measures over time, that is most suitable to the panel data structure of Dataset B, in which we observe repeated cross sections over time. Using pooled OLS, we estimated the effect of the same covariates as in Approach 1 (Family Type and Number of Children) on the total award received by the household in a given month.

The model output for the original and synthetic data are shown in Table 9. The models shows some minor differences between teams, but the most noteworthy is again the overestimation by team B and the underestimation by team C of the time variable, especially in the first months. It can also be

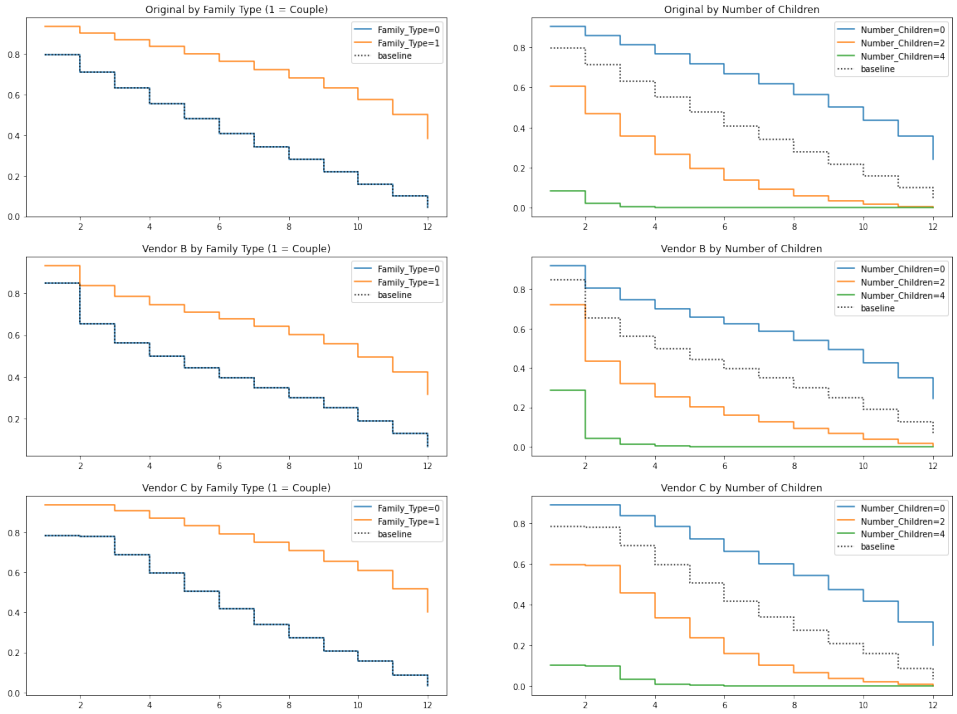

Figure 13: Impact of Family Structure: Conditional Survival Plots. The horizontal axes indicate time in months since the start of the series, and the vertical axis for each plot is the probability that the household does not receive Child Credit (that is, the probability of 'survival', or the event of receiving a positive amount of child allowance not occurring).

seen that team C greatly overestimates the effect of later months over the total award.

	Original data			Team B Synthetic Data			Team C Synthetic Data		
	Parameter	Std. Err.	T-stat	Parameter	Std. Err.	T-stat	Parameter	Std. Err.	T-stat
constant	1699	8.2277	206.5	1465.6	8.5527	171.37	1572.1	8.2025	191.66
Family Type (Single = 1, Couple = 0)	-188.18	4.4884	-41.927	-109.74	5.0644	-21.669	-40.298	4.5611	-8.8353
Number of Children	832.69	1.7025	489.1	856.21	1.8674	458.5	838.12	1.6972	493.83
Month 2 (indicator)	8.2606	9.8392	0.8396	91.173	9.7713	9.3307	-5.9798	9.6867	-0.6173
Month 3 (indicator)	16.774	9.8386	1.7049	117.49	9.753	12.046	41.598	9.6855	4.2949
Month 4 (indicator)	21.863	9.8405	2.2217	126.01	9.7726	12.894	86.594	9.6879	8.9384
Month 5 (indicator)	19.839	9.8444	2.0153	143.02	9.768	14.641	148.16	9.6956	15.282
Month 6 (indicator)	37.734	9.8496	3.831	158.31	9.7736	16.197	182.5	9.7061	18.803
Month 7 (indicator)	34.868	9.8617	3.5357	149.65	9.7583	15.335	188.02	9.7238	19.336
Month 8 (indicator)	44.667	9.8804	4.5207	148.09	9.8066	15.101	202.48	9.7421	20.784
Month 9 (indicator)	55.513	9.8993	5.6078	138.6	9.81	14.128	211.67	9.7652	21.676
Month 10 (indicator)	45.175	9.9183	4.5547	123.2	9.7958	12.577	251.4	9.785	25.693
Month 11 (indicator)	50.539	9.9372	5.0859	120.04	9.8139	12.232	285.62	9.8154	29.099
Month 12 (indicator)	59.089	9.9557	5.9352	131.62	9.8206	13.403	253.26	9.8405	25.737

Table 9: Impact of Family Structure: Pooled OLS Regression Output

Question 2: The distribution of Universal Credit awards across occupations

We then investigated the variation in Universal Credit award across occupations. In the data provided by DWP, the Occupation variable was generated by Faker with has a very large support of possible text values. Moreover, in the true DWP data, occupation is a free text field. To therefore pool the occupations into meaningful groups, we used string matching to map the list of observed occupations into the UK Standard Occupational Classification (SOC)[6], to use the International Standard Classification of Occupations 2008 (ISCO-08)[7] to group the observations.

We then plotted the total Universal Credit award by occupational group, both unconditionally and conditioning on basic demographics by first regressing total award on age and gender of the primary household member, and number of children. These plots are shown in Figures 14 and 15 respectively

[6] https://www.ons.gov.uk/methodology/classificationsandstandards/standardoccupationalclassificationsoc/soc2020/soc2020volume2codingrulesandconventions

[7] https://www.ilo.org/public/english/bureau/stat/isco/isco08/

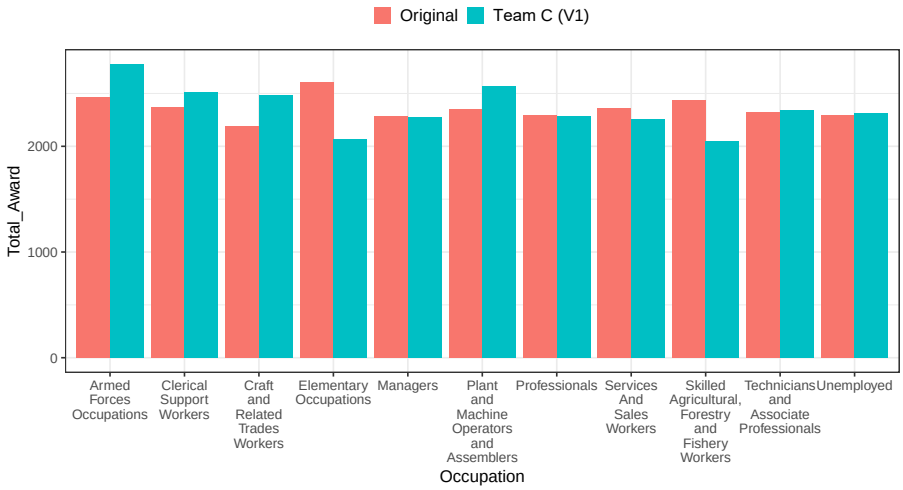

Figure 14: Total Universal Credit Award by Occupation

for the original data and team C's synthetic data. team B did not generate occupation labels (instead denoting occupation by "*").

