LTI Synthetic Data

Technical Report

ASC, 0259 D3 V1.2

Date: 08/06/2020

Author(s): Brijesh Patel, Gary Francis, Indika Wanninayake, Alan Pilgrim and Ben Upton (all BAE Systems Applied Intelligence Labs)

Supplied under terms of Contract No. DSTL/AGR/000616/01 – Task 259 – Study 3

This page is intentionally blank

Version history

Version	Date	Author	Action
1.0	9/4/20	A Pilgrim	First Release
1.1	4/6/20	A Pilgrim	Addressed comments from customer
1.2	8/6/20	A Pilgrim	Addressed comments from customer

Executive Summary

The Cyber and Data Science Capability of BAE Systems AI Labs completed this package of work for DSTL under the Logistics Technology Investigations (LTI) project to review the state of the art techniques in generating privacy-preserving synthetic data.

Synthetic data generation is an active area of research and is driven primarily by the need for sharing sensitive data for research and development purposes. Synthetically generated data should preserve the characteristics of real data while protecting the personal or sensitive information present in the original data. The primary aim for this is to allow external parties to access and use the synthetic data and to have a high degree of confidence that work performed on the synthetic dataset is transferable to the real dataset.

This project was split into two phases: research and experimentation. In the first phase, an in depth research into a number of areas including data obscuration methods, tools available for generating synthetic data, metrics used for assessing the quality of synthetic data as well as an investigation into potential open source datasets to support this work was conducted. Following this, a subset of these tools were selected and assessed using open source datasets relevant to LTI and by a set of metrics that assess how well the synthetic data matches the real dataset and if the sensitive information has been removed.

A summary of the outcome of this work include:

- The project identified a wide range of techniques to obscure sensitive or private information in datasets. These include statistical methods, deep learning techniques and natural language processing for a wide range of datatypes such as numeric, categorical, textual and geolocation data.
- Of all the techniques studied, Generative Adversarial Network (GAN) based techniques are the most active area of research. GANs have the potential to represent more complex distributions and relationships than basic statistical methods and can handle multiple data types within the same model. However, they can be difficult to train and training computation requirements and training time can be significant. Of all the other methods studied, many tools still uses statistical approaches and these are being explored and extended (e.g. for generating synthetic data from relational databases).
- A framework for assessing synthetic data generation tools was developed as part of this research and it looks into various aspects of synthetic data generation including factors that assess the quality of the outputs (e.g. versatility, privacy-utility trade-off) as well as the usability of the tools (e.g. ease of configuration, performance etc.). This framework can be used to assess and compare different data generation tools.
- The research into assessment metrics for synthetic data revealed a range of different qualitative and quantitative evaluation methods. The use of such methods depends on the type of analytical task to be performed on the synthetic data (e.g. for supervised machine learning – use F-measures, ROC (Receiver Operating Characteristic), RMSE (Root Mean Squared Error), for unsupervised machine learning – use dimension-wise prediction-based or dimension-wise distance-based methods). Statistical techniques such as distributional deviations and qualitative techniques (e.g. heat maps, feature histograms) can be used across all the datasets. However, it is important to note that interpreting the output of such metrics is not straightforward and would require expert knowledge in the field.

- Synthetic data generation tools and evaluation methods currently available are specific to the particular needs being addressed. Therefore, building a fully functional data obscuration tool that can handle a range of inputs, data obscuration requirements and synthetic dataset usages is not realistic at this time

Overall, it is clear from the work performed that the tools and metrics need to be tailored to meet the specific requirements of the sensitive information to be removed and the specific characteristics in the dataset to be preserved. It is therefore challenging to develop an all-encompassing tool that can be used for any dataset and any requirements, but developing tools that perform well on specific datasets and specific requirements is achievable. This has been both shown in the literature and by our assessment of the tools.

Contents

Executive Summary **1**

Contents **3**

1 Introduction **7**

1.1 Obscuration Requirements 9

1.2 Differential Privacy 11

2 Datasets **12**

2.1 Selected Datasets 12

3 Scoping Methods for Data Obscuration Methods **19**

3.1 Overview 19

3.2 Mimicking Methods 19

3.3 Redaction, Replacing/Masking, Coarsening and Simulation 23

4 Metric Development **26**

4.1 Evaluation Framework for Synthetic Data Generators 26

4.2 Evaluation Metrics for Synthetic Data 28

4.3 Conclusion 30

5 Tool Development and Testing **32**

5.1 DP-auto-GAN 33

5.2 Presidio 48

5.3 Synthetic Data Vault (SDV) 52

5.4 Conclusions 63

6 Scenario Examples **65**

6.1 Pattern of Life 65

6.2 Cloud computing 66

7 Conclusions and Recommendations **68**

7.1 Recommendations for Future Work 70

8 Abbreviations & Definitions 71

9 References 72

A Appendix A Datasets 76

B Appendix B Scoping Methods for Data Obscuration Methods 86

9.1 Overview 86

9.2 Mimicking Methods 86

9.3 Redaction, Replacing/Masking, Coarsening and Simulation 96

C Appendix C Metric Tools 102

D Appendix D DPautoGAN python packages 105

E Appendix D SDV (Financial Dataset) 108

Report Documentation Page v5.0 111

List of Figures

Figure 1 Summary of approach showing the components and data flow (coloured arrows) 7
Figure 2 Extract from an example CV 17
Figure 3 Screenshot of https://relational.fit.cvut.cz/ showing the top datasets available 18
Figure 4 Mostly GENERATE Overview 23
Figure 5 An evaluation framework for synthetic data generators. Figure adapted from [1] 27
Figure 6 Histogram of a synthetically generated feature with varying privacy budget. Figure adapted from [1] 30
Figure 7 Prediction scatterplot (ε = 1) 35
Figure 8 Crime in Los Angeles first run prediction scatter plot (ε = 1) 39
Figure 9 Crime in Los Angeles RUN-2 prediction scatter plot (ε = 1) 40
Figure 10 'Time Occurred' Histogram for training and synthetic data (RUN-2) 42
Figure 11 'Lat' and 'Lon' Histograms for training and synthetic data (RUN-2) 42
Figure 12 'Time Occurred' Histogram for training and synthetic data (RUN-4) 43
Figure 13 'Lat' and 'Lon' Histograms for training and synthetic data (RUN-4) 43
Figure 14 Heatmaps of training and synthetic location (RUN-4) 44
Figure 15 Histograms comparing 'Area' columns (RUN-4) 44
Figure 16 2D Histograms of 'Area' column correlation (Run-4) 45
Figure 17 Histograms comparing 'Crime' type columns (RUN-4) 45
Figure 18 Histograms comparing 'Victim' columns (RUN-4) 46
Figure 19 Histograms comparing 'Status' columns (RUN-4) 46
Figure 20 Crime in Los Angeles prediction scores for RUN-4, RUN-5 and RUN-6. 47
Figure 21 Example findings from Presidio web demo 50
Figure 22 The structure of the Czech financial dataset. 53

Figure 23 The SDV modelling process (top) and extended table (bottom) 54
Figure 24 Categorical data and their probabilities (left), and their representation as Gaussian distributions (right) 55
Figure 25 Table 'Account' with column 'account_id' 58
Figure 26 Table 'Account' with column 'date' 58
Figure 27 Table 'Account' with column 'district_id' 58
Figure 28 Table 'Disp' with column 'account_id' 58
Figure 29 Table 'Disp with column 'client_id' 58
Figure 30 Table 'Disp' with column 'disp_id' 58
Figure 31 Table 'Order' with column 'account_id' 59
Figure 32 Table 'Order' with column 'order_id' 59
Figure 33 Table 'Trans' with column 'account_id' 59
Figure 34 Table 'Trans' with column 'amount' 59
Figure 35 Table 'Trans' with column 'balance' 59
Figure 36 Table 'Trans' with column 'date' 59
Figure 37 Table 'Trans' with column 'trans_id' 60
Figure 38 Heat maps for the Account table for the real (left) and synthetic (right) data 61
Figure 39 Heat maps for the Disp table for the real (left) and synthetic (right) data 62
Figure 40 Heat maps for the Order table for the real (left) and synthetic (right) data 62
Figure 41 Heat maps for the Trans table for the real (left) and synthetic (right) data 63
Figure 42 Cloud computing overview 67
Figure 43 Original images from MNIST dataset (top) and the generated images (bottom) 86
Figure 44 Table shows examples of synthetic data produced for malaria patients. 87
Figure 45 Generated images (left column) and the three nearest neighbours from the real dataset for four different epsilons on the MNIST dataset 88
Figure 46 Table shows conditional samples of which the second one is from the IMDB dataset. 89
Figure 47 Example records from the original LACity dataset (left) and the synthesised table using tableGAN (right) using a low-privacy setting. 90
Figure 48 Synthetic samples produced for the MNIST dataset (top row) and CelebA (bottom row). Column (a) shows the results with the least privacy protection (epsilon=8), followed by Column (b) where epsilon=4, and then Column (c) which has the highest amount (epsilon=2). 92
Figure 49 Synthetic samples produced with G-PATE with low privacy protection (top row) and high privacy protection (bottom row) 93
Figure 50 Example input relational dataset consisting of 3 tables which can be used by the SDV model 95
Figure 51 Mostly GENERATE Overview 96
Figure 52 An example of the text redaction performed by Presidio 97
Figure 53 An example of Presidio using optical character recognition to mask out sensitive information in an image containing text. 98
Figure 54 Presidio will perform an analysis of the features it has redacted including the data type, the confidence score and the associated text. 98
Figure 55 OpenText Redact-It software showing examples of features which can be redacted (top) and an example of a PDF upon which redaction scripts have been run (right). 99
Figure 56 K-anonymity base privacy preserving system [58] 99

Figure 57 Example of the obfuscated point selection Randomize (a), N-Rand (b), N-Mix (c), Dispersion (d) and N-Dispersion (e) [59] 100
Figure 58 Summary of approach to generating privacy preserving traces [61] 101

List of Tables

Table 1 Data Obscuration requirements for LTI 10
Table 2 Summary of the information contained in the Chicago Crime dataset 13
Table 3 Summary of the information contained in the crime in Los Angeles dataset 14
Table 4 Summary of the information contained in the Chicago Taxi Rides dataset 15
Table 5 Summary of the information contained in the Amazon Fine Food Reviews dataset 16
Table 6 Example data obscuration techniques for sensor data 24
Table 7 Summary of Evaluation Metrics for Synthetic Data Generation 31
Table 8 Overview of the tools, datasets and metrics 33
Table 9 Prediction scores comparison 35
Table 10 Crime in Los Angeles data assessment 36
Table 11 Achieved Epsilon 46
Table 12 Comparison of synthetic performance 47
Table 13 Resume documents 49
Table 14 Presidio demo search filters 50
Table 15 Accuracy scores for 'LT CV 201608' 51
Table 16 Accuracy scores for '180517_Vasanthi Kasinathan' 51
Table 17 Accuracy scores for 'Resume --Rohini Prakash' 51
Table 18 Accuracy scores for 'CV-Gloria Cheng2018' 51
Table 19 Accuracy scores for 'eFinancialCareers_TT - CV' 52
Table 20 SDV data conversions that take place before being modelled 56
Table 21 Tool and data observation 64
Table 22 Guidance on using data obscuration techniques 69

1 Introduction

The aim of this project is to review state of the art techniques to create synthetic datasets that mimic the characteristics of a real dataset as closely as possible, but remove or obscure any private or sensitive information. The primary aim for this is to allow external parties to access and use the synthetic data and to have a high degree of confidence that work performed on the synthetic dataset will be transferable to the real dataset.

The work has been funded by the Logistics Technology Investigations (LTI) project at Dstl and has been focussed on datasets and scenarios applicable to LTI applications. Key applications areas of synthetic data for LTI are listed below. More detailed information on each is available in Section 1.1.

- To protect contents of data while increasing exploitation potential. For example, protecting personal, government and commercial data.
- To improve data quality and quantity. This could be by augmenting missing data, correcting erroneous data or generating more data where necessary.
- To exploit the full potential of cloud environments without compromising the sensitivity of information held by MOD (Ministry Of Defence)

The project consisted of an initial research phase and an experimentation phase. In the first, an in depth research into data obscuration methods, tools available for synthetic data generation and metrics for assessing the quality of synthetic data was conducted. The tools and evaluation metrics identified were then used to generate data and evaluate the accuracy and level of obscuration of the synthetic data when compared to the original real data. The Authority's real datasets are not available for this project and therefore open source datasets relevant to LTI were used as proxy real datasets. The diagram below shows a summary of the components and data flow.

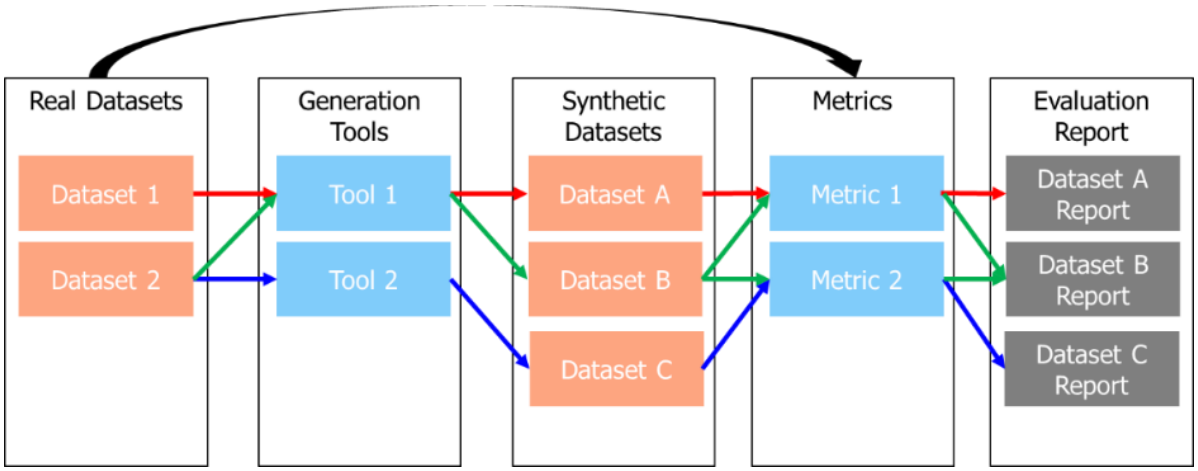


Figure 1 Summary of approach showing the components and data flow (coloured arrows)

The report is set out as follows. A summary of the types of data and obscuration requirements applicable to LTI is provided below. Section 1.2 describes the most relevant open source datasets available which are relevant to LTI and Appendix A provides a summary of all the datasets identified. Section 3 provides details of the techniques to create synthetic datasets that match the

characteristics of real data whilst removing or obscuring the sensitive data. Section 4 describes metrics used in the literature to assess how well the techniques have performed in matching the real dataset and removing sensitive information. Three tools from section 3 have been selected for further analysis; these are described in section 5, where the tools are assessed using a dataset from section 1.2 and selected metrics from section 4. Section 6 provides a discussion on two case studies provided by the Authority and section 7 provides the conclusion and recommendations.

1.1 Obscuration Requirements

Table 1 below provides a summary of the datasets and obscuration requirements for LTI.

Obscuration / Synthetic Data Generation Requirements	Type	Examples	Challenges	Example Techniques
To protect personal information MOD has a legal obligation for	Personal	Names, Addresses, etc. Date of Birth (DoB), Sex, Religion, etc. Medical information	Obscure the data so the individuals cannot be identified. One key challenge here is to consider how other publicly available data or leaked datasets could be used to locate records of individuals from the synthetic data	Names could be removed or replaced (e.g. masked, simulated) Addresses could be coarsened DoB, Sex, Religion etc. could be simulated or mimicked Medical information could be simulated or mimicked
To protect commercial sensitive information that could lead to financial damage or be exploited by a threat actor	Commercial	Intellectual property Costs Suppliers Manufacturing capability	Obscure the data to remove commercially sensitive information.	Remove any Intellectual property **Remove/replace any information that can be used to identify the commercial organisation and suppliers (e.g. company name, address)**
To protect strengths and weaknesses that a threat actor can exploit	Equipment	Operating performance Design characteristics Vulnerabilities Failure data Sensor data	Need to obscure the equipment identity and type and/or obscure the operating characteristics (e.g. performance, failures and weaknesses). One key challenge here is, as synthetic data generators are designed to retain statistical characteristics of original data, such information could be exploited to relate synthetic data with original equipment data	Replace or remove any information that can be used to identify the equipment Performance information (e.g. top speed) could be normalised

	Operation Security	Locations (past and future) Size of force Stockpiles	To protect operation personal and equipment. Ensuring current and future campaigns cannot be exploited	Locations can be coarsened or mimicked/simulated
	Supply Chain	Parts usage Node usage (flow) Lead times Inventory	Ensure strengths, weaknesses and characteristics of the supply chain cannot be exploited Need to obscure the part and/or the flow of the part through the supply chain	Replace or remove any information that can be used to identify the part Lead times and parts usage can be coarsened and/or normalised
	Infrastructure	Critical National Infrastructure Military Strategic base Deployed life support	Ensure strengths, weaknesses and characteristics of the infrastructure cannot be exploited	
To generate pattern of life data for monitoring purpose without compromising personal data	Personal/ Geographical	Daily ebbs and flows of people	Generate synthetic pattern of life data from real datasets to represents movements and trends people over time. Making such data relevant to individual geographical locations and making the data valid over time could be a challenge	Use of statistical or GAN based techniques to generate data Replace or remove any information that can be used to identify individuals or locations Locations can be coarsened or mimicked/simulated

Table 1 Data Obscuration requirements for LTI

1.2 Differential Privacy

Differential privacy [11] is an approach to providing data that represents the patterns from groups within the datasets but does not contain information relating to individuals. It is a term used throughout the report and by many of the papers and techniques described.

The differential privacy guarantee is determined using epsilon. The smaller the value of epsilon, the higher level of privacy is guaranteed. A value of 0 indicates maximum differential privacy protection whilst an infinite epsilon on the other hand would indicate there is no differential privacy. Generally there is a trade-off between preserving the accuracy of dataset and achieving an acceptable differential privacy guarantee.

In differential privacy, the randomness is introduced by adding a controlled amount of noise to the data. The challenge is that, with random noise, results from multiple datasets can be aggregated to reconstruct the original data by filtering out the noise through averaging. Systems can address this with a privacy "budget" - an absolute limit on the privacy loss that any individual or group is allowed to accrue. This could be achieved in synthetic data generation by limiting the number of synthetic datasets created from any original dataset - to reduce the possibility of merging datasets to gain insight into the original data.

The amount of acceptable distributional deviation depends on the use case and the application area. A number of academic publications have suggested a ε value as small as 0.001 to guarantee the masking of the private information. However, there is evidence [2] that such a small ε could result in significant loss in internal characteristics of the original data and may not deliver sufficient utility. Therefore it is essential that the most appropriate ε and the privacy budget is determined experimentally, based on data and application area and privacy requirements for each specific case. The ability to interact with and influence the data generation process will be key to this.