We can see that occupation has little explanatory power for total award both in the original and synthetic data, as confirmed by the regression analysis in the earlier section. This is the case even in the unconditional case. As occupation in the original data provided by DWP was itself generated by Faker, with no association with the rest of the dataset, this is unsurprising. We would expect to observe a stronger relationship in the real data held by DWP.

We can also see that the distribution of team C's synthetic data does not match that of the original data very strongly. This is reflected in the EDA described above.

Question 3: The causes of drop-outs from Universal Credit

How can we understand the causes of drop-outs from Universal Credit, and predict them? This was the final modelling question that we investigated. The dataset provided by DWP only contains basic demographic information and observed awards for a subset of elements of Universal Credit, and so did not

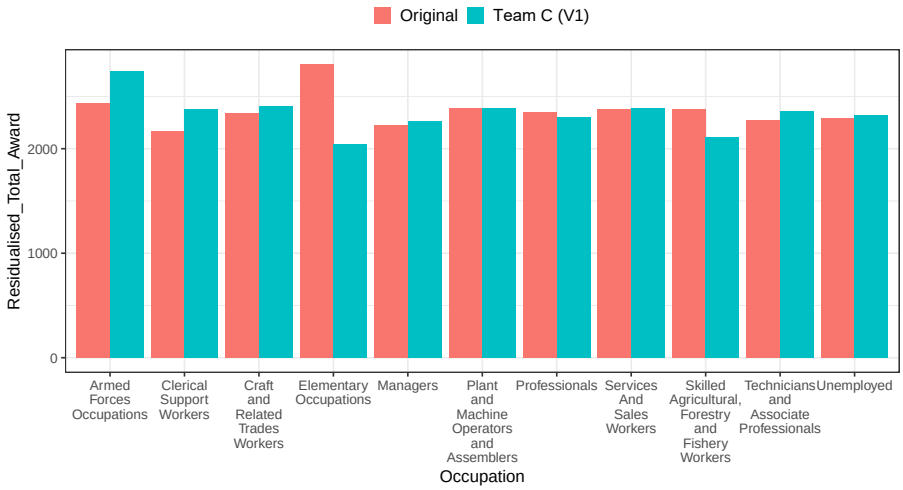

Figure 15: Residualised Total Universal Credit Award by Occupation

include any variables that could provide identifying variation to estimate the *causes* of drop-outs. We therefore treated this as an exploratory exercise in prediction, and interpret the results as conditional correlations, rather than causal effects.

Since this can be viewed as an exercise of prediction rather than as an evaluation of the synthetic datasets, we carried out the modelling on the original dataset only. We hope that this provides some insights for the possible applications of the synthetic dataset among researchers.

We consider two approaches. The problem of dropout prediction from a panel or longitudinal dataset can be interpreted as a survival problem, similarly to that of Question 1. Hence our first strategy adopts survival regression analysis, and the second approach explores zero-inflated negative binomial regression.

Approach 1: Survival regression analysis Similarly to Approach 1 to investigate the impact of family structure, we implement Cox's proportional hazard model to assess the relative contribution of demographic characteristics to drop out from the pool of UC recipients.

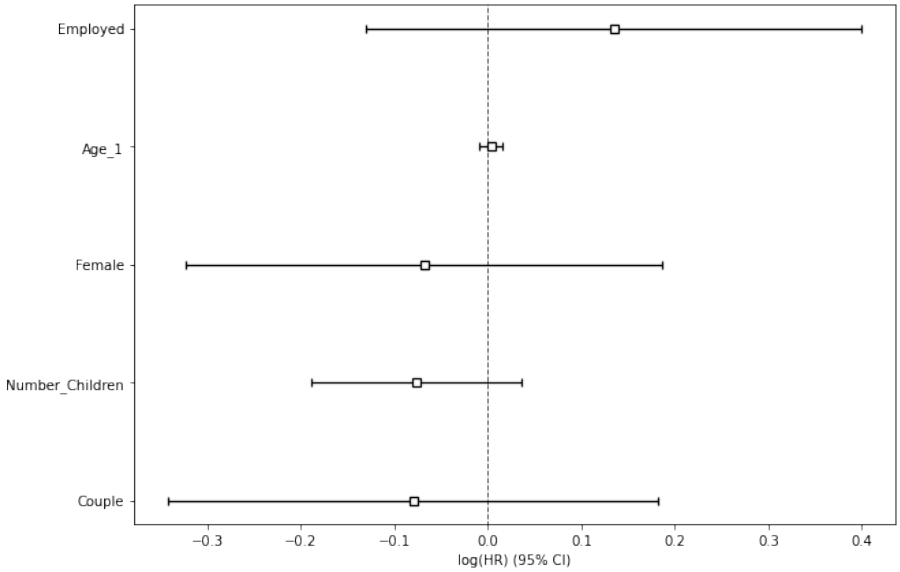

Figure 16: Drop-outs: Cox Proportional Hazards Model Coefficients

In this case in Equation 9, the hazard function $h(t)$ is the probability of dropping out of the dataset, and the set of covariates X consists of the family type (Couple = 1, Single = 0) and number of children in the household, and the employment status (Employed = 1, Unemployed = 0), age and gender (Female = 1, Male = 0) of the primary household member.

The estimated coefficients and their associated 95% confidence intervals are plotted in Figure 16. The wide confidence intervals, all including zero, confirm the low explanatory power which is reflected across the experiments conducted in this project, partly attributed to the simulated nature of the original data. Nonetheless, the conditional survival plots, shown in Figure 17 demonstrate that the respective magnitudes of the impact of Number of Children and Employment status are large.

Approach 2: Zero-inflated negative binomial (ZINB) regression Secondly, we estimated a ZINB regression model. ZINB regression combines a negative binomial distribution and logit distribution to model binary data with excess zeroes, where an event in this context is dropout over the twelve month

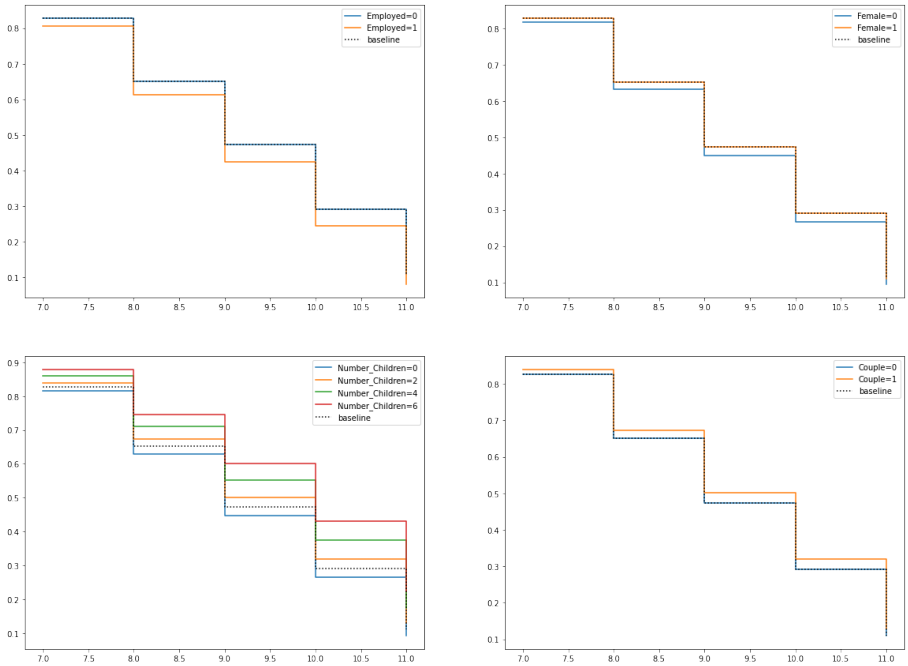

Figure 17: Drop-outs: Conditional Survival Plots. The horizontal axes indicate time in months since the start of the series, and the vertical axis for each plot is the probability that the household does not drop out (that is, the probability of 'survival', or that drop-out does not occur.

observation period. We therefore considered this model appropriate to reflect the rareness of dropout from the UC data: in dataset B, only 250 of the 20 000 households are observed to drop out.

The estimated coefficients are shown in Table 10. Unfortunately, the model failed to produce standard errors for the coefficient estimates, possibly due to the very small correlations across variables in the data.

	Coefficient	std err	z
Number of Children (LG)	-0.3398	NaN	NaN
Age (LG)	-0.1146	NaN	NaN
Couple (indicator) (LG)	6.3618	NaN	NaN
Employed (indicator) (LG)	-0.3132	NaN	NaN
Female (indicator) (LG)	-0.2233	NaN	NaN
Number of Children (NB)	-0.2639	NaN	NaN
Age (NB)	-0.0953	NaN	NaN
Couple (indicator) (NB)	1.4321	NaN	NaN
Employed (indicator) (NB)	-0.3908	NaN	NaN
Female (indicator) (NB)	-0.3078	NaN	NaN
constant	1.202×10^{-5}	NaN	NaN

Table 10: Causes of Dropout: ZINB regression coefficients. LG denotes the logit component which inflates the probability of a zero event, and NB denotes the negative binomial component. Unfortunately, the model failed to produce standard errors for the coefficient estimates (resulting in 'NaN' values for standard error and z), possibly due to the very small correlations across variables in the data.

5.3 Robustness

We conceive of robustness as a measure of the ability of a process or system to retain its use value in a range of conditions and environments. In the case of machine learning models, this property is conceptually quite straightforward—does the model perform to an acceptable standard, retaining the ability to return useful inferences, when exposed to shifts in the underlying data [51]?

In order for a generated dataset to demonstrate robustness however, it must however possess several attributes. First, the format of attributes must enforce the structure that obtains in the source data, i.e. when a column represents a UK Postal Code, it should conform to the real constraints rather than producing a plausible but invalid facsimile. We refer to that process as Format Validation—a set of filters was developed for testing compliance with the constraints presented by this project. We include these validation filters in the code artefact.

Second, adding a certain proportion of erroneous examples to the original data from which the synthetic set is generated should not cause a disproportionate loss of coherence or similarity to the original set. When applied to a machine learning context, this is usually referred to as Adversarial Robustness [4, 50, 23].

We considered other avenues of robustness testing, including the generation of textual representations from table data to provide an intuitive set of results that could prove easier for human sanity checking [21]. However, human evaluation proved impractical to carry out with a small team of researchers and limited resources.

5.3.1 Adversarial robustness

We follow the schema of Goodfellow et al. [19] in performing adversarial training and testing with our generative network. We create adversarial examples, perturbed versions of our training examples intended to maximise the probability of making inaccurate predictions with a learning model designed

trained to predict the total award to a claimant based on the generated record fields.

These examples have been used in two ways:

1. We added a small proportion of adversarial examples to the original dataset to establish the effect of this perturbed dataset on a standard battery of regression tests
2. We also used this procedure with our generated datasets, as well as the team-provided sets, in order to compare results with the baseline. This allowed us to determine how robust the synthetic sets are, and whether new sources of error have been introduced during generation.

When generating our testing examples, we attempted first to use the AutoAttack framework developed by Croce et al. [8] in order to obtain examples from several models in a single testing run [7, 2]. However, adapting this framework from the original classification intent to our need for regression-based testing proved impractical in the limited time available.

Instead, we adopted the system proposed by Neutatz et al. [38] for adversarial regression testing which extends the IBM Adversarial Robustness Toolkit [39]. We first generate and fit a linear regression model against our dataset, predicting the `Total_Award` column from preceding demographic data, then use the Fast Gradient Method proposed by Goodfellow to generate adversarial examples, which are then added to our test set.

We compare the predictions from our regression model to the target value for each record to determine how much error has been added, results for which can be seen in Table 11 and Figure 18.