2 Datasets

This section provides details on the online datasets aligned to the types of real data used for LTI purposes. There are very few online datasets that are directly relevant to LTI and therefore the focus is on finding datasets that have similar characteristics, namely:

- A mixture of basic data types (numeric, date, categorical, textual, etc.)
- A range of situations (journeys, sensor logs etc)
- Relational datasets
- Small and large datasets

Appendix A provides details of all the datasets identified. A subset of the most appropriate datasets are described in more detail below. These datasets have been used for the tool development and testing work package. The long list of datasets listed in Appendix A were narrowed down to a shorter list; this is because many datasets did not fit specific requirements such as licensing restrictions, had been fully anonymised, meaning they were no longer representative of real datasets that contain sensitive data, or the dataset was too small to be useful. We believe the selected list below covers the widest range meeting LTI requirements.

2.1 Selected Datasets

2.1.1 Chicago Crime

https://www.kaggle.com/chicago/chicago-crime

This dataset provides details of reported crime incidents that occurred in the City of Chicago. The dataset is extracted daily from the Chicago Police Department's CLEAR (Citizen Law Enforcement Analysis and Reporting) system. It covers dates from 2001 to the present (not including the seven most recent days) and at present has approximately 7 million entries. Table 2 provides a summary of the fields of the dataset. This dataset is similar in nature to LTI maintenance data.

Type	Fields
ID	• ID • Case Number
Numeric	N/A
Date	• Date (when incident occurred) • Year (when the incident occurred) • updated_on
Categorical	• IUCR (Illinois Uniform Crime Reporting code) • Primary Type (primary description of the IUCR code e.g. THEFT, DECEPTIVE PRACTICE)

	• Description (secondary description of the IUCR code e.g. FORGERY, FINANCIAL ID THEFT: OVER $300) • Location Description (e.g. COMMERCIAL/BUSINESS OFFICE, RESIDENCE) • fbi_code (FBI crime classification e.g. 10)
Boolean	• Arrest • Domestic
Location	• Block (partially redacted address e.g. 082XX S COLES AVE) • Beat (smallest police geographic area e.g. 424) • District (police district e.g. 2) • Ward (City Council district e.g. 4) • Community_area (e.g. 36) • X_coordinate (State Plane Illinois East NAD 1983 projection) • Y_coordinate (State Plane Illinois East NAD 1983 projection) • Latitude • Longitude • Location (combined lat, lon)
Personal	N/A
Free Text	N/A

Table 2 Summary of the information contained in the Chicago Crime dataset

2.1.2 Crime in Los Angeles

https://www.kaggle.com/cityofLA/crime-in-los-angeles

This dataset provides details of crime in Los Angeles dating back to 2010 and is updated weekly. The dataset fields are summarised in Table 3. This dataset is similar in nature to LTI maintenance data.

Type	Fields
ID	• DR Number
Numeric	• Victim Age
Date	• Date Reported • Date Occurred • Time Occurred (e.g. 1800, 2300)
Categorical	• Crime Code

	• Crime Code Description (e.g. INTIMATE PARTNER - SIMPLE ASSAULT) • MO Codes (partially present e.g. 0416 0446 1243 2000, 0329) • Victim Sex (partially present. F or M) • Victim Descent (partially present, e.g. W, O) • Premise Code (e.g. 502.0, 101.0) • Premise Description (e.g. STREET, MULTI-UNIT DWELLING (APARTMENT, DUPLEX, ETC)) • Weapon Used Code (partially present) • Weapon Description (partially present, e.g. STRONG-ARM (HANDS, FIST, FEET OR BODILY FORCE)) • Status Code (e.g. AO) • Status Description (e.g. Adult Other) • Crime Code 1 • Crime Code 2 (partially present) • Crime Code 3 (mostly empty) • Crime Code 4 (mostly empty)
Boolean	N/A
Location	• Area ID • Area Name (e.g. 77th Street, Olympic) • Reporting District • Address (e.g. 6300 BRYNHURST AV) • Cross Street (partially present e.g. 15th, WALL) • Location (lat & Lon e.g. (33.9829, -118.3338))
Personal	N/A
Free Text	N/A

Table 3 Summary of the information contained in the crime in Los Angeles dataset

2.1.3 Chicago Taxi Rides

https://www.kaggle.com/chicago/chicago-taxi-rides-2016

This dataset provides taxi trip information for 2016 for those trips reported to the City of Chicago through its Department of Business Affairs & Consumer Protection (BACP). Some of the fields have been altered to protect privacy, for example, Census Tracts (geographic regions) are sometimes not

present and the times are rounded to the nearest 15 minutes. To reduce the size of the dataset some of the fields have been remapped to integers but a lookup table is provided to convert the value back to the original value. This data is similar in nature to vehicle sensor LTI data. The dataset fields are summarised in Table 4.

Note, although the data is provided publicly, the indemnity is onerous, which may make it difficult to use.

Type	Fields
ID	• Taxi_id (remapped) • company (partially present. e.g. 107)
Numeric	• Trip_seconds (to the nearest minute. E.g. 180, 720) • Trip miles (e.g. 0.4, 0.7) • Fare (e.g. 4.50, 42.75) • Tips (e.g. 0.00, 4.45) • Tolls (e.g. 0.00,) • Extras (e.g. 0.00, 1.50) • Trip_total (sum of Fare, Tips, Tolls and Extras)
Date	• trip_start_timestamp (to nearest 15 mins) • trip_end_timestamp (to nearest 15 mins)
Categorical	• payment_type (e.g. Cash, Credit Card)
Boolean	N/A
Location	• pickup_census_tract (remapped - always empty) • dropoff_census_tract (remapped - partially present) • pickup_community_area (partially present) • dropoff_community_area (partially present) • pickup_latitude (remapped) • pickup_longitude (remapped) • dropoff_latitude (remapped) • dropoff_longitude (remapped)
Personal	N/A
Free Text	N/A

Table 4 Summary of the information contained in the Chicago Taxi Rides dataset

2.1.4 Amazon Fine Food Reviews

https://www.kaggle.com/snap/amazon-fine-food-reviews

This dataset contains ~500,000 fine food reviews from Amazon over a 10 year period. Each review contains the product and reviewer information as well as the review text and score. The text is generally short to medium in length (i.e. one or two sentences up to a couple of paragraphs). The dataset fields are summarised in Table 5. This dataset is similar in nature to LTI forms containing short free text fields.

Type	Fields
ID	• Id • ProductId (B001E4KFG0) • UserId (A3SGXH7AUHU8GW)
Numeric	• HelpfulnessNumerator (Number of users who found the review helpful) • HelpfulnessDenominator (Number of users who indicated whether they found the review helpful or not)
Date	• Time (of the review e.g. 1303862400)
Categorical	• Score (Rating between 1 and 5)
Boolean	N/A
Location	N/A
Personal	• ProfileName (e.g. delmartian, dll pa, Michael D. Bigham "M. Wassir")
Free Text	• Summary (e.g. Good Quality Dog Food) • Text (e.g. I have bought several of the Vitality canned dog food products and have found them all to be of good quality. The product looks more like a stew than a processed meat and it smells better. My Labrador is finicky and she appreciates this product better than most.)

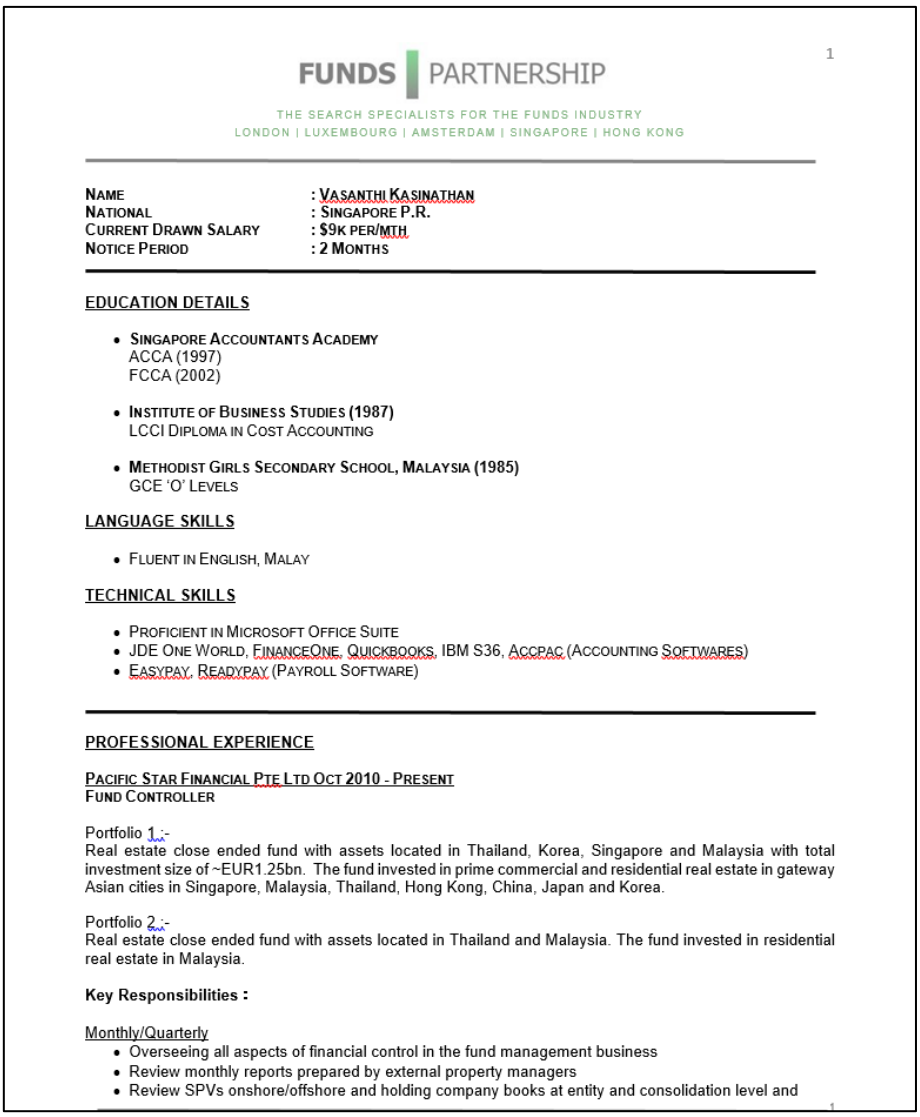
Table 5 Summary of the information contained in the Amazon Fine Food Reviews dataset

2.1.5 Resumes Dataset

https://github.com/JAIJANYANI/Automated-Resume-Screening-System

A small set of 21 resumes in PDF or MS word format (in the Original_Resumes folder on the GitHub page). There is no consistency in the information provided or the formatting of data. The resumes do contain personal information, so provide a good dataset to test privacy tools. A sample is shown

below in Figure 2. A larger set of resumes is available to download from the links provided on the GitHub page. The resumes are similar to LTI export licences.



FUNDS | PARTNERSHIP

THE SEARCH SPECIALISTS FOR THE FUNDS INDUSTRY
LONDON | LUXEMBOURG | AMSTERDAM | SINGAPORE | HONG KONG

NAME : VASANTHI KASINATHAN
NATIONAL : SINGAPORE P.R.
CURRENT DRAWN SALARY : $9K PER/MTH
NOTICE PERIOD : 2 MONTHS

EDUCATION DETAILS

- **SINGAPORE ACCOUNTANTS ACADEMY**
  ACCA (1997)
  FCCA (2002)
- **INSTITUTE OF BUSINESS STUDIES (1987)**
  LCCI DIPLOMA IN COST ACCOUNTING
- **METHODIST GIRLS SECONDARY SCHOOL, MALAYSIA (1985)**
  GCE 'O' LEVELS

LANGUAGE SKILLS

- FLUENT IN ENGLISH, MALAY

TECHNICAL SKILLS

- PROFICIENT IN MICROSOFT OFFICE SUITE
- JDE ONE WORLD, FINANCEONE, QUICKBOOKS, IBM S36, ACCPAC (ACCOUNTING SOFTWARES)
- EASYPAY, READYPAY (PAYROLL SOFTWARE)

PROFESSIONAL EXPERIENCE

PACIFIC STAR FINANCIAL PTE LTD OCT 2010 - PRESENT
FUND CONTROLLER

Portfolio 1 :-
Real estate close ended fund with assets located in Thailand, Korea, Singapore and Malaysia with total investment size of ~EUR1.25bn. The fund invested in prime commercial and residential real estate in gateway Asian cities in Singapore, Malaysia, Thailand, Hong Kong, China, Japan and Korea.

Portfolio 2 :-
Real estate close ended fund with assets located in Thailand and Malaysia. The fund invested in residential real estate in Malaysia.

Key Responsibilities :

Monthly/Quarterly
- Overseeing all aspects of financial control in the fund management business
- Review monthly reports prepared by external property managers
- Review SPVs onshore/offshore and holding company books at entity and consolidation level and

Figure 2 Extract from an example CV

2.1.6 Relational Dataset Repository

https://relational.fit.cvut.cz/

The Relational Dataset Repository is a large collection of relational datasets. The website provides a way to search and view details of the datasets. The datasets are typically stored on a MariaDB database and a MariaDB client can be used to export the datasets (e.g. CSV or SQL dump). No license or usage information is provided on the website, but the owner was contacted and gave permission for us to use the dataset. A screenshot of the Relational Dataset Repository website showing the top datasets is shown in Figure 3. The datasets in this website, such as the Financial dataset, are similar to LTI datasets containing relational tables (e.g. maintenance datasets).

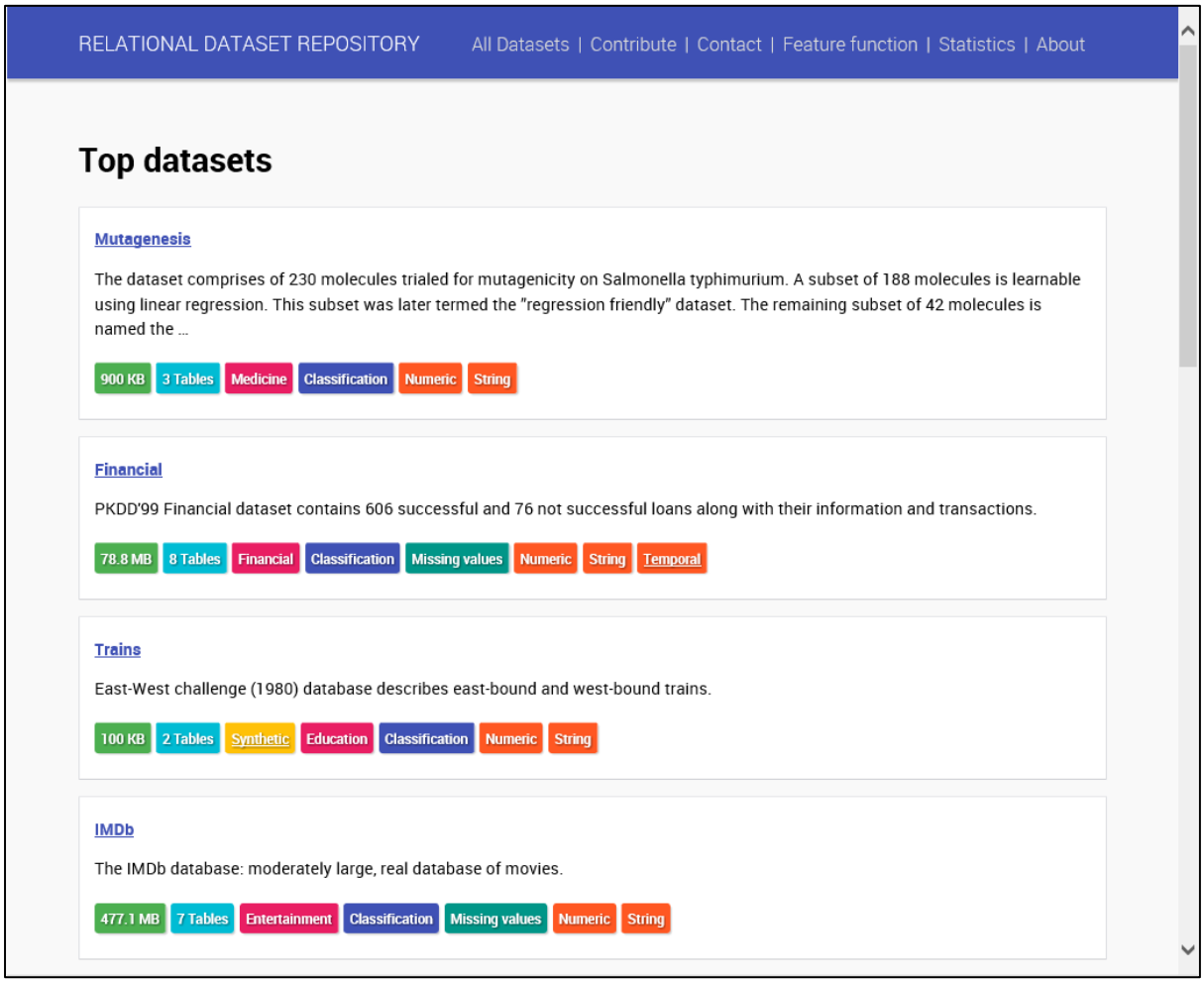


Figure 3 Screenshot of https://relational.fit.cvut.cz/ showing the top datasets available

3 Scoping Methods for Data Obscuration Methods

3.1 Overview

There are a wide range of data obscuration techniques, but typically they can be summarised as falling into the following categories:

- Redaction – completely removing data from the dataset
- Replacing/Masking – replacing parts of the dataset, e.g. using: hashing, encryption, tokenising or lookup datasets
- Coarsening – reducing the precision of the data, e.g. reducing number of decimal places in lat/lon, remove last 3 digits of postcode
- Mimicking – generate a dataset that closely matches the real dataset but does not contain exactly the same entries
- Simulation – generating part or all of the dataset that is similar in essential ways to the real data but is different with regard to sensitive information.

The actual techniques to perform these obscurations will vary depending on the type of dataset.

Synthetic data generation, by its very nature, may find common trends in the real dataset but are often unable to capture any anomalies present. However, this may not be a critical issue depending on the task. In addition, it is arguably not possible to have a universal synthetic generation tool; based on our findings, the tools developed often have to be tuned to their application and are therefore suited only towards specific datasets.

The first section focuses on mimicking techniques, as these are one of the most active areas of current research in generating synthetic data. Mimicking is the process of generating data that is as close as possible to the real dataset but cannot be used to recover the original data. To ensure this, differential privacy is a mechanism often considered when these models are produced [11] .

The primary mimicking methods which have been identified in producing synthetic data can be grouped into the following categories: variational autoencoders (VAEs), generative adversarial networks (GANs) and statistical tools [13] .

Following the mimicking approach, techniques for dealing with specific types of data are considered below, specifically addressing sensor/performance data, text-based methods and location-based methods. Appendix B contains further details on the papers which were explored in the below sections.

3.2 Mimicking Methods

3.2.1 Variational Autoencoders (VAEs)

A VAE is an example of a deep learning-based generative model which can be used to produce synthetic data by learning underlying correlations. They have become one of the most popular unsupervised methods to learn complicated distributions [14] As a result, they are widely used in

fields such as image generation and sentence interpolation . These generative models aim to learn by attempting to recreate a given input under constraints such as lower dimensional hidden layers and regularisation processes and are especially useful for imbalanced datasets. VAEs are often a quick way of getting data with a reasonable level of accuracy [15] .

Here, two papers were explored: the first explored the use of VAEs on an imbalanced image dataset [15] and the second explored it on synthetic data generation of patient records [17]. Even though the image datasets are not directly related to LTI requirements, it was considered necessary to examine the methods they used for completeness as this is still an emerging field. In this, the VAE was used to generate images of digits with different handwriting that was not present in the original MNIST dataset. It was found that the digits produced were clear and sharp, unlike other autoencoder methods but no attempt was made to evaluate if they were unique.