We can see from the results here that the metrics for testing carried out with sets derived from Dataset A are relatively similar, which provides some evidence that the datasets are reasonably robust. However, the results for sets derived from Dataset B are more mixed, with marked reduction in performance across all synthetic datasets—this suggests an issue with the

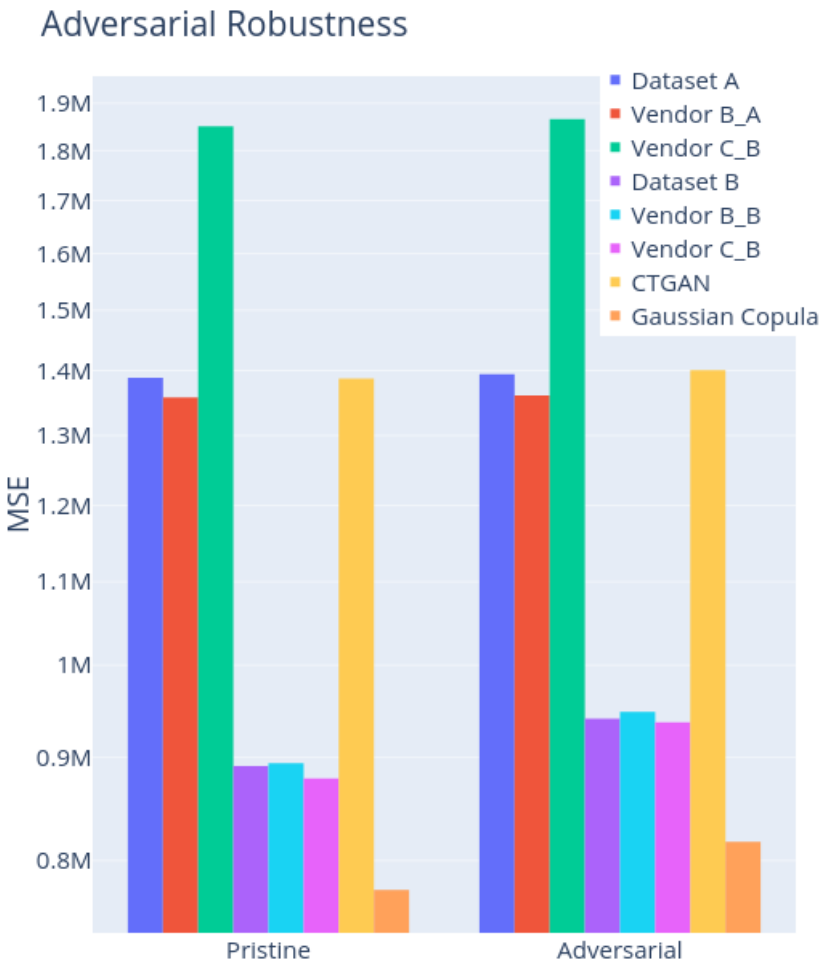

Figure 18: Robustness tests carried without (Pristine) and with adversarial perturbation (Adversarial). MSE = Mean Squared Error.

	Pristine			Perturbed		
	MSE	MAE	R^2	MSE	MAE	R^2
Dataset A	1388621	537.08	0.08071	1394003	534.01	0.07715
Team B	**1357668**	**535.18**	**0.07700**	**1360565**	**530.58**	**0.07503**
Team C	1850229	669.50	0.05817	1866003	674.93	0.05014
Dataset B	891226	687.08	0.54062	940699	714.72	0.51511
Team B	**893910**	**682.28**	0.50435	948009	**712.22**	0.47435
Team C	878847	715.88	**0.54143**	**936713**	750.61	**0.51124**
CTGAN	1387336	1005.02	0.00794	1400643	1011.6	-0.00158
GC	773519	674.083	0.57873	817335	699.123	0.55487

Table 11: Adversarial testing results

robustness of these datasets. This may be partially explained by the large size differential between the original sets.

5.4 Fairness

Fairness is concerned with the difference in outcomes from a particular model or dataset that may be caused by societal biases [65]. For example, it has been found that image recognition datasets contain fewer examples of household items from lower-income countries, and hence models developed with those datasets would perform less well for users in those places [58].

We extend this definition of fairness to the case of synthetic data in the following way: The process of substituting synthetic data for real data should not introduce new sources of bias against individuals that are not already present in the dataset. We can test if this has occurred by checking for the rate of disparate outcomes given a particular demographic attribute that may potentially occur in our original set, and repeat that process for all sets generated from it. If a test set has not become less fair, we would expect the results to be similar.

We adopt the process proposed by Feldman et al. [12], under which we define the disparate impact (DI) of a dataset D as:

Definition 5.4 (Disparate Impact)**.** *Given dataset $D = (X, Y, C)$ where X is a personal attribute such as race or gender, Y is all remaining attributes, and C is a target variable class that will be predicted with potential positive outcome pos, the disparate impact of D with respect to X is*

$$\frac{Pr(C = pos|X = 0)}{Pr(C = pos|X = 1)}.$$

In our experimental setup, we specify the protected variable X as the gender of the respondent, with the 1 value corresponding to Male. We generate our classes C by binning all `Total_ Award/Allowance` values into three equal categories, and designating the top third as our *pos* class. We then return the ratio of the probability of a positive allowance outcome given a non-Male gender to the probability given a Male gender. The results can be seen in Table 12.

Dataset	DI Ratio	Difference
Dataset A	1.008 74	
Team B	1.001 98	0.006 76
Team C	0.859 71	0.149 03
Dataset B	0.996 20	
Team B	1.001 98	−0.005 78
Team C	1.030 12	−0.033 92
CTGAN	0.982 81	0.013 39
GC	0.981 14	0.015 06

Table 12: Disparate Impact Ratio for each set. The synthetic dataset with the smallest difference from the original dataset is shaded—team B in each case.

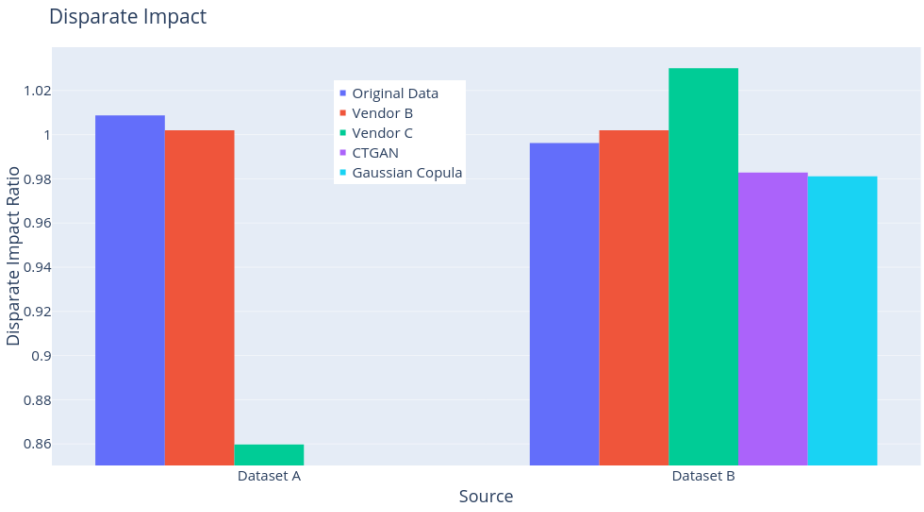

Figure 19: Disparate Impact Ratio. A value of 1 represents perfect equality of outcome probability for the chosen demographic attribute, here Gender.

As the plot in Figure 19 shows, results are reasonably similar for this test with the exception of the version of dataset A provided by team C. We observe a large drop in the DI ratio here, indicating that the outcome bias has been disturbed by the generation process. Without further data regarding the setup of the generation process, it would be inappropriate to make confident predictions about the source of this change in the level of disparate outcomes. However, we can state that one possible source of a shift of this type is from sub-sampling an initial dataset—if the sampled data happens to include an imbalance with regards to a particular variable, the data generated from it could conceivably exhibit this kind of behaviour.

6 Future work and research avenues

As mentioned during the earlier sections of this work, we considered a number of fruitful avenues for research that we could not fully explore due to the constraints of the challenge.

6.1 Generation

Using a trained model to generate the textual fields in the dataset, based on the conditional distribution of the other data as defined in Section 4.3, would be expected to improve utility and similarity over the current approach.

We would also suggest the consideration of differential privacy for use in model training, as a method for ensuring quantifiable row-level privacy. This could be achieved, for example, by applying a calibrated noise sample to the objective function of the discriminator, increasing the uncertainty in learning specific attributes [61, 27, 54, 34]. One promising approach is the application of local differential privacy. Under this schema, samples would be perturbed as inputs before being processed by the generative model [28, 60, 66]. In this sense, the uncertainty added by differential privacy would extend to entire model, offering the possibility of releasing trained models, perhaps alongside the synthetic data they were used to produce.

Including private-attribute-adversarial training could offer improvements to privacy in generation. Originally conceived to adapt models for domain generalisation [16], this technique has been adapted to extend to training specifically for improved insensitivity to selected private attributes [6, 63]. In this scenario, a complementary classifier is trained alongside the generative model in an attempt to infer the value of a private attribute from the intermediate representations within the network, the loss of which is added as a new negative term in the main objective function, promoting representations which obscure the target feature.

6.2 Evaluation

There are a great number of evaluation metrics that we did not consider during this research, all of which would be could offer insight into the disclosure risk inherent in a generated dataset. Among these are k-anonymity [24, 5, 1], l-diversity [35, 43], t-closeness [40, 9].

We also note that for this challenge, privacy risk was taken to mean the risk of re-identification. Other definitions are possible, which are then open to other privacy attacks that might be considered for a dataset of this type: for example, membership inference [48]. In this attack, data about individuals gathered from other sources is used with our outputs to determine whether an individual was part of the set used to train a particular model. This may present reputational risk, if, say, the model was used to generate a set of data for cancer patients or potential victims of fraud. There are a number of potential ways to quantify and measure this risk, including measuring the compound information leakage per row of released set or model output [46, 32, 11].

We also considered using the discriminator layers of a trained generative model to determine the probability that a particular dataset row was generated with that model—indeed, this property could form a useful vector of attack if there exist known flaws in the model selected [67], or allow the discovery of private model information [56, 14, 49].

In terms of novel evaluation techniques, we considered the application of table-to-text generation models [21] in order to produce a more easily-understood representation of data rows for human evaluators. A structured textual output may provide an efficient way to implement basic sanity checking more quickly by pointing out obvious outliers.

Team members

Oleksandr Deineha is a PhD student at V. N. Karazin Kharkiv National University in Machine Learning and Distributed Systems. He contributed to this project by developing the model based on CT-GAN for synthetic data generation and exploratory data analysis.

Charlotte Grace is a DPhil student at the University of Oxford in Economics. She contributed to this project by providing facilitation and developing and implementing the utility evaluation framework.

Samruddhi Mhatre is pursuing her MSc in Data Science from the University of Bristol. She contributed to this project by studying and implementing the existing CT-GAN model for synthetic data generation.

Tanut Treetanthiploet just finished the DPhil at the Mathematical Institute at the University of Oxford. He contributed to this project on the development of the Gaussian copula method for the data generation and marginal error evaluation.

Daniel Valdenegro is a Computational Social Science PhD student at the University of Leeds. He contributed to this project in the development of fairness metrics and privacy attacks methods, as well in the analysis of utility of the datasets.

Liang Zhou is a PhD student at University College London in theoretical neuroscience. He studied the myriad of metrics for distributional similarity as applied to this project.

Eden Packer is an economist on the Civil Service Fast Stream, with an educational background in economics, mathematics and policy. With a keen interest in data science's potential to refine governmental analysis, he has worked in data modelling and benefit forecasting. Through working on the department's richest Universal Credit dataset, Eden has supported the Innovation Lab as both data consultant, and representative for the Analytical Community in DWP.

Shruti Kohli is lead data scientist currently working as an Innovation Lead in the DWP Innovation lab , driving innovation projects to support DWP services. The role involves leading the innovation process, including triaging new ideas, discovering opportunities and advancing concepts which lead to new or improved product offerings, providing in-depth technical expertise throughout research, discovery, new product development, using Cloud Analytics for developing high performing machine learning models. She more than a decade of experience in leading digital transformation, data innovation, leadership and culture change projects. She has a PhD in Computer sciences, being an academician in past, this has given her a good appetite to learn quickly and share. Her career developments have come a long way mentoring, leading data and tech-driven projects, building relationships.

Aatish Thakerar Technical Delivery manager for Innovation lab. Have strong experience in handling data/Tech/AI delivery projects. Good background in Cloud and Machine Learning.

Richard Plant is a PhD student at Edinburgh Napier University researching statistical privacy methods for natural language processing. He contributed to this project by providing facilitation, technical assistance, background research, and development of the evaluation framework and utility met-rics.

References

[1] Charu C Aggarwal. “On k-Anonymity and the Curse of Dimensionality”. In: *Proceedings of the 31st International Conference on Very Large Data Bases*. VLDB ’05. Trondheim, Norway: VLDB Endowment, 2005. ISBN: 1-59593-154-6. (Visited on 03/11/2020).

[2] Maksym Andriushchenko et al. “Square Attack: A Query-Efficient Black-Box Adversarial Attack via Random Search”. en. In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi et al. Vol. 12368. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 484–501. ISBN: 978-3-030-58591-4 978-3-030-58592-1. DOI: 10.1007/978-3-030-58592-1_29. URL: https://link.springer.com/10.1007/978-3-030-58592-1_29 (visited on 04/26/2021).

[3] Dalal Al-Azizy et al. “Deanonymisation in Linked Data: A research roadmap”. In: *2014 World Congress on Internet Security, WorldCIS 2014*. Institute of Electrical and Electronics Engineers Inc., Jan. 2014, pp. 48–52. ISBN: 978-1-908320-42-1. DOI: 10.1109/WorldCIS.2014.7028165. (Visited on 10/27/2020).

[4] Max Bartolo et al. “Improving Question Answering Model Robustness with Synthetic Adversarial Data Generation”. In: *arXiv:2104.08678 [cs]* (Apr. 2021). arXiv: 2104.08678. URL: http://arxiv.org/abs/2104.08678 (visited on 04/23/2021).

[5] Claudio Bettini, X. Sean Wang, and Sushil Jajodia. “The Role of Quasi-identifiers in k-Anonymity Revisited”. In: (Nov. 2006). arXiv: cs/0611035. URL: http://arxiv.org/abs/cs/0611035 (visited on 03/04/2021).