The second paper [17] focussed on tabular data containing numerical and categorical data (e.g. age, gender, symptoms etc) for a series of patient visits in a hospital. The aim here was to see if the synthetic patient records were convincing enough for doctors. The authors stated that they explored Gaussian Mixture Models (GMMs) and adversarial networks but decided to use a VAE due to its simplicity and performance in identifying relationships in large amounts of unlabelled data.

The VAE used required a short training time and a subjective metric was used where the authors mixed the real patient data with the synthetic data and asked doctors whether they could identify whether which records were real and which were synthetic. They found that 20% of synthetic records were identified as synthetic, 23.3% of real were identified as synthetic and 80% of the synthetic records were identified as real, Hence, the technique they used was deemed successful.

The advantages of using a VAE for synthetic record generation include their simplicity in setting up and in training (compared to GANs) and they are also effective in obtaining reasonably accurate synthetic data. However, the most significant downside is that there is often no differential privacy-based focus and so there can be no guarantee that the network was not just memorising the real samples. Because of this, if differential privacy is vital for the problem at hand, a VAE could be used to generate more training data to be used for a more sophisticated tool such as a GAN.

3.2.2 GAN Methods

In recent years, GANs have become a very active area of research because they have been shown to be effective in preserving the privacy of the original dataset, unlike VAEs, by using a differential privacy guarantee.

GANS are often more difficult to train than VAE and can suffer from instability during training which leads to highly inaccurate outputs [16] . In this section, various architectures were explored including DPGAN, MaskGAN, TableGAN, AC-GAN, GANobfuscator, G-PATE, DP-auto-GAN plus use cases with binary classes and for medical time-series generation. Overall, it seems GANs have been shown to have mixed results when it comes to producing accurate data with a differential privacy guarantee. Traditional GANs often learn distributions of training data points and can end up memorising the distributions of training samples which is another reason noise is added into gradients during training.

As mentioned before, several architectures explored focussed on image-based datasets. However, the underlying approaches explored, for example in the use of differential privacy, are relevant to other datasets and therefore were kept for completeness. The first example looked at was DPGAN [18] where images of digits were produced, like with the autoencoder example in the previous section, with different values of epsilon. The aim was to produce digits but of different handwriting to those found anywhere in the MNIST dataset and the results indicated that the generated images produced a good balance between accuracy and privacy even at low epsilon values. A drawback of DPGAN is that it is not efficient (compared to other privacy-focussed GANs) which can limit training stability and convergence speed. It also encounters significant utility loss on synthetic data when a large amount is produced (or when epsilon is reduced by too much).

The G-PATE GAN [36] also used the MNIST dataset as well as a fully numerical tabular Credit Card Fraud Detection dataset and found it was able to minimise utility loss and improve privacy budget compared to similar GANs such as vanilla GAN, DPGAN and PATE-GAN. The resulting images generally looked quite poor but they preserved partial features from the real images which could be useful but they did not provide any further analysis or evidence to prove this.

GANobfuscator [35] looked at images of celebrities, digits and scenery using a similar method to DPGAN. However, the results here showed that there was a stronger privacy guarantee with the different method they used in injecting noise into the training procedure. The technique used was found to be stable during training and did not suffer from mode collapse or gradient vanishing. These are often common issues with GANs where the model stops learning and only limited varieties of data are produced. This model was also superior at creating large amounts of quality data compared to DPGAN. GANobfuscator was found to more effective at membership attack, which is a procedure in which the original training data is determined. However, the output synthetic images still sometimes had unrealistic details, but this also meant that it is unlikely any of the images had been memorised.

MaskGAN [21] dealt with filling in missing text into a sentence containing gaps, depending on the surrounding context. GANs have been used extensively for images but not as much for text generation due to difficulty with instabilities during training; hence, the authors used reinforcement learning to train the generator. The results were compared to those generated with a more traditional maximum-likelihood model (MaskMLE [18]). Reviews from the movie database IMDB were used and assessors were asked which extract was of higher quality (between two choices, randomly chosen between MaskMLE, MaskGAN or the real IMDB reviews). Results showed that people preferred the real outputs, followed by those of MaskGAN, and then MaskMLE.

TableGAN [24] is an implementation that works with comma-separated value (CSV) data and uses a widely adopted privacy model known as k-anonymity. The authors chose this model due to its flexibility in modelling distributions that would also be protected from information leakage. Four datasets which contained tables with numerical and categorical data were investigated and the proposed method was found to be effective against the three major types of privacy-based attack: membership attack, re-identification attack (where the obfuscated record is correctly linked with a person from the real dataset) and attribute disclosure (where information is inferred depending on what values are shared between points).

The authors claimed that this was the first method to use deep learning when it came to relational databases, which are tables that have defined relationships between them. TableGAN only works with numerical and categorical data but the authors expressed interest in exploring other data types such as strings. However, despite its success in privacy guarantees, it suffers from common GAN issues such as mode collapse. Recently, another model has been produced for tabular data which builds upon tableGAN known as Conditional Tabular GAN (CTGAN) is similar, but is successful when there are many categories of data.

AC-GAN [28] for SPRINT trial was a GAN developed to simulate participants in a clinical trial. It is similar to the aforementioned patient records VAE but with added privacy guarantees. The table data in this case contained numerical data only for systolic/diastolic blood pressure and number of medications provided in a given visit for a patient. The results were evaluated by generating Perason coefficients between real and synthetic data which resulted in a value of 0.89 indicating high correlation in their distributions. Clinicians looked at any inconsistencies (e.g. blood pressure in normal region but medication prescribed) and gave a 'realism' score between 0 (not realistic) and 10 (very realistic) for a record. The mean score for the real data was 5.16 and was 5.01 for synthetic suggesting it was difficult to distinguish between real and the synthetic data.

In addition, a recurrent conditional GAN [40] was used to mimic time-series data and focused on generating regularly sampled data measured by monitors, such as the heart rate, with differential privacy. A random forest classifier was trained on the real data, and another was trained on the synthetic data and results showed that there was a small to moderate drop in performance with the synthetic data compared to real data.

Another paper [30] explored how four GANs – vanilla (normal) GAN, Wasserstein GAN (WGAN), Conditional GAN (CGAN), Wasserstein Conditional GAN (WCGAN) performed in generating a synthetic dataset (table) that matched the characteristics of US census data. The assessment was performed by training a classifier to predict the final column of the table (salary of either >$50k or <$50k) based on the preceding numerical columns (for age, marital status, race, hours worked per week etc). The results were quite similar between them and an average of 80% was obtained in predicting the correct binary class of the final column. Nevertheless, the authors believed WGAN produced the optimum performance as it did not suffer from common GAN issues; the authors are now exploring the possibility of using more than two classes.

DP-auto-GAN is a very recent model that can work with binary, categorical and numerical data. It had been used on tabular data (UCI ADULT Census) and showed significantly better performance compared to prior work on the same dataset. It also demonstrated the ability to produce superior results with a high level of privacy protection compared to prior work, which were not able to produce comparable accuracies at this same level. This tool was explored further and is described in detail in Section 5.

3.2.3 Statistical & Other Deep Learning Methods

Alongside VAEs and GANs, statistical-based tools have also been popular in generating synthetic data. One example is the Python package Synthetic Data Vault (SDV) [41] that can be used to build generative models of relational databases. In other words, it allows users to model an entire multi-table relational dataset using a statistical model that can then be used to model numerical,

categorical and date-time data. It is currently limited to a Gaussian distribution and was tested on five datasets containing these types of sata. Freelance data scientists were told to write features scripts to assess the accuracy of the predictions and found no significant difference between real and synthetic dataset in terms of accuracy but there were issues such as unrealistic values for ages. No rigorous privacy guarantee proof was provided and the tool is explored in greater depth in Section 5.

An R package called SynthPop [43] uses hidden Markov models and regression to model the data and works with CSVs containing numerical, categorical and Boolean data. The paper mentions that it can work with confidential data but no rigorous differential privacy proof was provided here either. Another tool called SynSys [45] was used using similar processes for time-series data containing date-time, categorical and Boolean data. Here, the aim was to produce synthetic data that looked similar to that collected by a smart home device usage. A classifier, which aimed to classify between activities, was trained and having the combined synthetic/real data produced a 10% improvement over using just the real data. SynSys focussed on producing accurate synthetic data, as with the VAEs, and had no privacy guarantee

There were some commercial tools found too such as Mostly GENERATE [47] . This tool has a free demo which uses generate deep neural networks to produce up to 500 rows and 50 columns of synthetic data, with a privacy guarantee (an overview is shown in Figure 4). Nevertheless, there was very little information available regarding the methods used.

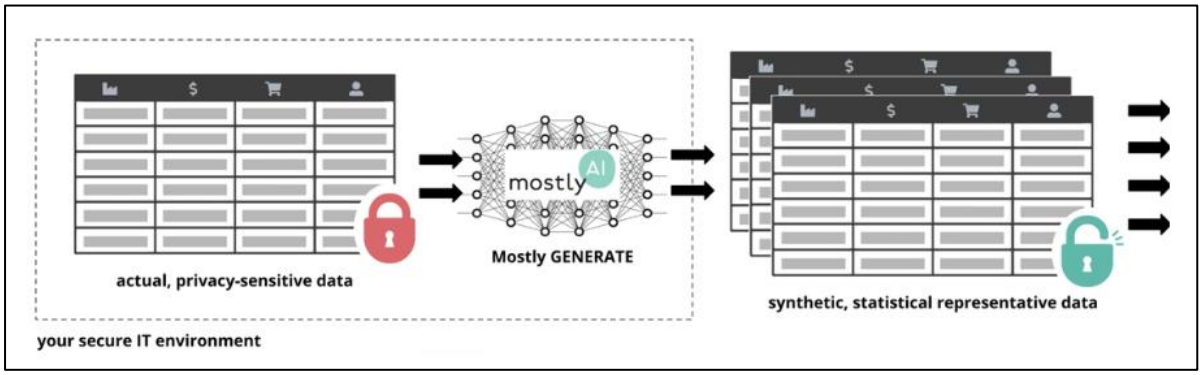


Figure 4 Mostly GENERATE Overview

3.3 Redaction, Replacing/Masking, Coarsening and Simulation

As discussed in obscuration requirements, (section 1.1) datasets may contain a range of information that needs to be obscured, such as information about performance of equipment in certain conditions. For the dataset to be useful it needs to contain correct and meaningful values, but this is in conflict with removing information that can be exploited. Potential options obscure the data are:

- Redaction – remove the data, either completely or specific data points
- Anonymise the field name
- Normalise the data – preserves the characteristics, but the data can be reversed engineered if enough real data points are known

- Coarsening - Coarsen the data, e.g. into ranges – help preserve privacy, but impacts the usefulness of the data

Table 6 below shows a simple example of these techniques applied to a sensor that measures the speed of a vehicle. It should be noted that there are still risks a threat actor might reverse engineer the obscured data to obtain the original real values if they have access to the expected distributions of real values through prior knowledge or from observations, and they have enough obscured data. The more fields that are obscured in the dataset the harder this becomes.

Speed	A1 (norm)	A2 (coarsen)	A3 (norm & coarsen)
110	1.00	91-120	0.76-1.0
50	0.45	31-60	0.26-0.50
80	0.73	61-90	0.51-0.75
20	0.18	1-30	0-0.25
100	0.91	91-120	0.76-1.0
85	0.77	61-90	0.76-1.0
25	0.23	1-30	0-0.25

Table 6 Example data obscuration techniques for sensor data

There is an active area of research looking at protecting user's geolocation data when using location based services (LBS) real time, for example when using services that use the user's mobile GPS as described in [59] . These techniques are grouped into three categories: anonymisation, obfuscation and encryption. Whilst the papers are specific to geolocation data a lot of the base techniques could be used for other purposes as well.

Some techniques have explored how sensitive text can be redacted from documents. One method is through using commercial tools such as Rosette Text [49] . These tools can mask identifiable information using regex-based techniques. A 30 day free trial is offered for the former and could be used in conjunction with the tool Baleen [52] (DSTL) in order to extract text from unstructured or semi-structured text. The SciBite Termite tool [50] uses named entity recognition to find specific information and can be tuned to look for specific details and can output the redacted text in HTML, Word, XML and JSON formats.

Bitcurator Redact [53] is a java-based PDF redaction tool which employs statistical named entity recognition. It is a GUI-based application that highlights Personally Identifiable Information (PIIs) and asks if you want to remove it or not. If so, it will replace the PDF text areas with empty space with a black border. The OpenText Redact-It [55] is a regex-based commercial tool which works with PDFs too, but also Word documents and scanned images. It covers up sensitive text with a filled black box and allows you to add in custom regex expressions to look for other PIIs.

The final tool, which is the most advanced out of these options, is Presidio API [54] . This open-source tool has been pre-trained using machine learning and can remove PIIs such as credit card numbers, names, locations, US phone numbers etc. It works with structured documents where it will identify PIIs and replace it with a placeholder (e.g. "NAME" or "LOCATION"). It is also able to give confidence scores on each bit of text removed which can be useful for analysis. For unstructured text such as images, it can use optical character recognition (OCR) to cover up sensitive bits of text with a

black box. The tool is also highly customisable and can be programmed to look for other features. Hence, this tool was explored further in Section 5.

Simulation techniques are often based on modelling the behaviours of entities/agents of interest (e.g. vehicles or people) and then using these behaviours, along with a model of the environment, to generate synthetic datasets. Simulation base data synthesis can create completely new data (i.e. no privacy concerns) but there are challenges in defining the behaviours, especially in automatically calculating the behaviours from real datasets. Typically, a human will define the behaviours based on information extracted from the data and at a granularity required to generate the required datasets, though common core blocks such as agent based modelling can used to structure and perform the simulation.

4 Metric Development

As discussed in detail under section 3, many tools and techniques have been developed over the years to generate synthetic data. These have been tested using a data sources of varying complexity. Although the performance of such systems have always been subject to some form of evaluation, our investigation into the academic literature has shown that the performance measures and evaluation metrics employed by those tools and techniques vary from application to application, and there does not seem to be a unified approach. This makes comparison of such tools against each other challenging. Therefore, the purpose of this section is to explore:

- different types of assessment metrics available to compare and contrast performance of different synthetic data generation tools,
- the ability of such tools to obscure sensitive information,
- and the quality of the outputs produced by such tools.

We first define an evaluation framework for assessing various synthetic data generation tools. The focus here is to establish an overall approach that needs to be followed when evaluating such tools. We then discuss metrics available to assess the quality of synthetically generated data. Conclusions of this work can be used as a guide when selecting and evaluating a data generation tool for LTI applications.

4.1 Evaluation Framework for Synthetic Data Generators

An overview of the proposed evaluation framework is shown in Figure 5. The key elements of the evaluation framework includes assessing system's ability to: handle different data types and datasets of varying dimensionality and length; retain internal characteristics of original data; mask sensitive information and options to influence the balance between privacy and utility trade-off; understand or interpret the decision making process; ability to adapt or reconfigure the tools for new data or patterns and performance of the system both in terms of algorithmic complexity and time and resource requirements.

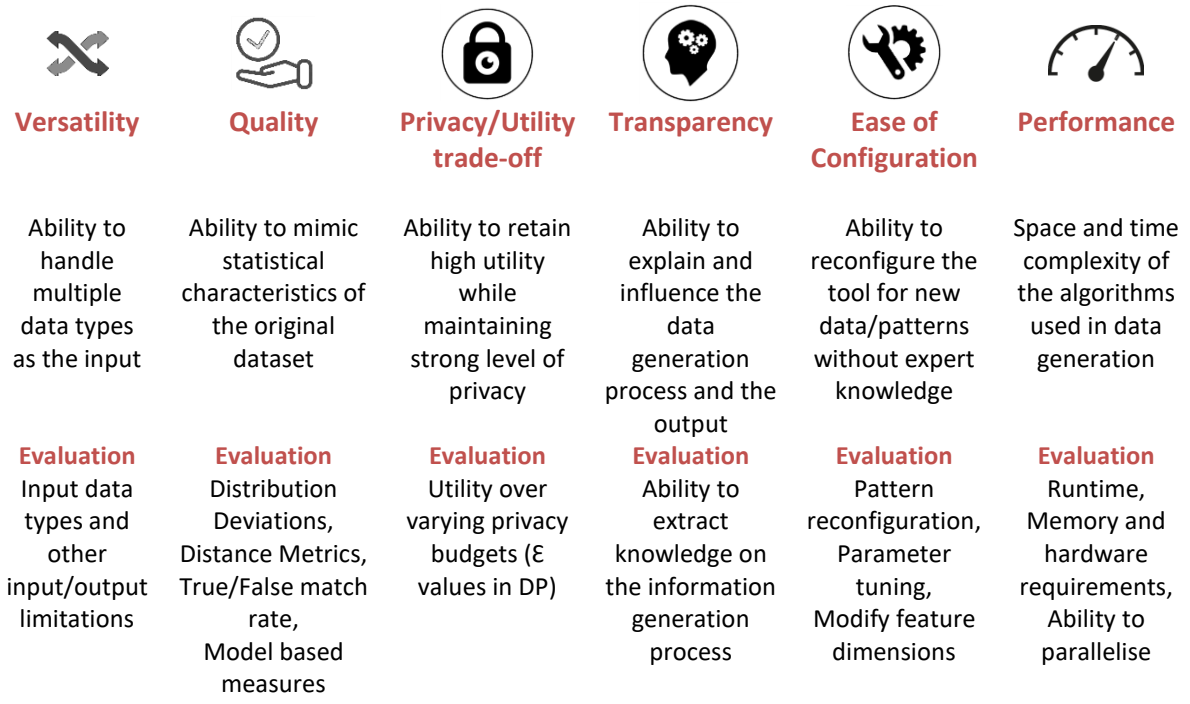


Figure 5 An evaluation framework for synthetic data generators. Figure adapted from [1]

Versatility: Large complex datasets can come in varying lengths, dimensionality and have a range of different data types: categorical – ordinal or non-ordinal, numerical, binary, time-series and spatial etc. However, many tools currently available come with various limitations including the number of data types each can handle. The ability of a data generation tool to handle a number of different data types is an important feature as it allows easier management of data generation and evaluation workflows. Where this is not possible, different data types need to be handled through different tools, which could make retaining inter-feature relationships more challenging.

Data quality: Data quality is an indicator of system's ability to produce synthetic data that retains the characteristics of original data - algorithm's ability to adequately capture the statistical characteristics and patterns of original data and reproduce new data with similar characteristics. This can be done by exploring general and specific measures of utility.

General utility measures are explored by examining distribution deviations and distance matrices between original and synthetic data. Specific utility measures compare the difference between results from a particular type of analysis conducted on the data. These could include comparing data summaries or coefficients of models fitted to both original and synthetic data. However, for such methods to be successful, data generators needs to have a good understanding of the type of analytical task to be performed on the synthetic data. Without such insights (e.g. general data exploration task) data generated or assessed through one analytical technique may not be relevant to the desired analytical task. As an example, various machine learning tasks (e.g. classification, clustering and regression) performed on real and synthetic data should lead to comparable results if the performance matrices are appropriate and relevant to the datasets in use. If the inferences drawn from both the original and synthetic data are comparable, then the synthetic data have high utility. A more detailed description of data quality measures used by various tools and algorithms are discussed in section 4.2.