[6] Maximin Coavoux, Shashi Narayan, and Shay B. Cohen. “Privacy-preserving Neural Representations of Text”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 1–10. DOI: 10.18653/v1/D18-1001. URL: 10.18653/v1/D18-1001 (visited on 04/20/2021).

[7] Francesco Croce and Matthias Hein. “Minimally distorted Adversarial Examples with a Fast Adaptive Boundary Attack”. en. In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. 2020, p. 10.

[8] Francesco Croce and Matthias Hein. “Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks”. In: *arXiv:2003.01690 [cs, stat]* (Aug. 2020). arXiv: 2003.01690. URL: http://arxiv.org/abs/2003.01690 (visited on 04/26/2021).

[9] J. Domingo-Ferrer and J. Soria-Comas. “From t-closeness to differential privacy and vice versa in data anonymization”. In: *Knowledge-Based Systems* 74 (Dec. 2015). arXiv: 1512.05110 Publisher: Elsevier B.V., pp. 151–158. DOI: 10.1016/j.knosys.2014.11.011. URL: http://arxiv.org/abs/1512.05110 (visited on 03/04/2021).

[10] Cynthia Dwork. “Differential privacy”. In: *Proceedings of the 33rd international conference on Automata, Languages and Programming - Volume Part II*. ICALP’06. Berlin, Heidelberg: Springer-Verlag, July 2006, pp. 1–12. ISBN: 978-3-540-35907-4. DOI: 10.1007/11787006˙1. URL: 10.1007/11787006˙1 (visited on 04/20/2021).

[11] Farhad Farokhi and Mohamed Ali Kaafar. “Modelling and Quantifying Membership Information Leakage in Machine Learning”. In: (Jan. 2020). arXiv: 2001.10648. URL: http://arxiv.org/abs/2001.10648 (visited on 10/26/2020).

[12] Michael Feldman et al. “Certifying and removing disparate impact”. In: *arXiv:1412.3756 [cs, stat]* (July 2015). arXiv: 1412.3756. URL: http://arxiv.org/abs/1412.3756 (visited on 04/23/2021).

[13] M. Fernández-Delgado et al. “An extensive experimental survey of regression methods”. en. In: *Neural Networks* 111 (Mar. 2019), pp. 11–34. ISSN: 0893-6080. DOI: 10/ggh652. URL: https://www.sciencedirect.com/science/article/pii/S0893608018303411 (visited on 04/27/2021).

[14] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. “Model inversion attacks that exploit confidence information and basic countermeasures”. In: *Proceedings of the ACM Conference on Computer and Communications Security*. Vol. 2015-Octob. ISSN: 15437221. New York, New York, USA: Association for Computing Machinery, Oct. 2015, pp. 1322–1333. ISBN: 978-1-4503-3832-5. DOI: 10.1145/2810103.

2813677. URL: http://dl.acm.org/citation.cfm?doid=2810103.2813677 (visited on 03/11/2020).

[15] Keith B. Frikken and Yihua Zhang. “Yet another privacy metric for publishing micro-data”. In: *Proceedings of the 7th ACM workshop on Privacy in the electronic society*. WPES ’08. New York, NY, USA: Association for Computing Machinery, Oct. 2008, pp. 117–122. ISBN: 978-1-60558-289-4. DOI: 10/dk3zzr. URL: https://doi.org/10.1145/1456403.1456423 (visited on 04/30/2021).

[16] Yaroslav Ganin et al. “Domain-adversarial training of neural networks”. In: *The Journal of Machine Learning Research* 17.1 (Jan. 2016), pp. 2096–2030. ISSN: 1532-4435.

[17] Quan Geng et al. “Privacy and Utility Tradeoff in Approximate Differential Privacy”. In: *arXiv:1810.00877 [cs]* (Feb. 2019). arXiv: 1810.00877. URL: http://arxiv.org/abs/1810.00877 (visited on 03/05/2021).

[18] Neil Zhenqiang Gong and Bin Liu. “You are who you know and how you behave: Attribute inference attacks via users’ social friends and behaviors”. In: *Proceedings of the 25th USENIX security symposium*. 2016, pp. 979–995. ISBN: 978-1-931971-32-4. URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/gong.

[19] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. “Explaining and Harnessing Adversarial Examples”. In: *ICLR (Poster)*. arXiv: 1412.6572. San Diego, CA, USA, Mar. 2015. URL: http://arxiv.org/abs/1412.6572 (visited on 04/26/2021).

[20] Arthur Gretton et al. “A Kernel Two-Sample Test”. In: *Journal of Machine Learning Research* 13.25 (2012), pp. 723–773. URL: http://jmlr.org/papers/v13/gretton12a.html (visited on 04/30/2021).

[21] Hamza Harkous, Isabel Groves, and Amir Saffari. “Have Your Text and Use It Too! End-to-End Neural Data-to-Text Generation with Semantic Fidelity”. In: *Proceedings of the 28th International Conference on*

Computational Linguistics. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 2410–2424. DOI: 10/gjtk7v. URL: 10/gjtk7v (visited on 04/28/2021).

[22] Markus Hittmeir, Andreas Ekelhart, and Rudolf Mayer. "Utility and Privacy Assessments of Synthetic Data for Regression Tasks". In: *2019 IEEE International Conference on Big Data (Big Data)*. Los Angeles, CA, USA: IEEE, Dec. 2019, pp. 5763–5772. ISBN: 978-1-72810-858-2. DOI: 10.1109/BigData47090.2019.9005476. URL: https://ieeexplore.ieee.org/document/9005476/ (visited on 04/21/2021).

[23] Chia-Yi Hsu et al. "Adversarial Examples for Unsupervised Machine Learning Models". In: *arXiv:2103.01895 [cs]* (Apr. 2021). arXiv: 2103.01895. URL: http://arxiv.org/abs/2103.01895 (visited on 04/23/2021).

[24] Dai Ikarashi et al. "K-anonymous microdata release via post randomisation method". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 9241. arXiv: 1504.05353 ISSN: 16113349. Springer Verlag, Apr. 2015, pp. 225–241. ISBN: 978-3-319-22424-4. DOI: 10.1007/978-3-319-22425-1˙14. URL: http://arxiv.org/abs/1504.05353 (visited on 03/04/2021).

[25] Magnus Jändel. "Decision support for releasing anonymised data". In: *Computers and Security* 46 (Oct. 2014). Publisher: Elsevier Ltd, pp. 48–61. ISSN: 01674048. DOI: 10.1016/j.cose.2014.07.001. (Visited on 10/27/2020).

[26] joke2k. *Faker: Faker Is a Python Package That Generates Fake Data for You.*

[27] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. "PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees". en. In: Sept. 2018. URL: https://openreview.net/forum?id=S1zk9iRqF7 (visited on 04/20/2021).