Privacy/Utility Trade-off: Privacy mechanisms used to mask sensitive information introduce a proportion of randomness into the dataset, and as a result, the synthetic data distributions deviates from the original distribution. The amount of randomness introduced is a trade-off and increased level of randomness makes the data more anonymous, but less useful. Therefore it is imperative to not only evaluate the privacy guarantee, but also examine the balancing effect it has on the utility and functionality of the data.

Transparency: One other important aspect of synthetic data generation is the ability to control or influence the data generation process. Transparency (i.e. ability interpret the outputs and also understand the internal workings of the model) plays an important role here. For example, when generating a dataset for a classification task, if the degree of separation between classes needs to be controlled, that cannot be achieved if the internal workings of the model are unknown. Unfortunately, most of the advanced data generation methods (e.g. GAN (Generative Adversarial Networks) methods) are black box methods and provide very limited information on the internal data generation process. Therefore, users do not have much control or influence over the synthetic data generated though such methods.

Ease of Configuration: Ability to adapt a particular tool to varying needs – data types, dataset sizes, feature dimensions etc. – is an essential factor if a tool to be used for generating synthetic data from different datasets. How easy a particular tool is to be adapted for a given application, and the level of technical expertise and involvement necessary, will need to be considered before adopting any such tools for practical applications beyond academia.

Performance: Although the synthetic data generation is not a real-time issue, it is still important to understand the computationally efficiency of the algorithms so that the hardware requirements and other limitations of the models are understood. How well each model scales with increase in dataset length, feature dimension and the type of data fields are some of the factors to consider. Computational time, hardware and memory requirements will help establishing most suitable network architecture to perform synthetic data generation tasks.

4.2 Evaluation Metrics for Synthetic Data

As discussed previously, synthetic data generation mechanisms are always subject to some form of evaluation. Such evaluation methods focus on two areas: assessing the utility of the dataset and also the risk of disclosure.

Disclosure risk is an essential factor to consider when dealing with partially synthetic data. That is when only the sensitive information of the data is masked or synthetically generated while retaining the remainder of the original dataset. In such scenarios, where an intruder already possesses a portion of the actual dataset and can attempt to use that to find the values of sensitive records – locate individual entries based on another identifying dataset (or by merging a number of datasets together).

In such situations, disclosure risk could be measured used a methodology based on True/False match rate as discussed in [7] . Here; let **t** be the list of records the intruder is interested in and **R** be the records in **t** for which only one record in the dataset is matched with highest probability. **R** can be

further split into two subsets: **T** - records in **R** with true matches and **F** - records in **R** with false matches. Then true match rate and false match rate can be defined as:

True Match Rate (True MR) = |T| / |t|

False Match Rate (False MR) = |F| / |R|

When it comes to evaluating the relevance or the utility of synthetic data, a number of approaches are possible:

- Use of statistical measures – e.g. statistical distribution deviations and distance measures
- ML (Machine Learning) based techniques – e.g. comparison of performance on classification/regression/clustering etc. tasks performed on real and synthetic data
- Qualitative methods – e.g. histogram based visualisation methods, scatter plots

Although there is no unified approach in evaluating the quality of synthetic data, almost all the methodologies available can be divided broadly into two categories: supervised and unsupervised. Supervised techniques are used when dealing with labelled data. Unsupervised technique are applicable when a dataset does not contain ground truth labels. These metrics can further be divided into three categories: qualitative (visualisation based), distributional distance based and prediction based. Choice of the evaluation metric is dependent on the nature of the dataset (labelled/unlabelled), data types and other specific requirements attached to the type of analytical task to be performed on synthetic data.

4.2.1 Supervised Evaluation Metrics

Supervised learning metrics can be used when dealing with labelled data (i.e. datasets that have labels attached to each data entry). Here the focus will be to generate synthetic data that retains the relationship between the data points and labels in the original data. One way to test this is to train a machine learning model on the synthetic data and evaluate its performance on the original dataset [4] Jordon et al. [5] extends this approach by training and evaluating synthetic data using a range of different models. The aim here is to evaluate how the performance of a range of machine learning models trained on original data is preserved when the same models are trained on the synthetic data.

Another approach possible with labelled datasets is to use a feature importance algorithm to score features. Comparable propensity scores (e.g. Random forest feature importance) on both the original and synthetic datasets can be used to determine how representative the features of synthetic dataset is of the original data. The obvious disadvantage of the supervised approach is that the dataset needs to have features that can be termed decisively as a label.

4.2.2 Unsupervised Evaluation Metrics

In an unsupervised learning setting, there are a number of different methods to evaluate the quality of the synthetically generated data. These include feature (dimension)-wise prediction, distributional distance and visualisation methods.

Feature-wise prediction: One such approach is to generalise the evaluation method used in the supervised learning case where, each feature of the dataset is predicted using the remaining features. Predicting each feature from the rest of the features captures both the statistical characteristics of each feature and also the correlation between them. This process needs to be conducted on both the original and synthetic datasets. Similar feature-wise prediction scores on both can demonstrate that the synthetic dataset has captured inter-feature relationships in the real data [6]

Another most common way of evaluating a synthetic dataset for utility is to use a distributional distance measure such as normalised KL Divergence [7] confidence interval overlap [8] or the Wasserstein metric [9] This could be done at two different levels. Firstly, it can be done at each attribute level where each statistical distribution of each of the feature in the synthetic dataset is evaluated against the relevant feature in the original dataset. The mean distance score calculated on one dataset can then be used to compare with different datasets. Similarly, in [10] a 3-way marginal density distribution between the original and synthetic datasets is considered. Here, 3 features of the real and synthetic datasets are chosen at random to compute the total variational distance between the datasets. The process is repeated a number of times to return an average score. A similar method is proposed in [9] in which, instead of 3 random features, the performance metric is based on k-way (k features) PCA (Principal Component Analysis).

The other common approach is to use qualitative (visualisation) measures to evaluate synthetic data. Such methods are particularly useful when a feature is highly skewed or sparse and hard to predict well with models. Histogram based techniques (see Figure 6), cross correlation metrics and heat map based correlation assessment technique methods and 2-way PCA marginal plots are some of the qualitative techniques that could be employed.

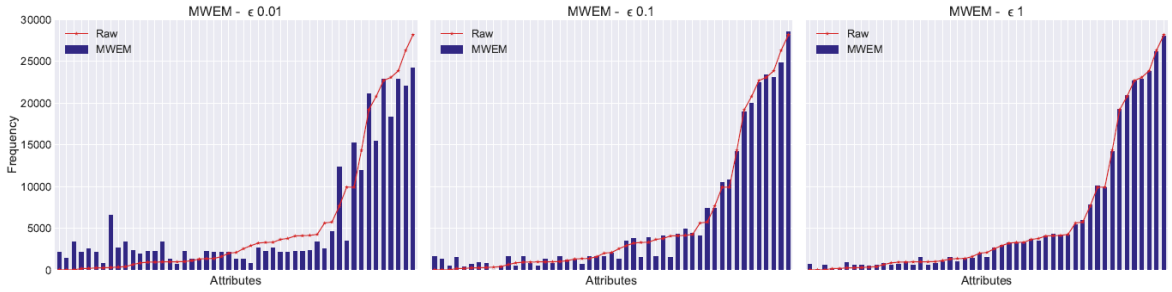


Figure 6 Histogram of a synthetically generated feature with varying privacy budget. Figure adapted from [1]

4.3 Conclusion

Overall, the type of evaluation metric to use depends on the nature of the dataset (supervised/ unsupervised) and categories of data types presented within a dataset. A summary of different options available is presented in the table below. It is advisable to employ a combination of different evaluation methods when assessing a data generation tool and its output. For example, the methods such as histogram techniques can be used to visualise the statistical distribution of each data field in the original data as well as to compare data fields between original and synthetic data. Similarly, the cross correlation metrics can be used to visualise and compare relationships between the data fields in the original and synthetic datasets. Such visualisation methods combined with distance-based

methods (e.g. distributional distance) and machine learning based performance measure can provide a good overall method of synthetic data. The Table 7 below provides a summary of the recommended evaluation techniques to use when assessing the quality of synthetic data.

Dataset Type	Evaluation Method to Use	Evaluation Metrics	Data Types
Labelled dataset	Supervised Machine Learning based techniques (Classification/Regression) and Unsupervised Machine Learning based techniques	F-measures Receiver Operating Characteristic (ROC) Area Under the Curve (AUC) Root Mean Square Error (RMSE) Gini Index Predictive model ranking [5] Feature Importance Scoring [12] Dimension-wise prediction-based measures [6]	Binary Numerical Categorical
	Histogram based techniques	Visualisation techniques to assess and compare statistical distributions	Binary Numerical Categorical
	Heat maps / Correlation metrics	Visualisation techniques to assess correlation between features (within a dataset as well as between original and synthetic data)	Numerical
Unlabelled dataset	Unsupervised Machine Learning based techniques (e.g. Clustering, Principal component analysis (PCA))	Dimension-wise prediction-based measures [6] Dimension-wise distance-based measures [7] -[9] k-way feature marginals /PCA marginals [9]	Binary Numerical Categorical
	Histogram based techniques	Visualisation techniques to assess and compare statistical distributions	Binary Numerical Categorical
	Heat maps / Correlation metrics	Visualisation techniques to assess correlation between features (within a dataset as well as between original and synthetic data)	Numerical
Partially Synthetic Data	Disclosure Risk (In addition to supervised/ unsupervised ML based evaluation tasks mentioned above)	True/False match rate	Binary Numerical Categorical
	Histogram based techniques	Visualisation techniques to assess and compare statistical distributions	Binary Numerical Categorical
	Heat maps / Correlation metrics	Visualisation techniques to assess correlation between features (within a dataset as well as between original and synthetic data)	Numerical

Table 7 Summary of Evaluation Metrics for Synthetic Data Generation

5 Tool Development and Testing

In this section, three tools identified from the literature review (section 3) have been selected and evaluated using a dataset from section 1.2, and metrics from section 4. The three tools are:

1. DP-auto-GAN – A GAN based mimicking technique.
2. Synthetic Data Vault (SDV) – A statistics-based mimicking technique that supports relational datasets.
3. Presidio – A data redaction/masking tool for text.

GANs offer the possibility to mimic complex data distributions. DP-auto-GAN was the chosen GAN implementation as it supports both numerical and categorical data and it showed good accuracy scores whilst maintaining a high level of differential privacy compared to other methods. DP-auto-GAN was assessed with selected columns from the Los Angeles Crime dataset, selected for its similarity to maintenance data.

The synthetic Data Vault (SDV) is a statistics based data mimic technique, able to work with relational data set. This technique was chosen as the Authority expressed interest in mimicking relational data. SDV was assessed using the Czech Financial dataset from the Relational Dataset Repository described in 2.1.6.

Presidio is a data protection and anonymization tool, able to detect a variety of potentially sensitive information within text and take action to remove, obscure or anonymize that information. Presidio was chosen as it is free to use and appears more advanced than other tools, using a combination of rule based and trained ML models for detection. Presidio was assessed using an arbitrary subset of the Resumes dataset described in 2.1.5, selected for its similarity to free text from export licences and as it would likely contain entities that Presidio is configured to detect by default. With anonymization and redaction well understood, the assessment focused on measuring the accuracy of detecting sensitive information.

A combination of histograms and correlation heatmaps were chosen to visualise how well the data mimic techniques reproduce features and the relationships between them. These graphs give greater insight into their performance than a single measure would. For DP-auto-GAN, feature-wise prediction scores were also calculated and graphed to further examine how well inter-feature correlation is preserved.

Presidio was assessed on its ability to classify sensitive information. There is no ground truth information for the Resumes data set and it is impractical to create this manually within available time, so Presidio was manually assessed on the accuracy of the detections it reported.

These three tools provide a good coverage of the techniques identified in the literature review and therefore should provide a good idea of performance and maturity of the state of art techniques. The table below summarises which dataset and metrics have been selected for each tool.

Tool	Dataset	Metrics
DP-auto-GAN	LA Crime	Unsupervised evaluation metrics
Synthetic Data Vault (SDV)	Relational Dataset Repository	Unsupervised evaluation metrics
Presidio	Résumé dataset	Manual accuracy assessment

Table 8 Overview of the tools, datasets and metrics

The following sections describe the work performed and our experience of each tool.

5.1 DP-auto-GAN

DP-auto-GAN is downloaded or cloned from GitHub [39] . The implementation consists of a mixture of Python code using PyTorch and scikit-learn, Jupyter notebooks and some evaluation code in 'R'.

5.1.1 Evaluation Environment

To run the Python code required an appropriate Python environment. This environment was setup using Anaconda3 which provides specified packages and resolves their dependencies. Information regarding Anaconda3 can be found from the Anaconda website [63] . The conda command required to setup a Python environment named ASC259.dpautogan is:

```
$ conda create --name ASC259.dpautogan python=3.6 pytorch-gpu matplotlib
scikit-learn pandas numpy jupyter cudatoolkit=10.0
```

The environment could then be activated using

```
$ source activate ASC259.dpautogan
```

From here Jupyter notebooks can be started and accessed from a web browser at the indicated URL.

```
(ASC259.dpautogan) $ jupyter notebook --no-browser
```

A complete list of installed conda packages and their versions is shown in Appendix A.

The environment was setup on two cluster nodes from AI Labs Avalon2 cluster. Two nodes were used to aid availability in competition with other tasks. Both nodes have four 16 core AMD Opteron processors, 512GiB DRAM and a GPU processor with 11GiB DRAM. One GPU is an nVidia TESLA K40c and the other an nVidia GTX 1080ti. The two GPUs differing mostly in achieved processing speed, but assumed to produce equivalent results.

5.1.2 Evaluation of Published Results

The DP-auto-GAN conference paper [38] evaluates the technique against two datasets. These are MIMIC-III which contains unlabelled binary data and ADULT which contains unlabelled mixed-type data. The MIMIC-III dataset is not readily available as it contains sensitive medical data and requires the user to take a training course and apply formally for access to this data. The ADULT dataset is contained within the GitHub download. Evaluation of published results was performed with the ADULT dataset. The ADULT dataset comprises 48,842 rows of 15 columns and contains information

about working adults extracted from U.S. Census data. Of the 15 columns, 10 are categorical data, four are real-valued. The remaining numeric 'fnlwgt' column is ignored. This data is split into 32,562 training samples and 16,280 test samples. To assess data generation the paper compares accuracy scores achieved using a random forest classifier to predict one column of test data from the remaining columns, comparing the performance between classifiers that were trained on real data and synthetic data.

For UCI data generation the 'README.md' from the DPautoGAN GitHub provides minimal guidance on the use of the uci.ipynb Jupyter notebook. The notebook contains no comments to describe its operation. For the remainder of this section reference to the cells in the notebook will be by number, starting at 1 for the cell at the top of the notebook and incrementing for each subsequent cell. Examination of the notebook reveals that cells 1 through 4 load and prepare the ADULT dataset. Cells 5 through 7 setup and perform training of the auto encoder, saving it to file named 'ae_eps_inf.dat' for later use. Cells 8 through 10 setup and perform training of the GAN, saving it to a file named 'gen_eps_inf.dat' for later use. It is expected that this filename would normally be adjusted to match the epsilon value achieved with the GAN hyper parameters. Cell 11 performs accuracy and F1 scoring using the real training and real test data, giving a reference value for the performance of a classifier on real data. Cell 12 uses the trained Auto-encoder and GAN to generate synthetic data, where the data generated has categorical fields encoded as an index value and would require further work to replace this with category text. Cell 13 calculates accuracy and F1 score for a classifier trained on synthetic data to predict the 'salary' category. Cell 14 plots histograms of 'age' for both real and synthetic data, supporting a comparison of age distribution. Cell 15 calculates prediction scores of each field for real and synthetically trained classifiers and produces a dimension-wise prediction scatterplot for comparison. This uses a smaller random forest than used in Cells 11 and 13, with n_estimators = 10, and so typically produces lower F1 scores.

The uci.ipynb notebook, found within the uci folder of the cloned GitHub repository, was used to train an auto-encoder, train a generator and to evaluate the synthetic data generated. It is assumed that the hyper-parameters provided for training of the auto-encoder and the GAN are representative of those used to generate the results found in the conference paper. When training the GAN the output reports achieved epsilon of 0.977. Assessing 'salary' field prediction based solely on real data using Cell 11 gave 84.4% accuracy, compared to the paper-published real dataset accuracy of 86.63%. The reason for the slightly lower accuracy score is unknown and not investigated further as it is deemed similar. The associated F1 score measured for predicted salary field was 0.647. Creation of the dimension-wise prediction scatterplot using Cell 15 gave the results shown in Table 9 and depicted in the graph shown in Figure 7. In this scatterplot the red dots correspond to real-valued fields with performance measured using the R2 score. The blue dots correspond to categorical fields with performance measured using the F1 score. The single green dot corresponds to the salary field which is a real-valued field that is treated as categorical binary by applying a threshold at $50,000 and is scored using the F1 score. Real-valued capital-gain and capital loss have negative synthetic scores and do not appear within the range of the graph. This graph is similar to those in the conference paper. Each run of the assessment produces similar but slightly different results.

Table 9 Prediction scores comparison

Field	Type	Score method	Real	Synthetic
age	positive int	r2_score	0.282	0.219
workclass	categorical	F1 score	0.715	0.652
education-num	categorical	F1 score	0.338	0.293
marital-status	categorical	F1 score	0.818	0.731
occupation	categorical	F1 score	0.314	0.144
relationship	categorical	F1 score	0.759	0.549
race	categorical	F1 score	0.837	0.392
sex	categorical binary	F1 score	0.867	0.776
capital-gain	positive float	r2_score	0.084	-0.183
capital-loss	positive float	r2_score	0.025	-3.34
hours-per-week	positive int	r2_score	0.039	-0.001
native-country	categorical	F1 score	0.884	0.815
salary	categorical binary	F1 score	0.633	0.302

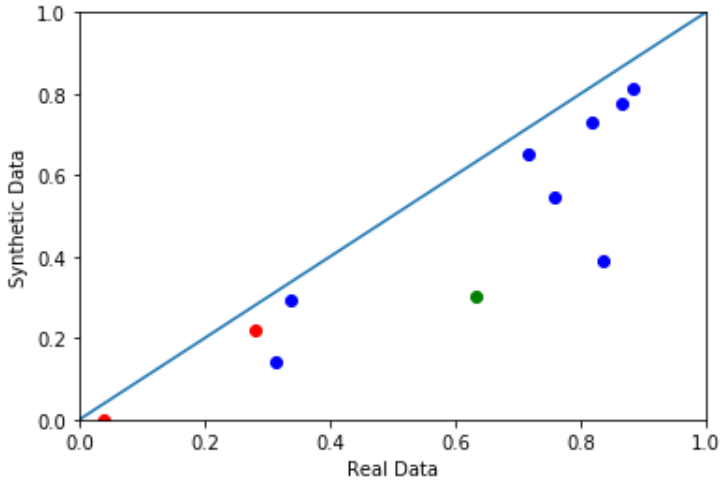


Figure 7 Prediction scatterplot (ε = 1)

5.1.3 Evaluation Using Selected Datasets

The crime in Los Angeles dataset summarised in 2.1.2 was selected for evaluation of DP auto GAN. The Jupyter notebook used to train and evaluate the ADULT data set is closely bound to the dataset itself so the source code from the 'uci' folder was copied to a new 'la' folder, renaming 'uci.ipynb' to 'la.ipynb' as a base for training and assessment with the crime in Los Angeles dataset.

An initial inspection and analysis of the dataset was performed for the purpose of deciding how each column should be used and if any pre-processing would be required. Where this dataset differs from the ADULT data set is that some fields are not 100% populated. When fields that are not populated are read from CSV files by the pandas library functions they are filled with NaN (Not a Number) for numbers or the equivalent "nan" string. The crime in Los Angeles dataset is also much larger than the ADULT dataset, containing circa 1.58 million rows compared to 49 thousand rows. Analysis of the dataset properties to find the number of empty fields and unique values for each column is shown in

Table 10. Where a column contains NaNs this will be counted as a unique value. The analysis was used to aid selection of columns for assessment of DP auto GAN.

Table 10 Crime in Los Angeles data assessment

Column Name	Rows	NaNs (empty field)	% Populated	Unique Values
DR Number	1584316	0	100.000	1584316
Date Reported	1584316	0	100.000	2809
Date Occurred	1584316	0	100.000	2809
Time Occurred	1584316	0	100.000	1438
Area ID	1584316	0	100.000	21
Area Name	1584316	0	100.000	21
Reporting District	1584316	0	100.000	1280
Crime Code	1584316	0	100.000	138
Crime Code Description	1584316	412	99.974	135
MO Codes	1584316	171759	89.159	347659
Victim Age	1584316	128659	91.879	91
Victim Sex	1584316	145199	90.835	6
Victim Descent	1584316	145232	90.833	21
Premise Code	1584316	76	99.995	296
Premise Description	1584316	2751	99.826	211
Weapon Used Code	1584316	1059559	33.122	81
Weapon Description	1584316	1059560	33.122	80
Status Code	1584316	2	100.000	10
Status Description	1584316	0	100.000	6
Crime Code 1	1584316	7	100.000	146
Crime Code 2	1584316	1484319	6.312	140
Crime Code 3	1584316	1582133	0.138	54
Crime Code 4	1584316	1584247	0.004	12
Address	1584316	0	100.000	70968
Cross Street	1584316	1321583	16.583	11122
Location	1584316	9	99.999	60609

To support initial evaluation and avoid significant pre-processing of the data a subset of the columns was used:

- Time Occurred – Time of day in minutes from 0 to 2400 and treated as a positive integer.
- Area ID and Area Name – A numeric id and related string description, both treated as categorical.
- Crime Code and Crime Code Description - A numeric id and related string description, both treated as categorical. Crime code description is not fully populated.
- Victim Age – Numeric value treated as positive integer. Not fully populated.
- Victim Sex - String treated as categorical. Not fully populated.
- Victim Descent – String treated as categorical. Not fully populated.
- Status Code and Status Description – Related strings both treated as categorical. Only 2 Status Codes are not populated.

- Location – This contains 2 floats which are converted into separate Lat (latitude) and Lon (longitude) fields and treated as float values.

Where pairs of columns represent the same information, both were included as they provide columns with strong correlation and allow assessment of how well data synthesis preserves that relationship. Victim Age was initially selected, but was later removed as the software was unable to work with NaNs in numeric fields. Examples of numeric value are provided by Location and Time Occurred columns and removing rows where Victim Age contained NaNs would have reduced the dataset by 8%. The location column, converted to two numeric float values, also has some rows with empty fields. As there are only nine empty entries, those rows were removed from the dataset to avoid errors. The dataset was split into training and test datasets with 4/5:1/5 proportions respectively.

The auto encoder component of DP auto GAN was trained using the selected data. As the number of columns is similar to the ADULT dataset and choosing appropriate hyper-parameters may be an iterative and time-consuming process, the original hyper-parameters from uci.ipynb were used.

```
ae_params = {
    'b1': 0.9,
    'b2': 0.999,
    'binary': False,
    'compress_dim': 15,
    'delta': 1e-5,
    'device': 'cuda',
    'iterations': 10000,
    'lr': 0.005,
    'l2_penalty': 0.,
    'l2_norm_clip': 0.012,
    'minibatch_size': 64,
    'microbatch_size': 1,
    'noise_multiplier': 2.5,
    'nonprivate': True,
}
```

Training converged to low loss values within several thousand iterations. The final loss printed was at 9,000 iterations and showed a loss of 0.006241 and a validation loss of 0.007770.

The generator (GAN) component of DP auto GAN was then trained with the same dataset and previously trained auto encoder. The hyper-parameters used were also taken from uci.ipynb, but with the noise multiplier reduced to 0.84 to produce an epsilon value close to 1.0.

```
gan_params = {
    'alpha': 0.99,
    'binary': False,
    'clip_value': 0.01,
    'd_updates': 15,
    'delta': 1e-5,
    'device': 'cuda',
    'iterations': 15000,
    'latent_dim': 64,
    'lr': 0.005,
```

```
    'l2_penalty': 0.,
    'l2_norm_clip': 0.022,
    'minibatch_size': 128,
    'microbatch_size': 1,
    'noise_multiplier': 0.84,
    'nonprivate': False,
}
```

It was found that training the GAN directly after training the auto encoder gave a CUDA memory error. This was assumed to result from the large amount of GPU memory still in use following auto encoder training. Restarting the notebook kernel and then running only those cells needed to prepare for GAN training allowed it to run. It was observed that the GPU utilisation during training was typically 11% to 12%, indicating potential for performance improvement.

Initial assessment of synthetic data was performed using performance scores of predictions for random forest classifiers trained on both real data and synthetic data. The scores measured and comparison scatterplot is shown in Figure 8. This generally shows good alignment between real and synthetic predictions, with most points remaining close to the diagonal. Looking at the columns where we expect high prediction scores we observe they correlate very strongly with other columns.

Name	*Type*	*Method*	*Real*	*Synthetic*
Time Occurred	positive int	R2 score	-0.008	-0.509
Area ID	categorical	F1 score	0.684	0.742
Area Name	categorical	F1 score	0.702	0.776
Crime Code	categorical	F1 score	0.978	0.482
Crime Code Description	categorical	F1 score	0.982	0.478
Victim Sex	categorical	F1 score	0.616	0.516
Victim Descent	categorical	F1 score	0.467	0.348
Status Code	categorical	F1 score	0.999	0.909
Status Description	categorical	F1 score	1	0.949
Lat (Location)	float	R2 score	-0.002	-0.001
Lon (Location)	float	R2 score	-0.002	-0.006

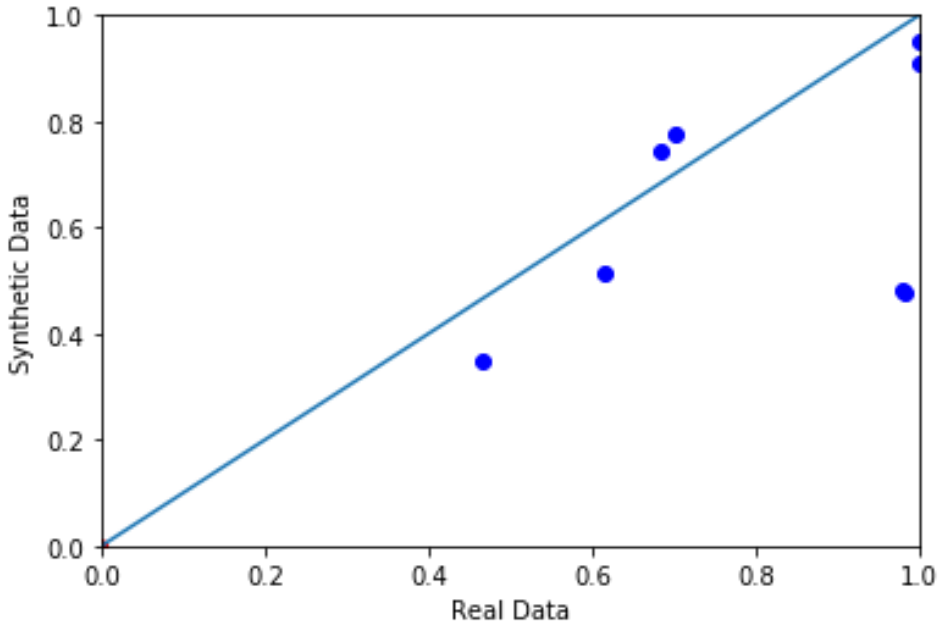


Figure 8 Crime in Los Angeles first run prediction scatter plot (ε = 1)

'Status Code' and 'Status Description' both score above 0.9 for both real and synthetic scores. 'Area ID' and 'Area Name' both score in the region of 0.75, which is lower than anticipated as each can potentially be predicted accurately from the other. This may indicate differences in category coverage between test and training data, but has not been examined. 'Crime Code' and 'Crime Code Description' have prediction scores slightly less than 1.0 when using a classifier trained on real data, the score drops to a little under 0.5 when using a classifier trained on synthetic data. This indicates that the trained generator is not synthesising representative data for these columns. As these

columns have a larger number of categories (See Table 10) this may be caused by insufficient entropic capacity of the auto-encoder network, latent space representation, and GAN network or insufficient training of the GAN network.

To see if additional training iterations would improve generated data scores, the GAN training was rerun with 60,000 iterations (RUN-2). The prediction performance for the resulting GAN is shown in Figure 9. This shows a small drop in F1 score for the 'Area' columns but a general increase in score of other categorical columns. For example, Crime Code score increased from 0.482 to 0.612 with iterations increased from 15,000 to 60,000. Training iterations were kept at 60,000 for following training runs.

Name	*Type*	*Method*	*Real*	*Synthetic*
Time Occurred	positive int	r2_score	-0.008	-0.643
Area ID	categorical	F1 Score	0.684	0.710
Area Name	categorical	F1 Score	0.702	0.755
Crime Code	categorical	F1 Score	0.978	0.612
Crime Code Description	categorical	F1 Score	0.982	0.613
Victim Sex	categorical	F1 Score	0.616	0.538
Victim Descent	categorical	F1 Score	0.467	0.389
Status Code	categorical	F1 Score	0.999	0.955
Status Description	categorical	F1 Score	1	0.959
Lat	float	r2_score	-0.002	0
Lon	float	r2_score	-0.002	-0.003

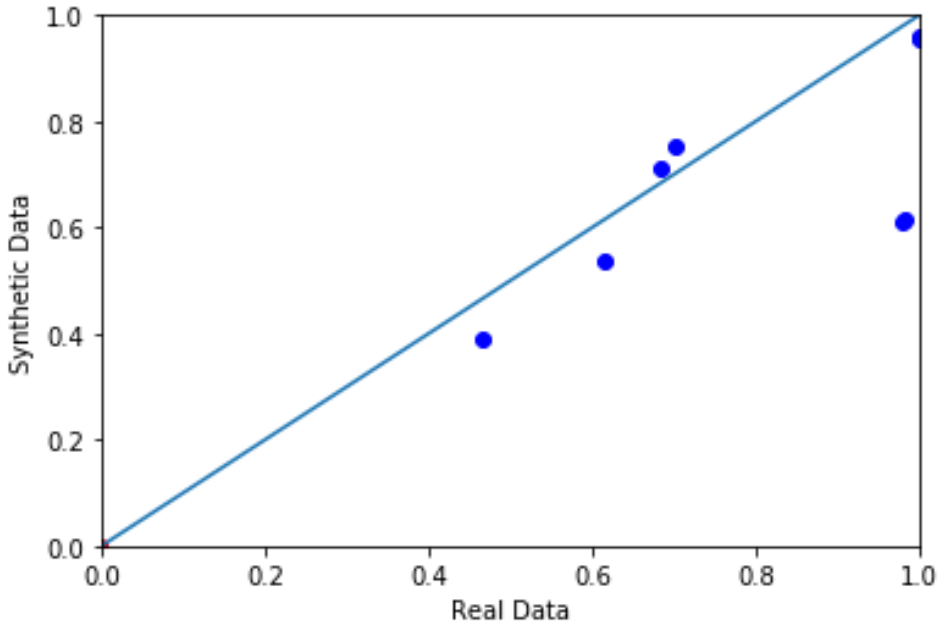


Figure 9 Crime in Los Angeles RUN-2 prediction scatter plot (ε = 1)

Histograms were produced to investigate how well the synthetic data matched the statistical distribution of the original data. These include:

- Histograms comparing the distribution of categorical data in a single column between training and synthetic data.
- Histograms comparing the distribution of numerical data in a single column between training and synthetic data.
- 2D Histograms showing the correlation between two categorical columns from a single data set. Used to see if correlation between columns is preserved in the synthetic data.
- 2D Histogram of 2 numerical columns from a single data set. Used to compare a heatmap of real and synthetic location data.

Inspection of graphs produced for numerical fields for RUN-2 identified two potential problems with the data, adversely affecting the synthetic data distributions.

Figure 10 shows a histogram of the 'Time Occurred' column for both training and synthetic data taken from RUN-2. 'Time Occurred' is an integer representing the time of day with the first 2 digits (1,000s and 100s) representing the hour of the day and the last two the minutes in the hour. Therefore, for each hour only the first 59 minutes of the 100 minute range is a valid value. For example, for the 11th hour of the day 1,000 through 1,059 is valid and 1,060 through 1,099 is invalid. This can be seen in the histogram train data which shows gaps at intervals of 100.It is also evident that time occurred reports are often rounded to an hour and less so to the half hour.

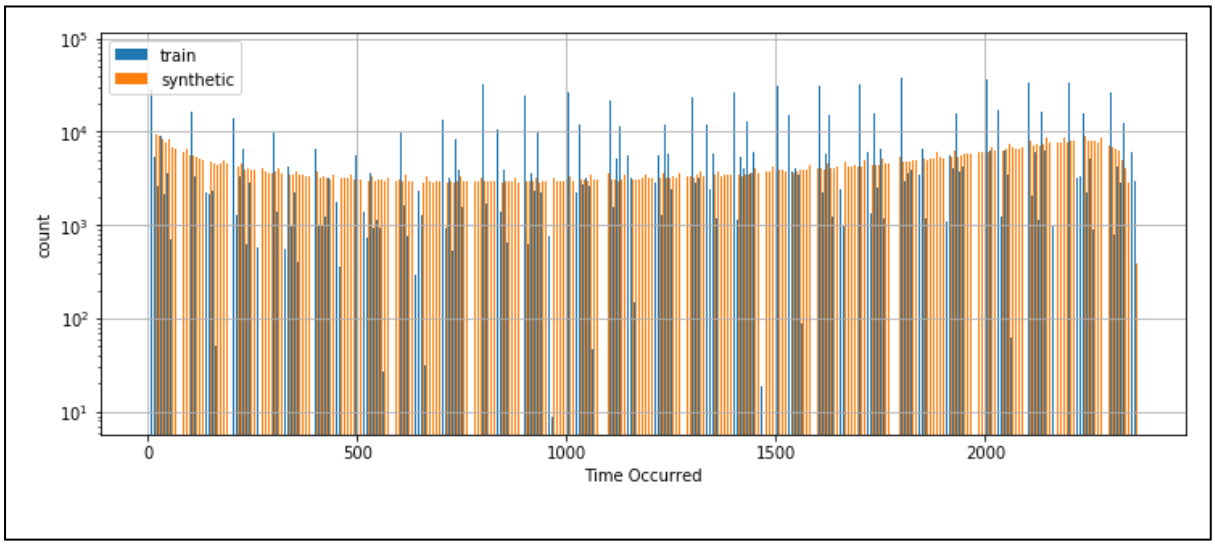


Figure 10 'Time Occurred' Histogram for training and synthetic data (RUN-2)

Figure 11 shows histograms of latitude and longitude of crime locations for both training and synthetic data. This shows significant difference between the real and synthetic data. It also reveals that there are a small number of zero 'lat' and 'lon' values. These may be causing the long tails toward zero in the synthetic data distribution.

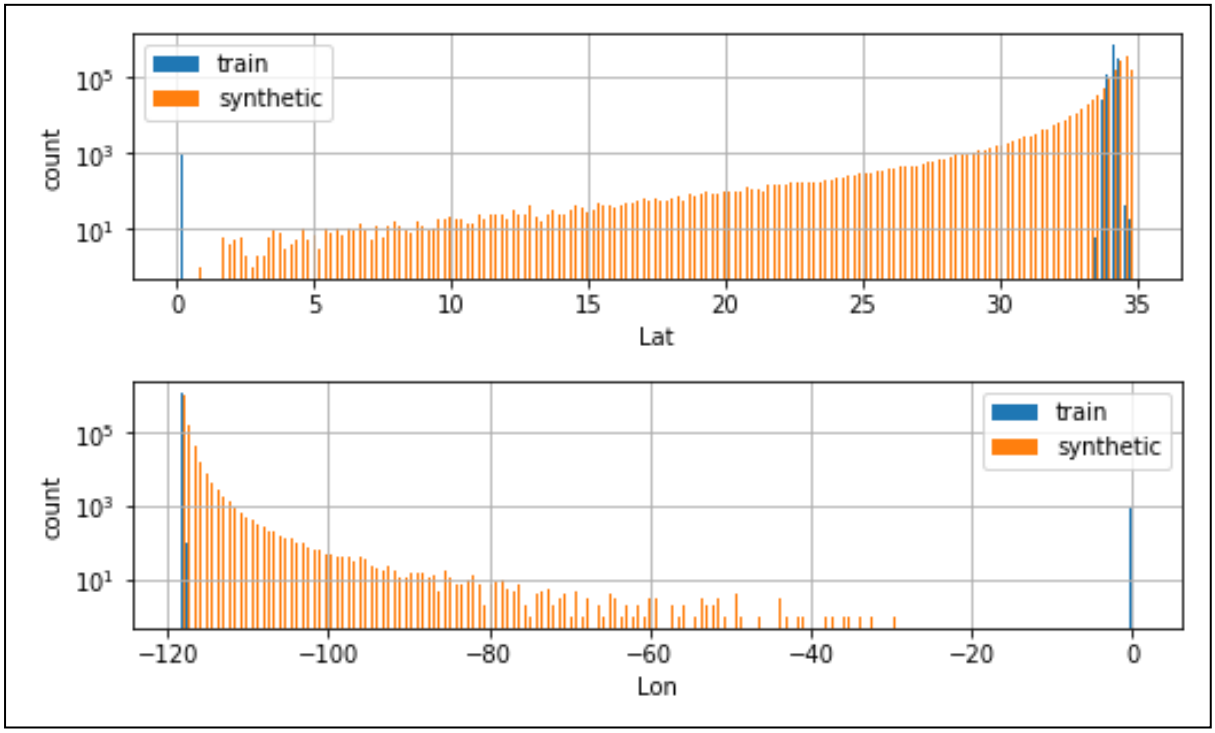


Figure 11 'Lat' and 'Lon' Histograms for training and synthetic data (RUN-2)

For the next two DPautoGAN training runs, RUN-3 and RUN-4, firstly 5,482 rows containing zero in 'lat' or 'lon' were removed and then 'Time Occurred' format was converted to integer minutes of the day. For brevity the results from RUN-3 are not shown here as the effects of removing zero 'lat','lon' rows are included in RUN-4. Results are shown in Figure 12.