[28] Yilin Kang et al. “Input Perturbation: A New Paradigm between Central and Local Differential Privacy”. In: (Feb. 2020). arXiv: 2002.08570. URL: http://arxiv.org/abs/2002.08570 (visited on 04/26/2020).

[29] Tiancheng Li and Ninghui Li. “On the tradeoff between privacy and utility in data publishing”. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2009, pp. 517–525. ISBN: 978-1-60558-495-9. DOI: 10.1145/1557019.1557079. (Visited on 02/28/2021).

[30] Tiancheng Li, Ninghui Li, and Jian Zhang. “Modeling and integrating background knowledge in data anonymization”. In: *Proceedings - International Conference on Data Engineering*. ISSN: 10844627. 2009, pp. 6–17. ISBN: 978-0-7695-3545-6. DOI: 10.1109/ICDE.2009.86. URL: www.nhlbi.nih.gov/health/public/lung/other/ (visited on 10/27/2020).

[31] Tianshi Li et al. “Decentralized is not risk-free: Understanding public perceptions of privacy-utility trade-offs in COVID-19 contact-tracing apps”. In: *arXiv preprint* (May 2020). arXiv: 2005.11957. URL: http://arxiv.org/abs/2005.11957 (visited on 11/02/2020).

[32] Xiyang Liu et al. “MACE: A Flexible Framework for Membership Privacy Estimation in Generative Models”. In: (Sept. 2020). arXiv: 2009.05683. URL: http://arxiv.org/abs/2009.05683 (visited on 10/26/2020).

[33] Xuanqing Liu and Cho-Jui Hsieh. “Rob-GAN: Generator, Discriminator, and Adversarial Attacker”. In: *arXiv:1807.10454 [cs, stat]* (Apr. 2019). arXiv: 1807.10454. URL: http://arxiv.org/abs/1807.10454 (visited on 04/28/2021).

[34] Chuan Ma et al. “RDP-GAN: A Renyi-Differential Privacy based Generative Adversarial Network”. In: (July 2020). Publication Title: arXiv arXiv: 2007.02056. URL: http://arxiv.org/abs/2007.02056 (visited on 12/01/2020).

[35] A. Machanavajjhala et al. “L-diversity: privacy beyond k-anonymity”. In: *22nd International Conference on Data Engineering (ICDE’06)*. ISSN: 2375-026X. Apr. 2006, pp. 24–24. DOI: 10.1109/ICDE.2006.1. URL: 10.1109/ICDE.2006.1.

[36] Tehila Minkus et al. “The city privacy attack: Combining social media and public records for detailed profiles of adults and children”. In: *COSN 2015 - Proceedings of the 2015 ACM Conference on Online Social Networks*. New York, New York, USA: Association for Computing Machinery, Inc, Nov. 2015, pp. 71–81. ISBN: 978-1-4503-3951-3. DOI: 10.1145/2817946.2817957. URL: http://dl.acm.org/citation.cfm?doid=2817946.2817957 (visited on 10/26/2020).

[37] Arvind Narayanan and Vitaly Shmatikov. “Robust de-anonymization of large sparse datasets”. In: *Proceedings - IEEE Symposium on Security and Privacy*. ISSN: 10816011. 2008, pp. 111–125. ISBN: 978-0-7695-3168-7. DOI: 10.1109/SP.2008.33.

[38] Felix Neutatz, Felix Biessmann, and Ziawasch Abedjan. “Enforcing Constraints for Machine Learning Systems via Declarative Feature Selection: An Experimental Study”. In: *ACM SIGMOD/PODS International Conference on Management of Data*. Xi’an, Shaanxi, China, June 2021. DOI: 10.1145/3448016.3457295.

[39] Maria-Irina Nicolae et al. “Adversarial Robustness Toolbox v1.0.0”. In: *arXiv:1807.01069 [cs, stat]* (Nov. 2019). arXiv: 1807.01069. URL: http://arxiv.org/abs/1807.01069 (visited on 04/27/2021).

[40] Li Ninghui, Li Tiancheng, and Suresh Venkatasubramanian. “t-Closeness: Privacy beyond k-anonymity and l-diversity”. In: *Proceedings - International Conference on Data Engineering*. ISSN: 10844627. 2007, pp. 106–115. ISBN: 1-4244-0803-2. DOI: 10.1109/ICDE.2007.367856. (Visited on 03/04/2021).

[41] Jiaqi Pan et al. “Twitter homophily: Network based prediction of user’s occupation”. In: *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*. Association for Computational Linguistics (ACL), 2020, pp. 2633–2638.

ISBN: 978-1-950737-48-2. DOI: 10.18653/v1/p19-1252. (Visited on 10/26/2020).

[42] Neha (Neha R.) Patki. “The Synthetic Data Vault : generative modeling for relational databases”. eng. Accepted: 2017-06-06T18:44:28Z Journal Abbreviation: SDV : generative modeling for relational databases. Thesis. Massachusetts Institute of Technology, 2016. URL: https://dspace.mit.edu/handle/1721.1/109616 (visited on 04/29/2021).

[43] Ajay Prasad et al. “Applying l-Diversity in anonymizing collaborative social network”. In: *IJCSIS International Journal of Computer Science and Information Security* 8.2 (July 2010). arXiv: 1007.0292. URL: http://arxiv.org/abs/1007.0292 (visited on 03/04/2021).

[44] Vibhor Rastogi, Dan Suciu, and Sungho Hong. “The boundary between privacy and utility in data publishing”. In: *33rd International Conference on Very Large Data Bases, VLDB 2007 - Conference Proceedings*. arXiv: cs/0612103. Dec. 2007, pp. 531–542. ISBN: 978-1-59593-649-3. URL: http://arxiv.org/abs/cs/0612103 (visited on 03/04/2021).

[45] Lucas Rosenblatt et al. “Differentially Private Synthetic Data: Applied Evaluations and Enhancements”. In: *arXiv* (Nov. 2020). arXiv: 2011.05537. URL: http://arxiv.org/abs/2011.05537 (visited on 12/01/2020).

[46] Sara Saeidian et al. “Quantifying Membership Privacy via Information Leakage”. In: (Oct. 2020). arXiv: 2010.05965. URL: http://arxiv.org/abs/2010.05965 (visited on 10/26/2020).

[47] Pierangela Samarati. “Protecting respondents’ identities in microdata release”. In: *IEEE Transactions on Knowledge and Data Engineering* 13.6 (2001), pp. 1010–1027. ISSN: 10414347. DOI: 10.1109/69.971193. URL: http://www.dti.unimi.it/ (visited on 03/04/2021).