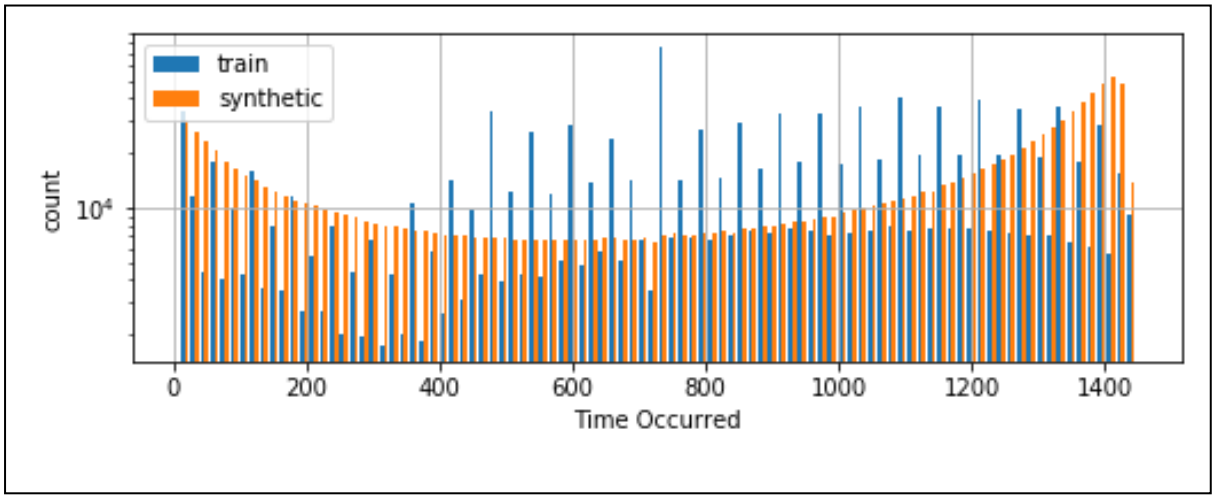


Figure 12 'Time Occurred' Histogram for training and synthetic data (RUN-4)

Figure 12 Shows histograms comparing the 'Time Occurred' column from training and synthetic data following the change to time of day in minutes. The synthetic numeric range is consistent with the training data but the shape of the distribution is not a good match. It is again clear to see the pattern

of more times recorded on the hour then half past the hour and in turn, a quarter past and a quarter to the hour. This pattern is not evident in the synthetic data.

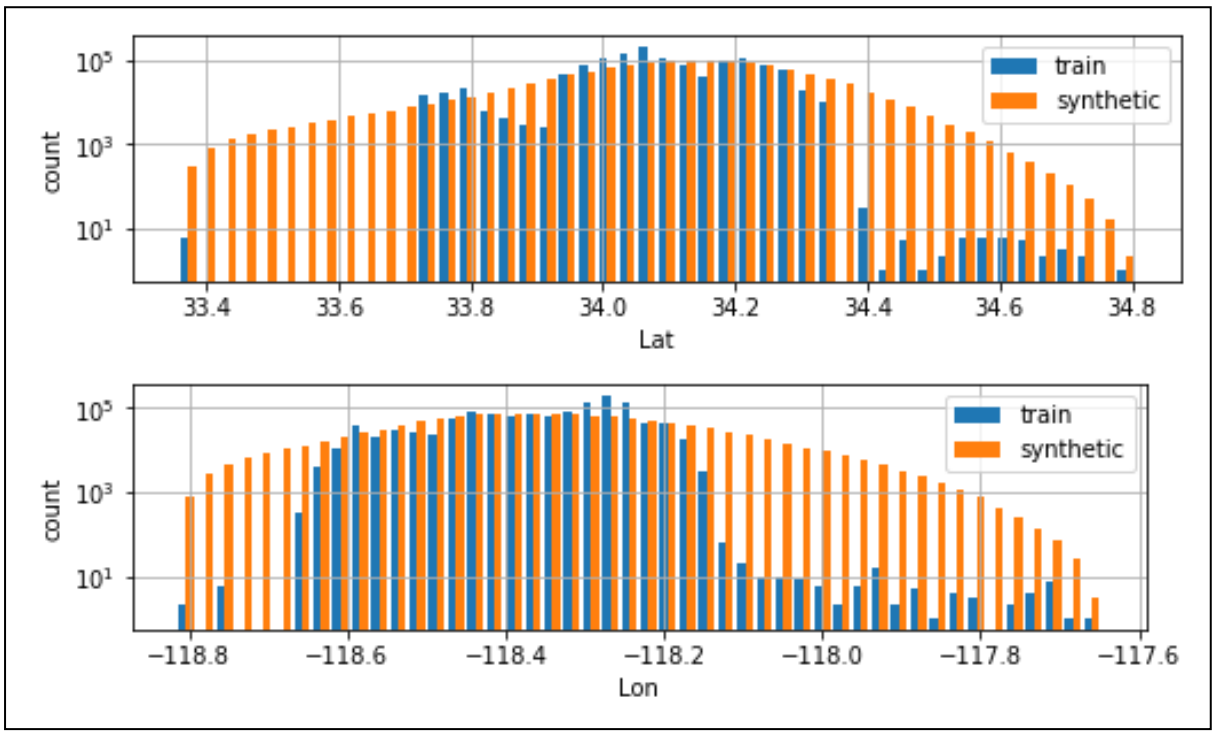


Figure 13 'Lat' and 'Lon' Histograms for training and synthetic data (RUN-4)

Figure 13 shows histograms of the crime location latitude (lat) and longitude (lon), comparing training data and synthetic data. With the zero value rows removed from the training data, the synthetic data is again consistent with the range of training data. The synthetic distributions show some consistency of shape with the training data but does not mimic the detail. Figure 14 shows the same location data as heatmaps. Further showing that the synthetic data covers the same area but does not represent the detail.

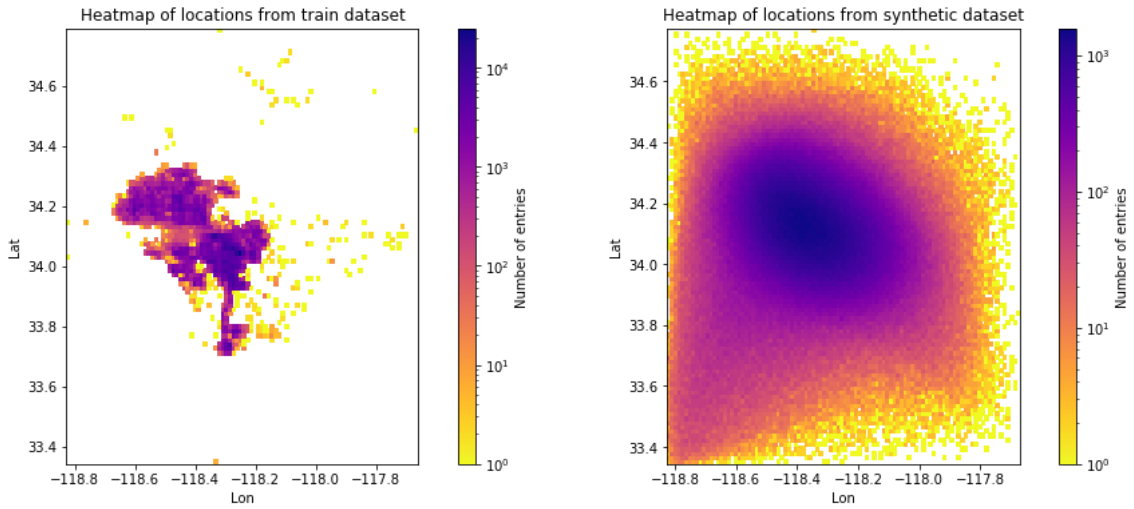


Figure 14 Heatmaps of training and synthetic location (RUN-4)

Histograms for each column comparing training and synthetic data are shown in following figures to compare the statistical distribution of categorical columns. These were all created from RUN-4 data.

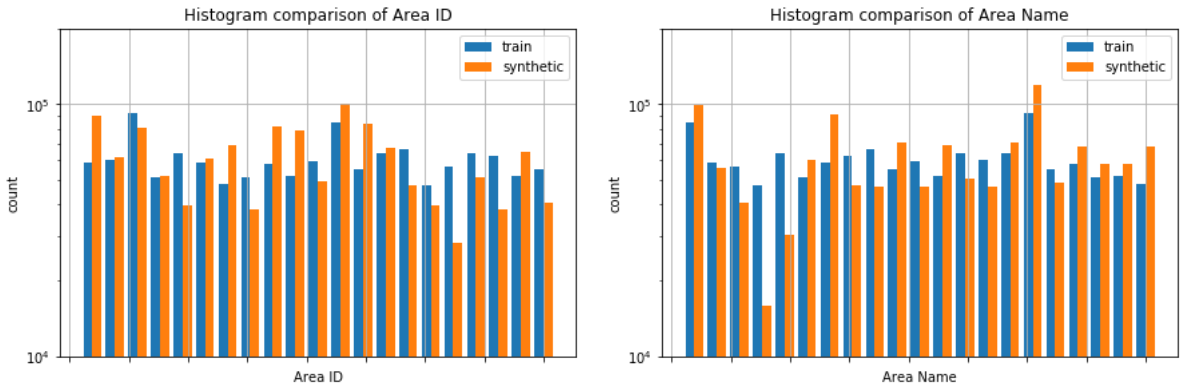


Figure 15 Histograms comparing 'Area' columns (RUN-4)

Figure 15 show histograms comparing the 'Area' columns from training and synthetic data. The graphs show some variation between training and synthetic categories, but category counts are generally similar with only a couple deviating significantly.

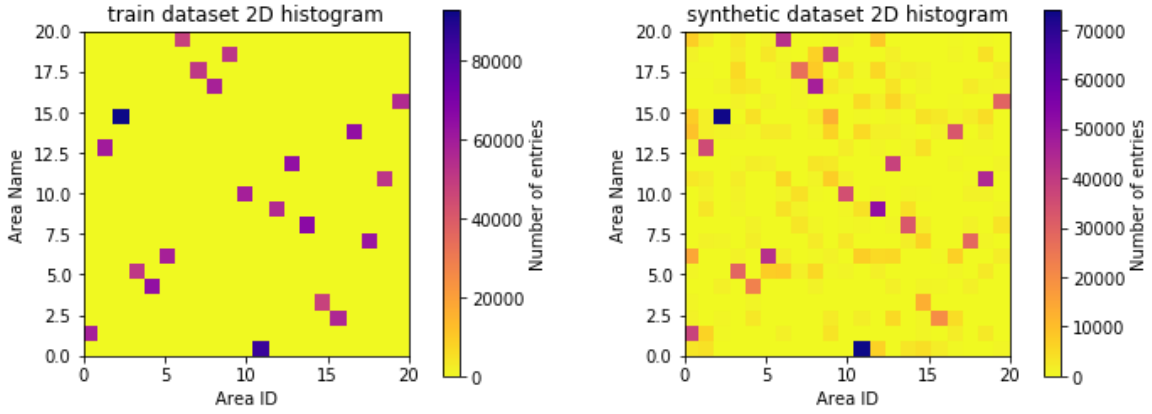


Figure 16 2D Histograms of 'Area' column correlation (Run-4)

'Area ID' and 'Area Name' columns represent the same information and are therefore strongly correlated. They were both included in the training data to support investigation of how well synthesis preserved the correlation between columns, although in a real application only one would be included. 2D histograms or 'correlograms' of these columns were plotted to examine whether this correlation is preserved in the synthetic data, shown in Figure 16. With the yellow background representing zero counts, the left histogram of training data shows one-to-one relationship between the 'Area ID' and 'Area Name' column. Each row or column has only one non-zero count. The right histogram of synthetic data shows a similar pattern, with the correlated categories showing significant counts. However, many category combinations have a non-zero count where zero is expected.

Figure 17, Figure 18, and Figure 19 show histograms for the remaining categorical columns. Inspection of these shows that by proportion the synthetic data gives similar counts for the higher populated categories and deviates more significantly for the comparatively lower population

categories. This shows that the generator is able in some degree to mimic the statistical distribution of categorical data.

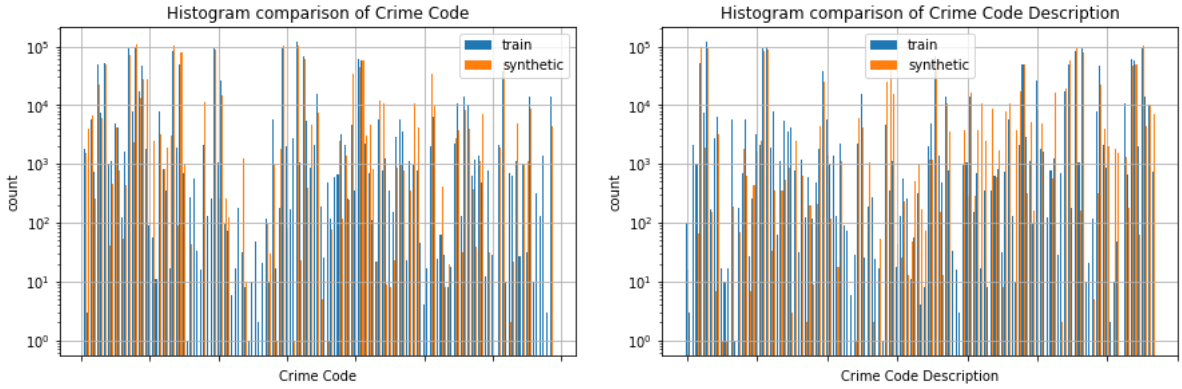


Figure 17 Histograms comparing 'Crime' type columns (RUN-4)

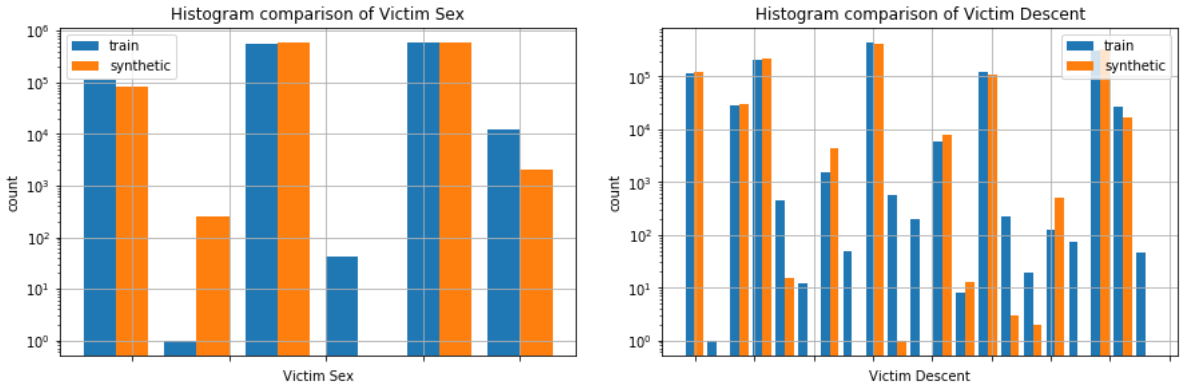


Figure 18 Histograms comparing 'Victim' columns (RUN-4)

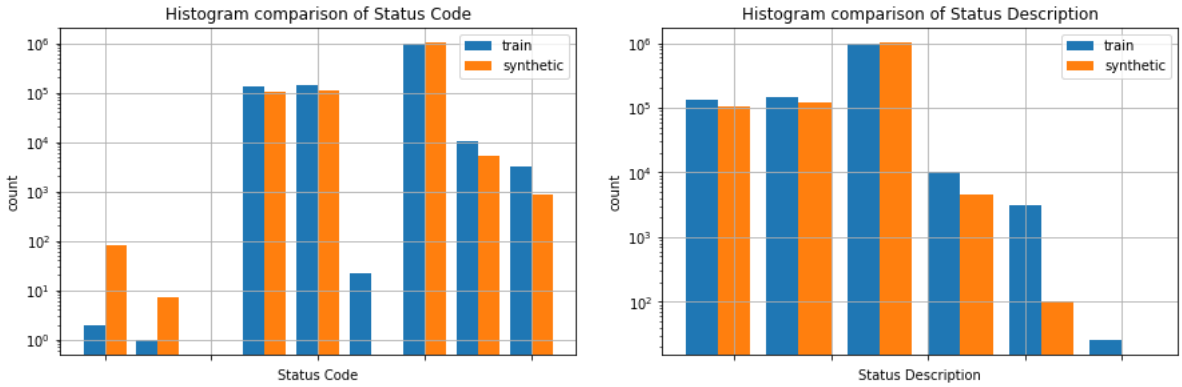


Figure 19 Histograms comparing 'Status' columns (RUN-4)

In RUN-4 the noise multiplier parameter used in generator (GAN) training was set to produce an Epsilon value close to 1. Two further training runs were performed to examine the effects that changing Epsilon has upon classifier performance. With other training parameters fixed, for RUN-5 and RUN-6, the noise multiplier value was adjusted to achieve values for Epsilon close to 1000 and 0.024 respectively. The initial aim was to configure RUN-6 for an achieved Epsilon of 1/1000. As the noise multiplier was increased, Epsilon approached a minimum value that was greater than the desired target. With the desired target value not possible, a value of 10.0 was chosen as it is a similar factor increase compared with the ratio between RUN-4 and RUN-5 and it gives a value of Epsilon

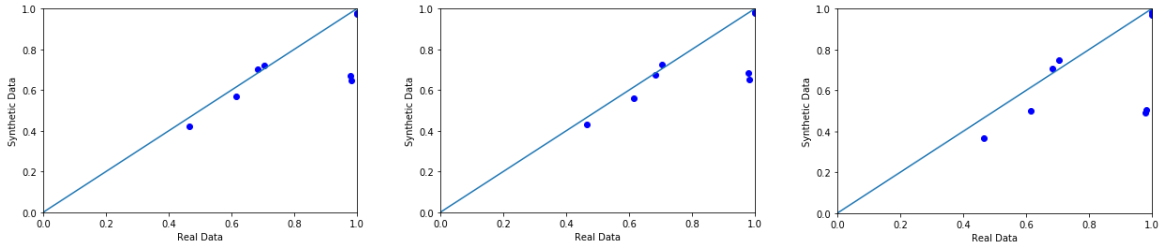
close to the minimum available. The noise multiplier values used and the resulting achieved Epsilon are shown in Table 11 for training runs RUN-4, RUN-5 and RUN-6.

Run	*Noise multiplier (GAN Training)*	*Epsilon (ε) Achieved*
RUN-4	0.84	1.024
RUN-5	0.0954	1000
RUN-6	10.0	0.024

Table 11 Achieved Epsilon

Figure 20 shows the performance scores for predictions of random forest classifiers trained on real data and on synthetic data with varying achieved Epsilon. Referred to in the following text as real performance and synthetic performance respectively. All of the scores for predicting numeric values are very low so these are not considered further. In general the synthetic performance is similar to the real performance for categorical data, with values close the diagonal in the scatter plots. Exceptions to this are the 'Crime Code' and 'Crime Code Description' where the synthetic performance is significantly lower, indicating the generated synthetic data does not match the real data well.