[48] Reza Shokri et al. “Membership Inference Attacks Against Machine Learning Models”. In: *Proceedings - IEEE Symposium on Security and Privacy*. arXiv: 1610.05820 ISSN: 10816011. Institute of Electrical and Electronics Engineers Inc., June 2017, pp. 3–18. ISBN: 978-1-5090-5532-6. DOI: 10.1109/SP.2017.41. (Visited on 10/26/2020).

[49] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. “Machine learning models that remember too much”. In: *Proceedings of the ACM Conference on Computer and Communications Security*. arXiv: 1709.07886 ISSN: 15437221. Sept. 2017, pp. 587–601. ISBN: 978-1-4503-4946-8. DOI: 10.1145/3133956.3134077. URL: http://arxiv.org/abs/1709.07886 (visited on 10/28/2020).

[50] David Stutz, Matthias Hein, and Bernt Schiele. “Disentangling Adversarial Robustness and Generalization”. In: *arXiv:1812.00740 [cs, stat]* (Apr. 2019). arXiv: 1812.00740. URL: http://arxiv.org/abs/1812.00740 (visited on 04/23/2021).

[51] Adarsh Subbaswamy, Roy Adams, and Suchi Saria. “Evaluating Model Robustness and Stability to Dataset Shift”. In: *arXiv:2010.15100 [cs, stat]* (Mar. 2021). arXiv: 2010.15100. URL: http://arxiv.org/abs/2010.15100 (visited on 04/23/2021).

[52] Xiaoxun Sun et al. “Publishing anonymous survey rating data”. In: *Data Mining and Knowledge Discovery* 23.3 (Nov. 2011). Publisher: Springer, pp. 379–406. ISSN: 13845810. DOI: 10.1007/s10618-010-0208-4. URL: http://www.imdb.com/. (visited on 10/27/2020).

[53] Yupan Tian et al. “Inferring private attributes based on graph convolutional neural network in social networks”. In: *Proceedings - 2019 International Conference on Networking and Network Applications, NaNA 2019*. Institute of Electrical and Electronics Engineers Inc., Oct. 2019, pp. 186–190. ISBN: 978-1-72812-629-6. DOI: 10.1109/NaNA.2019.00041. (Visited on 10/26/2020).

[54] Amirsina Torfi, Edward A. Fox, and Chandan K. Reddy. “Differentially Private Synthetic Medical Data Generation using Convolutional GANs”. In: *arXiv:2012.11774 [cs]* (Dec. 2020). arXiv: 2012.11774. URL: http://arxiv.org/abs/2012.11774 (visited on 04/20/2021).

[55] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. “DP-CGAN: Differentially Private Synthetic Data and Label Generation”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. arXiv: 2001.09700. Jan. 2019. URL: http://arxiv.org/abs/2001.09700 (visited on 04/26/2020).

[56] Florian Tramèr et al. “Stealing machine learning models via prediction APIs”. In: *Proceedings of the 25th USENIX Security Symposium*. arXiv: 1609.02943. USENIX Association, Sept. 2016, pp. 601–618. ISBN: 978-1-931971-32-4. URL: http://arxiv.org/abs/1609.02943 (visited on 10/27/2020).

[57] Ashish Vaswani et al. “Attention is all you need”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS’17. Pages: 6000–6010. Long Beach, California, USA: Curran Associates Inc., Dec. 2017. ISBN: 978-1-5108-6096-4. URL: http://arxiv.org/abs/1706.03762 (visited on 11/22/2019).

[58] Terrance de Vries et al. “Does Object Recognition Work for Everyone?” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2019, pp. 52–59. URL: https://openaccess.thecvf.com/content˙CVPRW˙2019/html/cv4gc/de˙Vries˙Does˙Object˙Recognition˙Work˙for˙Everyone˙CVPRW˙2019˙paper.html (visited on 04/27/2021).

[59] Isabel Wagner and David Eckhoff. “Technical privacy metrics: A systematic survey”. In: *ACM Computing Surveys* 51.3 (Apr. 2018). ISSN: 15577341. DOI: 10.1145/3168389.

[60] Qinglong Wang et al. “Using Non-invertible Data Transformations to Build Adversarial-Robust Neural Networks”. In: *arXiv:1610.01934 [cs]*

(Dec. 2016). arXiv: 1610.01934. URL: http://arxiv.org/abs/1610.01934 (visited on 04/23/2021).

[61] Liyang Xie et al. “Differentially private generative adversarial network”. In: *arXiv* (Feb. 2018). arXiv: 1802.06739 Publisher: arXiv. URL: http://arxiv.org/abs/1802.06739 (visited on 12/01/2020).

[62] Lei Xu et al. “Modeling Tabular data using Conditional GAN”. In: *arXiv:1907.00503 [cs, stat]* (Oct. 2019). arXiv: 1907.00503. URL: http://arxiv.org/abs/1907.00503 (visited on 04/19/2021).

[63] Qiongkai Xu et al. “Privacy-aware text rewriting”. In: *INLG 2019 - 12th International Conference on Natural Language Generation, Proceedings of the Conference*. Association for Computational Linguistics (ACL), 2019, pp. 247–257. ISBN: 978-1-950737-94-9. DOI: 10.18653/v1/w19-8633. URL: https://www.aclweb.org/anthology/W19-8633 (visited on 12/02/2020).

[64] Yabo Xu et al. “Anonymizing transaction databases for publication”. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, New York, USA: ACM Press, 2008, pp. 767–775. ISBN: 978-1-60558-193-4. DOI: 10.1145/1401890.1401982. URL: http://dl.acm.org/citation.cfm?doid=1401890.1401982 (visited on 10/27/2020).

[65] Kaiyu Yang et al. “Towards fairer datasets: filtering and balancing the distribution of the people subtree in the ImageNet hierarchy”. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. FAT* ’20. New York, NY, USA: Association for Computing Machinery, Jan. 2020, pp. 547–558. ISBN: 978-1-4503-6936-7. DOI: 10/gjs7sg. URL: https://doi.org/10.1145/3351095.3375709 (visited on 04/27/2021).

[66] Min Ye and Alexander Barg. “Optimal schemes for discrete distribution estimation under locally differential privacy”. In: *IEEE Transactions on Information Theory* 64.8 (Feb. 2018). arXiv: 1702.00610 Publisher: Institute of Electrical and Electronics Engineers Inc., pp. 5662–5676.

ISSN: 00189448. DOI: 10.1109/TIT.2018.2809790. URL: http://arxiv.org/abs/1702.00610 (visited on 11/13/2020).

[67] Yuheng Zhang et al. “The secret revealer: Generative model-inversion attacks against deep neural networks”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. arXiv: 1911.07135 ISSN: 10636919. Institute of Electrical and Electronics Engineers (IEEE), Nov. 2020, pp. 250–258. DOI: 10.1109/CVPR42600.2020.00033. URL: http://arxiv.org/abs/1911.07135 (visited on 10/26/2020).

turing.ac.uk
@turinginst