Name	*Type*	*Method*	*Real*	*RUN-4* $\varepsilon = 1.024$	*RUN-5* $\varepsilon = 1000$	*RUN-6* $\varepsilon = 0.024$
Time Occurred	positive int	r2_score	0.016	-0.684	-0.671	-0.731
Area ID	categorical	F1 Score	0.685	0.701	0.674	0.709
Area Name	categorical	F1 Score	0.703	0.720	0.724	0.75
Crime Code	categorical	F1 Score	0.978	0.672	0.685	0.492
Crime Code Description	categorical	F1 Score	0.982	0.648	0.652	0.506
Victim Sex	categorical	F1 Score	0.616	0.571	0.561	0.502
Victim Descent	categorical	F1 Score	0.467	0.421	0.433	0.367
Status Code	categorical	F1 Score	0.999	0.976	0.978	0.968
Status Description	categorical	F1 Score	1	0.977	0.983	0.985
Lat	float	r2_score	-0.093	-0.013	-0.002	-0.057
Lon	float	r2_score	-0.575	0	0	0



RUN-4 RUN-5 RUN-6

Figure 20 Crime in Los Angeles prediction scores for RUN-4, RUN-5 and RUN-6.

As the noise multiplier is increased and Epsilon decreases, we expect the synthetic data to become less accurate and the synthetic performance to diverge from real performance. Inspection of the

results show this to be true for 13 of the 16 comparisons between synthetic performance results. See performance comparison in Table 12.

	Difference to Real performance				
Categorical Name	RUN-5 ε=1000	RUN-4 E=1	RUN-6 E=0.024	RUN-4 > RUN-5	RUN-6 > RUN-4
Area ID	0.011	0.016	0.024	TRUE	TRUE
Area Name	0.021	0.017	0.047	FALSE	TRUE
Crime Code	0.293	0.306	0.486	TRUE	TRUE
Crime Code Description	0.330	0.334	0.476	TRUE	TRUE
Victim Sex	0.055	0.045	0.114	FALSE	TRUE
Victim Descent	0.034	0.046	0.100	TRUE	TRUE
Status Code	0.021	0.023	0.031	TRUE	TRUE
Status Description	0.017	0.023	0.015	TRUE	FALSE

Table 12 Comparison of synthetic performance

5.1.4 Summary and Thoughts

DP-auto-GAN is a technique for mimicking a data set while introducing a measure of privacy to individual entries. The implementation used for assessment was created by its authors to support repetition and scrutiny of results presented in their published paper. As such, it is tailored to a specific data set and it synthesises data sufficient only for performance assessment.

Adapting the published code from the paper, we have applied this technique to the Los Angeles crime dataset with some success. We have shown that the statistics of the original data set are replicated in the synthesised data, although with varying quality. In numerical cases the distribution shape lacked the detail observed in the original data. In categorical cases the lower population categories were less accurately represented. This has demonstrated the technique has potential, with a more detailed investigation required to determine what level of performance is achievable.

Further thoughts:

- Tuning of auto-encoders and GANs requires appropriate machine learning expertise and experience. Training of GANs is computationally expensive, requiring significant time to iterate different implementation choices. Rules of thumb or automated methods for determining the geometry and training parameters of those components based on measurable characteristics of data would be beneficial.
- Pre-processing/transforming reference data into a difference representation may allow the technique to achieve better performance. This requires application of knowledge specific the data set.
- Strategies are required to address artefacts found in real data sets. E.g. missing numeric fields.
- For each application, consider whether the privacy provided by differential privacy is appropriate.
- The implementation is immature, with the current code developed for research purposes. This would need to be re-implemented in a more appropriate form to be applied more generally.

5.2 Presidio

Presidio is a tool for detecting and removing or anonymizing personally identifiable information found in text or images. It is a Microsoft development with source for the application, installation instructions, etc. published on GitHub as microsoft/presidio [54] . Presidio may be run either as a service under Docker or Kubernetes with access via a web based API or the core analyser component may be installed as a Python module. A demonstration web page is also provided at http://presidio-demo.azurewebsites.net/ which can process pasted text.

5.2.1 Evaluation Environment

The initial approach taken to setup an evaluation environment was to clone the microsoft/presidio git repository and use the included 'build.sh' script to build and configure a local Docker-based deployment. Initial builds were not successful as our Docker environment accesses the Internet via a corporate proxy that prevented some Docker build activities from operating correctly. Investigation of the build process found the cause of build failures was failure to verify downloads and timeout downloading the large 'en_core_web_lg' python package. To allow the build to complete, proxy certificates were added to the Docker builds and the large 'en_core_web_lg' python package was downloaded separately and provided locally. The built Docker containers did not function correctly, with the presidio-api container failing to connect to presidio-analyzer. Further investigation into this problem was abandoned and the demonstration web server used instead.

5.2.2 Evaluation Using Selected Dataset

Evaluation of Presidio's detection of sensitive information within text was performed using the demonstration web page with the Resume data described in section 2.1.5. This data is not labelled so assessment was performed manually, checking whether each detection was valid and scoring based upon the proportion of correct detections. To reduce the effort to a manageable level the assessment was limited to the five documents found at the top level of the Resume data set. A list of resume documents used is shown in Table 13.

LT CV 201608.docx
180517_Vasanthi Kasinathan.docx
eFinancialCareers_TT - CV.DOCX
CV-Gloria Cheng2018.doc
Resume --Rohini Prakash.pdf

Table 13 Resume documents

To convert the documents into a form suitable for processing on the web site and to provide a reference for comparison, each of the documents was converted into plain text format. This was achieved by opening in a viewer or editor suitable for the original file format, selecting all of the text, copying it and then pasting it into a new text document.

The Presidio demo web site was accessed using the google chrome browser. The site offers a list of search filters for a variety of sensitive text. By default, all of the search filters are enabled and are set to replace detected text with a string containing the filter's name surrounded by angle braces. This default configuration was used for the assessment. The list of search filters is shown in Table 14.

Filter Name	*Shown in findings*	*Detected but not shown in findings*
CREDIT_CARD		
CRYPTO		
DATE_TIME	y	
DOMAIN_NAME	y	
EMAIL_ADDRESS	y	
IBAN_CODE		
IP_ADDRESS		
LOCATION	y	
NRP	y	
PERSON	y	
PHONE_NUMBER		y
UK_NHS		
US_BANK_NUMBER		y
US_DRIVER_LICENSE	y	
US_ITN		
US_PASSPORT		
US_SSN		

Table 14 Presidio demo search filters

The web site has a box in which to paste text to analyse and shows the resulting text in an adjacent box as well as 'Findings' showing a list of detections, each with field name, a confidence score and start:end character positions of the sensitive text. There is a 'Text' column in the findings that was not populated. An example from the web site of the first two lines of findings is shown in Figure 21.

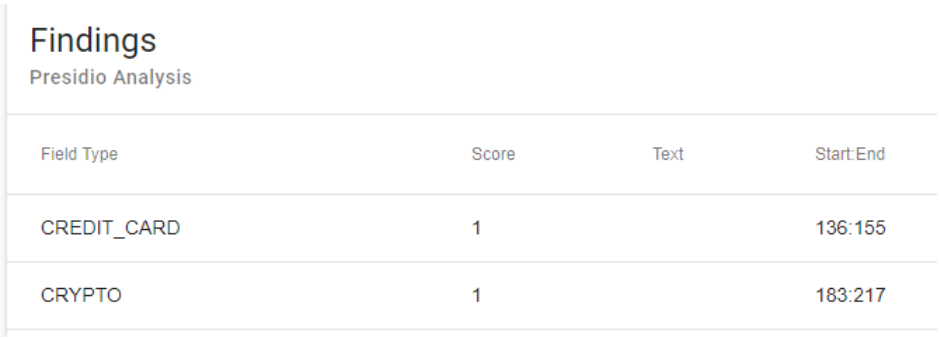


Figure 21 Example findings from Presidio web demo

To analyse each Resume document the following sequence of actions were performed:

- Copy the content of the text document into the web page's 'Input text' box.
- Copy the transformed text from the 'Anonymized text' box and record to a text file.
- Copy the list of detections from the 'Findings' box and paste into an Excel sheet.
- Split the start:end column in the Excel sheet into two columns using the ':' delimiter and sort the list by ascending 'start' character offset.
- Assess each detection in the list for accuracy, marking those that correctly identify text fields. To aid this process TortoiseSVN's diff function was used to compare original and transformed text.
- Produce a pivot table for the document to calculate accuracy for each detector and an overall score for the document.

During this process it was evident that the US_DRIVER_LICENSE filter produced a significant number of detections, but since there is no US driver license information in the documents, all of these detections are false. From inspection of detected text, the filter was triggered by any 12 letter word. E.g. stakeholders, improvements, satisfactory or requirements. As this filter appeared to be of poor quality, with a high false alarm rate, it was not included in the results. It was also observed that PHONE_NUMBER and US_BANK_NUMBER filters both produced a small number detections, but those detections were not listed in 'Findings'. The number of occurrences appeared small, so these were also not used in the results. Search filters that detected text in the Resume documents are indicated in Table 14.

The results from each of the five documents is shown in Table 15 through Table 19.

Row Labels	Detections	True Positive	Accuracy
DATE_TIME	20	20	1.000
DOMAIN_NAME	1	1	1.000
EMAIL_ADDRESS	1	1	1.000
LOCATION	6	6	1.000
NRP	1	1	1.000
PERSON	3	1	0.333
Grand Total	**32**	**30**	**0.938**

Table 15 Accuracy scores for 'LT CV 201608'

Row Labels	Detections	True Positive	Accuracy
DATE_TIME	27	27	1.000
LOCATION	21	21	1.000
NRP	2	1	0.500
PERSON	6	0	0.000
Grand Total	**56**	**49**	**0.875**

Table 16 Accuracy scores for '180517_Vasanthi Kasinathan'

Row Labels	Detections	True Positive	Accuracy
DATE_TIME	6	4	0.667
LOCATION	10	9	0.900
PERSON	2	1	0.500
EMAIL_ADDRESS	3	3	1.000
DOMAIN_NAME	3	3	1.000
Grand Total	**24**	**20**	**0.833**

Table 17 Accuracy scores for 'Resume --Rohini Prakash'

Row Labels	Detections	True Positive	Accuracy
DATE_TIME	35	35	1.000
LOCATION	45	42	0.933

Row Labels	Detections	True Positive	Accuracy
PERSON	2	1	0.500
EMAIL_ADDRESS	1	1	1.000
DOMAIN_NAME	2	1	0.500
NRP	2	2	1.000
Grand Total	**87**	**82**	**0.943**

Table 18 Accuracy scores for 'CV-Gloria Cheng2018'

Row Labels	Detections	True Positive	Accuracy
DATE_TIME	17	16	0.941
LOCATION	41	36	0.878
PERSON	5	0	0.000
EMAIL_ADDRESS	1	1	1.000
DOMAIN_NAME	2	2	1.000
Grand Total	**66**	**55**	**0.833**

Table 19 Accuracy scores for 'eFinancialCareers_TT - CV'

In total there were 265 detections of which 236 were assessed as being correct. This gives an overall accuracy of 0.891.

During assessment of detection performance, we did not quantity the number of missed detections. The following is a list of observations made during the assessment:

- As previously mentioned above, there are many false positives for US_DRIVER_LICENSE as it appears at minimum to detect 12 character words.
- Some organisation names are identified incorrectly as PERSON. E.g. J P Morgan or Apax.
- DATE_TIME detects a range of different forms of time. As well as typical date formats, it also detects period expressions such as Monthly, Annual, Weekly, etc.
- In several places the word ‘Liaise’ was identified as PERSON.
- Detection of year ranges appeared unreliable. Given a range like ‘2000 – 2005’, this might be detected as a single DATE_TIME, or as two separate ‘DATE_TIME’ fields, or with only one year marked as ‘DATE_TIME’.
- Some formats for date found in the evaluation documents were not detected. E.g. Aug’17 or July’11.
- Where a place name is included in the name of something such as an organisation or corporation. E.g. Singapore Investment Corp. This is often not identified as a LOCATION, which is correct. It is however a good location clue and may not be the desired behaviour.

5.3 Synthetic Data Vault (SDV)

Synthetic data vault (SDV) is a Python package which can be used to generate synthetic data based around relational tables containing numerical, categorical and date-time data. The tool cannot work with free text-based data. In addition, even though this process is relatively quick (compared to a GAN which needs to be trained), there are no rigorous privacy-based guarantee measures in place;

the paper for SDV states that noise could be injected by taking the covariances and then halving them (reducing the strength of covariance). However, the current version of the package does not have a parameter which can perform this.

5.3.1 Selected Dataset Characteristics

To test the tool, a dataset containing multiple tables which were related had to be selected. The Czech Financial dataset was chosen, and has been described as a PKDD'99 Financial dataset which contains 606 successful and 76 unsuccessful loans, along with information on transactions Permanent Orders ('order'), Transactions ('trans'), Disposition ('disp') and Accounts ('account').

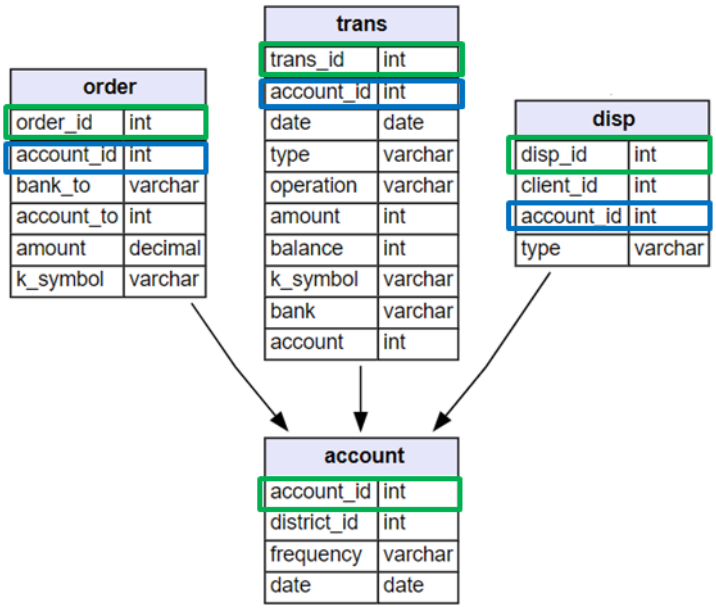


Figure 22 The structure of the Czech financial dataset.

The relationships between tables are defined by the primary and foreign keys. The primary keys are circled in green and the foreign keys are circled in blue in Figure 22. The type of data is indicated by integers ('int'), date-time ('date'), text ('varchar') and floats ('decimal').

Appendix D contains some example rows from each of the 4 tables. There are also translations for the Czech terminology present in some columns.

More details regarding the data are provided here:
https://webpages.uncc.edu/mirsad/itcs6265/group1/domain.html

5.3.2 Method

First, the SDV package was installed using a simple "pip install SDV" command and the CSV file for each table was then read in and stored in a Numpy array. After this, metadata was produced using an in-built function from the package. This would define the links between the tables – in this case, the primary and foreign keys were set for each table., In the current implementation of SDV, there can only be one foreign key per table which means a table can only be linked to one other table and not more. Nevertheless, the Github repository for SDV has stated that support for multiple foreign keys has been developed and this feature is now pending review (as of March 2020). Once the metadata has been defined and the number of synthetic rows required has been specified, the SDV package is able to produce these automatically without further instruction. Appropriate metrics include correlation heat maps and feature-wise histograms which can be used to compare the strength of relationships between columns and the distributions of values.

5.3.3 Theory

The modelling process used by SDV builds generative models for individual tables and performs extra computations using Conditional Parameter Aggregation (CPA). The input CSV tables and metadata undergo CPA to produce an extended table containing covariances and distribution values. The Gaussian Copula process is then used to generate the model. This process is outlined in Figure 23.

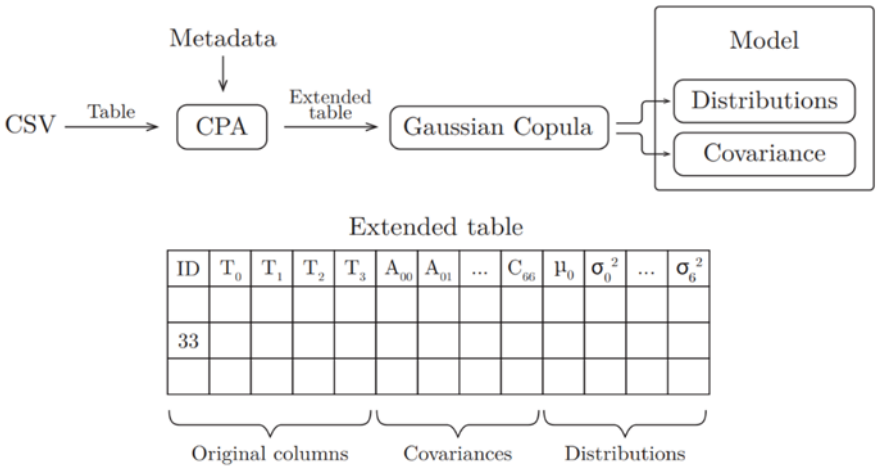


Figure 23 The SDV modelling process (top) and extended table (bottom)

In the current implement is assumed that the distribution for each column can be modelled as a multivariate Gaussian distribution. This type of distribution cannot currently be changed using SDV, but the paper describing the package has mentioned the possibility of supporting truncated Gaussian, uniform, beta and exponential distributions.

The process is as follows, as outlined in the paper [41] :

1) Iterate through each row in the table.
2) Perform a conditional primary key lookup in the entire database using the ID of that row. If there are *m* different foreign key columns that refer to the current table, then the lookup will yield *m* sets of rows. We call each set conditional data.
3) For each set of conditional data, perform the Gaussian Copula process. This will yield *m* sets of distributions, and *m* sets of covariance matrices, Σ. The paper call these values conditional parameters, because they represent parameters of a model for a subset of data from a child, given a parent ID.
4) Place the conditional parameters as additional values for the row in the original table. The new columns are called *derived columns*.
5) Add a new derived column that expresses the total number of children for each parent.

The CPA and Gaussian Copula processes assume there are no missing entries so if this is not the case, additional pre-processing must occur. Numerical, categorical or date-time columns containing missing values are hence represented with 2 columns:

- A column of the same type as the original column with missing values filled in by randomly choosing non-missing values from somewhere else in that column.
- A column containing "Yes" if the original data is present and "No" if the data was missing for that row.

As a result of this, the synthetic data generated would also contain missing values too; the authors viewed this is as a necessary characteristic to simulate since extra information of the data can be inferred from these missing values. There are several reasons why missing values may be present in the dataset:

1) Data is missing because of what it represents. For example, for a column containing the weight of people, overweight people may have chosen to not disclose their weight so a missing value may suggest they had a large weight.
2) Data is missing at random as a subgroup decided not to supply data. For instance, perhaps the majority of females did not disclose their weight so knowing that a person is female may make it more likely that their weight value will be missing.
3) Data is missing completely at random. For example, an admin error meant the values were not entered in or accidentally deleted so this does not tell you anything about the rest of the data.

Steps 1 and 2 above provide further information about the data and therefore it was decided that it would be important to model the missing values and the current implementation this cannot be changed when generating the data.

Categorical data cannot be directly modelled by Gaussian Copula or CPA, without further pre-processing; SDV will automatically replace a categorical column with a numerical column containing values in the range [0, 1]. The process is outlined in Figure 24 and described in the following steps:

1) The categories present (e.g., Yes, No, Maybe) are sorted from most frequently occurring to least to give Pr(No)=0.5, Pr(Yes)=0.4 and Pr(Maybe)=0.1.
2) The interval [0, 1] is split into sections based on the cumulative probability for each category, as shown on the x-axis under Numerical.
3) To convert a category, the interval [a, b] ∈ [0, 1] is identified which corresponds to the category. Hence, No represents $0 \leq x < 0.5$, Yes represents $0.5 \leq x < 0.9$ and Maybe represents $0.9 \leq x < 1$ under Numerical.
4) A value between a and b is selected by sampling from a truncated Gaussian distribution with the mean μ at the centre of the interval, and standard deviation indicated by $\sigma = \frac{b-a}{6}$.

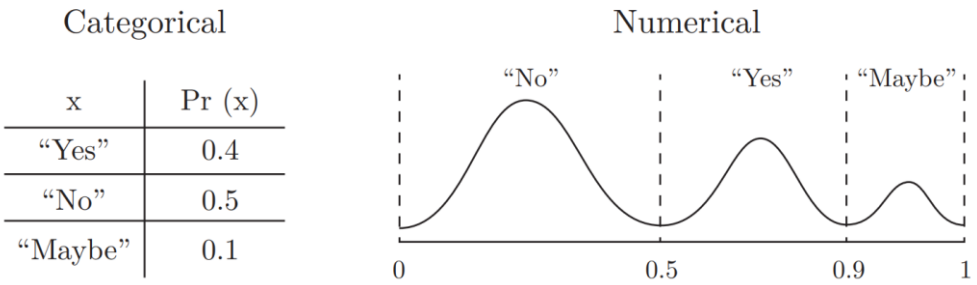


Figure 24 Categorical data and their probabilities (left), and their representation as Gaussian distributions (right)

For date-time values that are represented as text, SDV will replace such columns with numerical values. Timestamps are converted into the number of seconds past Epoch (1st January 1970), with negative values representing a time before then. Table 20 summarises how the data in the original column becomes pre-processed. Here, categorical and date-time data is replaced with numerical values; numbers, categories and date-time with missing values are separated into 2 columns, containing a value and a Boolean indicating "Yes" if it was missing, or "No" if it was not missing with the value specified being a random selection from elsewhere in the column, as discussed earlier.

Original column type	Replaced column(s) type
Categorical	Number
Date-time	Number
Number with missing values	Number & Categorical
Categorical with missing values	Categorical & Categorical
Date-time with missing values	Date-time and categorical

Table 20 SDV data conversions that take place before being modelled

5.3.4 Results and Analysis

As the original dataset contained 4,500 rows, the same amount were generated which took around 660 seconds to perform. Correlation maps were plotted to compare the relationships between the columns and histograms were produced to compare the distribution of data for each column. Both of these comparisons were made using the Python 'seaborn' and 'matplotlib' visualisation packages.

The histograms shown on the following pages indicate the overlap of distributions ; the greater the overlap, the more accurately the distributions of the real (blue) and synthetic (red) columns match. Only the columns with numerical values can be represented in these plots so categorical columns have been omitted. It should also be noted that the plot titles have the format of table name plus column name (i.e. trans_account_id refers to the "trans" table with column "account_id".

In addition, the histograms have a kernel density estimate (KDE) curve fitted for each column. The KDE estimates the probability distribution where the larger the height of the curve at a given x-value, the larger the density of observations at that value and its neighbourhood. Having histograms will show if the number of values in each bin (i.e. over a given range) has been modelled effectively, whereas the KDE allows you to compare the probability density of values (i.e. where the values are most likely to lie for the synthetic and real data).

For the trans_balance graph in Figure 35, it is clear that there is significant overlap between the distribution of the original data and the simulated data produced via Gaussian approximations. This shows that SDV has been able to model the distribution accurately for this column. In addition, the KDE curve shows that the real values for balance are, on average, a little lower compared to the synthetic dataset. The other numerical fields were very poorly simulated by the Gaussian distribution. For example, account_date (Figure 26), account_district_id (Figure 27) and trans_date (Figure 36) had multiple peaks (as shown by their KDEs) and hence, could not be represented by just one Gaussian distribution with one peak. Sampling from these distributions to produce the synthetic data is unlikely to provide an accurate representation of the real dataset. A possible reason for Account_date and Trans_date having the KDEs shown is that the information to produce the real tables was gathered on specific dates (i.e. every n days) as suggested by the gaps between columns , so instead of SDV modelling the dates as categorical/discrete, they were incorrectly modelled as continuous. On first inspection, the histograms for trans_amount (Figure 34) appear to match, however on closer inspection the distribution for synthetic data has failed to match the significant peak of lower valued transactions and much lower distributions for higher values. This means that transaction values are generally much higher than expected in the synthetic data. The Disp_client_id (Figure 29) in our example is treated as a purely numeric field (as the client table was omitted for the experiments). Synthesis produced negative numbers, which never occur in the real data. Currently a feature to enable the use of a truncated Gaussian model and set constraints, which would rectify the issue, is being developed but for now, some post-processing would need to take place.

The histograms comparing the primary keys of the tables (Figure 25, Figure 30, Figure 32 and Figure 37) suggests that SDV makes the assumption that the primary keys can start at 0 and incrementally increase. This is not representative of real data and does not cover the full range of values in the original dataset, but may be acceptable depending on the requirements. This makes the comparison of the foreign keys (Figure 28, Figure 31 and Figure 33) less meaningful as the primary keys do not cover the same range in the synthetic and real datasets. Looking at the synthetic data the range of values for the foreign keys do appear to match the corresponding range for the primary keys, suggesting that it is correctly creating valid relational data (as a foreign key entry must be present as a primary key in the corresponding table). However, at this stage, it is difficult to determine how well SDV is mimicking the foreign keys columns.

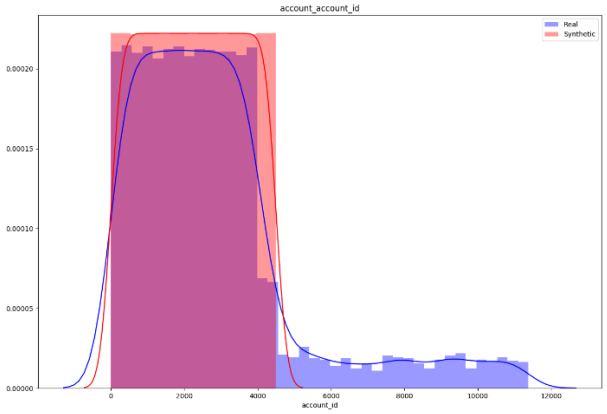


Figure 25 Table 'Account' with column 'account_id'

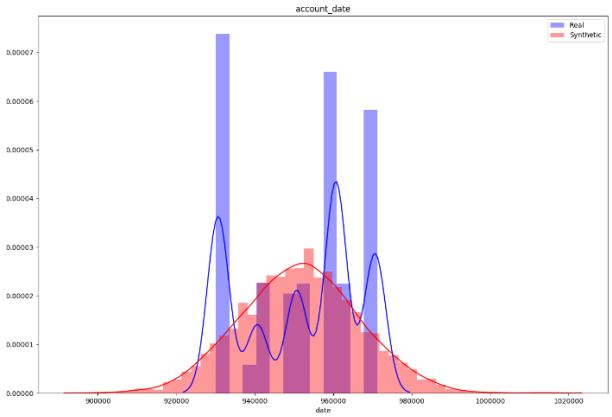


Figure 26 Table 'Account' with column 'date'

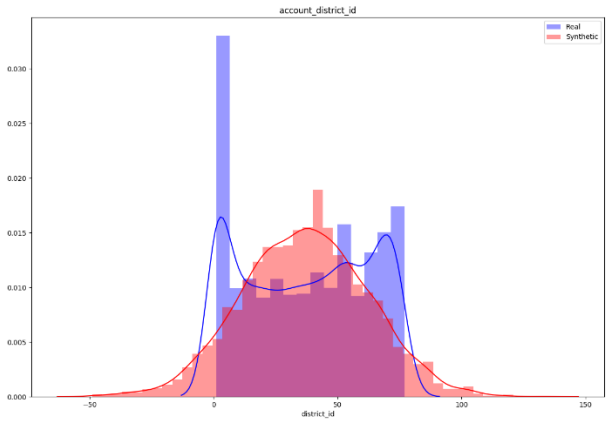


Figure 27 Table 'Account' with column 'district_id'

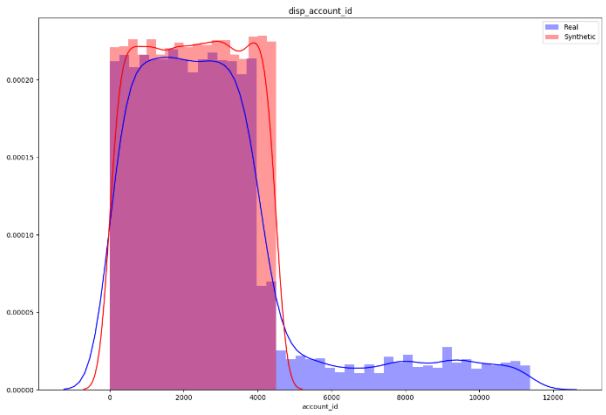


Figure 28 Table 'Disp' with column 'account_id'

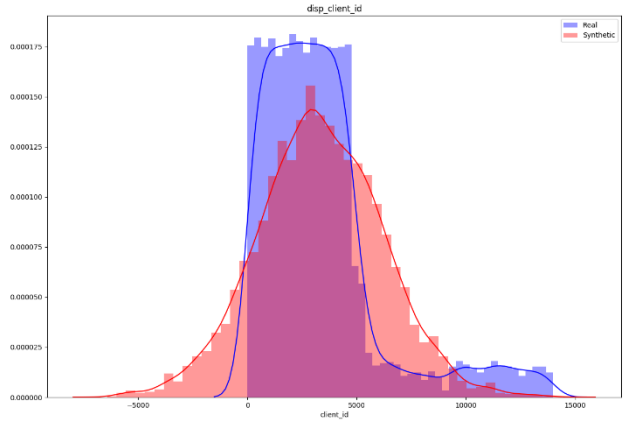


Figure 29 Table 'Disp with column 'client_id'

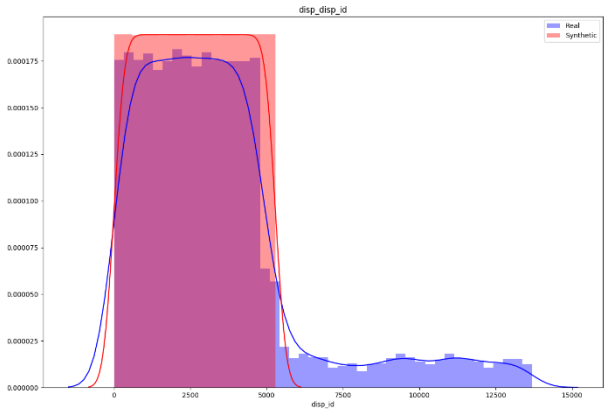


Figure 30 Table 'Disp' with column 'disp_id'

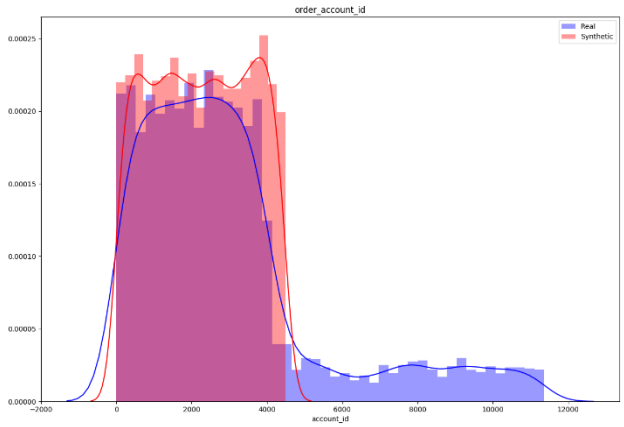


Figure 31 Table 'Order' with column 'account_id'

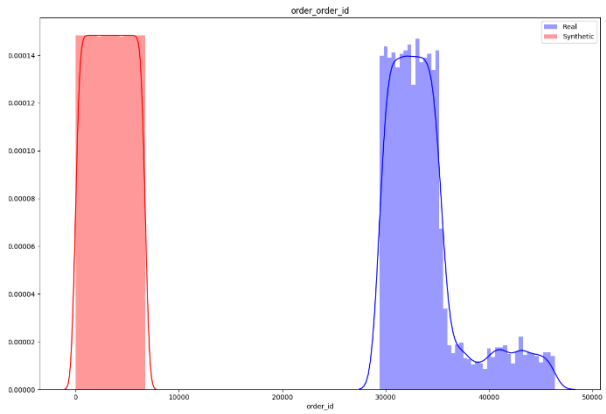


Figure 32 Table 'Order' with column 'order_id'

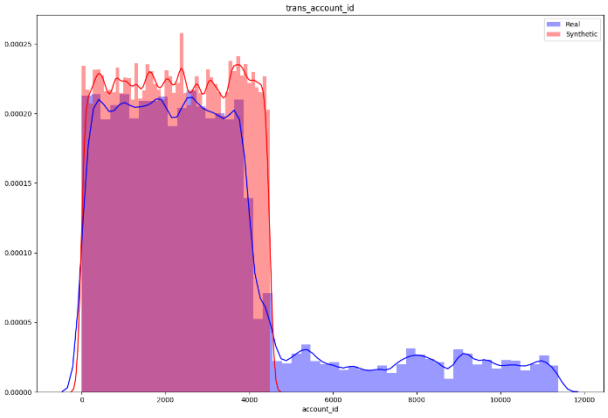


Figure 33 Table 'Trans' with column 'account_id'

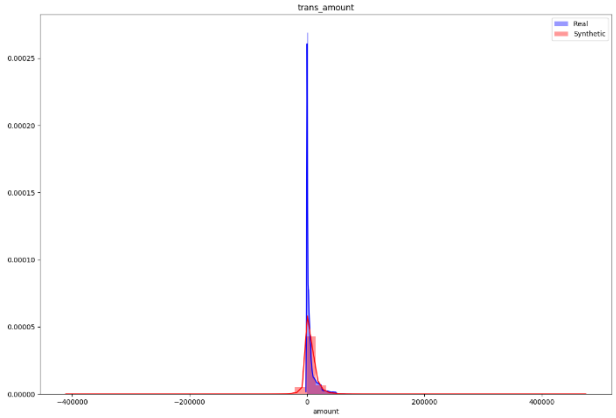


Figure 34 Table 'Trans' with column 'amount'

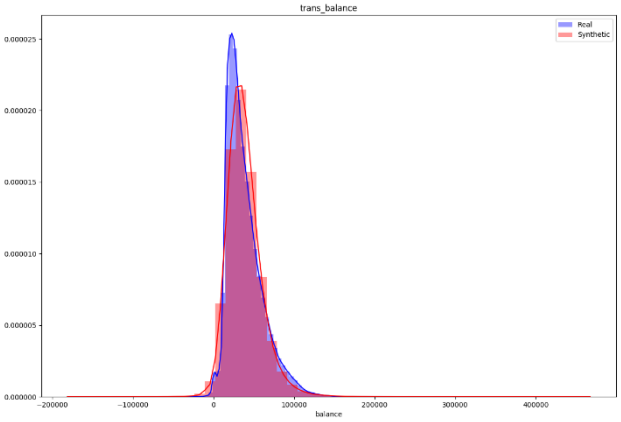


Figure 35 Table 'Trans' with column 'balance'

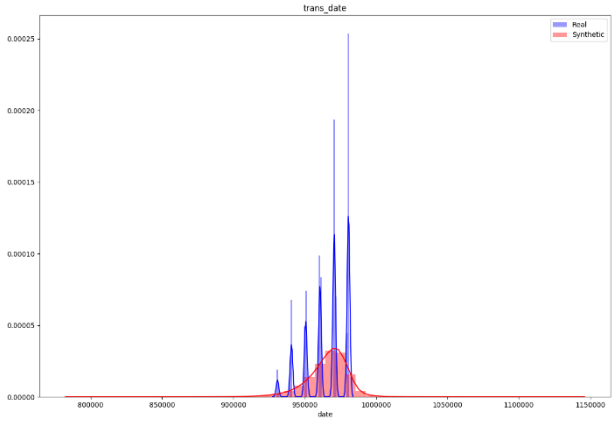


Figure 36 Table 'Trans' with column 'date'

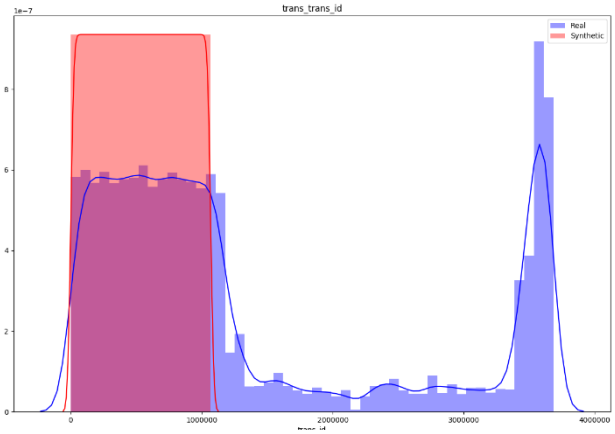


Figure 37 Table 'Trans' with column 'trans_id'

In order to establish how well SDV was able to identify relationships between columns, correlation heat maps were produced using the 'seaborn' package, which automatically calculates the Pearson correlation coefficient for each column to every other column in that table. The heat maps are symmetrical so only the values say below the diagonal need to be considered. The Pearson coefficients are stated in each cell in the heat map where a value of 1 indicates perfect correlation and 0 indicates none; this is also represented by the colour where the darker shade of blue indicates stronger correlation.

There was no correlation between the columns for the Account table (Figure 38) in the real set and that was also the case for the synthetic set, which is worth mentioning as some of the cases mentioned later found correlations that were not present.

The heat maps for the Disp table (Figure 39) show that the real set had correlation between all the columns. The synthetic table was not able to reproduce the same mapping as the real, instead, it only showed perfect correlation between disp_id/account_id but not client_id/account_id or disp_id/client_id. However, as these parings all involve at least one primary or foreign key the correlations may not be a characteristic of the real dataset that needs to be preserved.

The correlations for the Order table (Figure 40) show a good match between the real and synthetic datasets. The synthetic dataset contained strong correlations that are present in the real dataset, however there are very slightly higher correlations than expected for amount/account_id and order_id/amount.

The Trans table (Figure 41) was the most sophisticated out of these, containing the most columns and interesting correlations between numerical fields balance and amount. The real dataset shows a much stronger correlation between balance and amount than is shown in the synthetic data, suggesting that SDV has struggled to preserve the correlation. Intuitively, a correlation between balance and amount is expected, as those with a low balance would tend to have smaller valued transactions while larger valued transactions would be only tend to occur when the funds are available (i.e. account balance is large). In comparison, the trans_id/account_id showed much stronger correlation in the synthetic than was actually present. However, as these columns are keys (primary and foreign) the correlation may not be meaningful in the real dataset and the increase

seen in the synthetic data is likely to be a feature of the way SDV adds data to the table in a sequential way – it appears to add all transactions for account 0, then all transactions for account 1 and so on. The synthetic data also contained slightly stronger correlation between date/amount than was present in the real set.

Overall, it seems there were mixed results in detecting correlation between columns where they were either detected and simulated well, not detected at all or wrongly identified. It is likely that the correlations detected are dependent on whether the underlying variables can be represented by a single multi-variate Gaussian distribution.

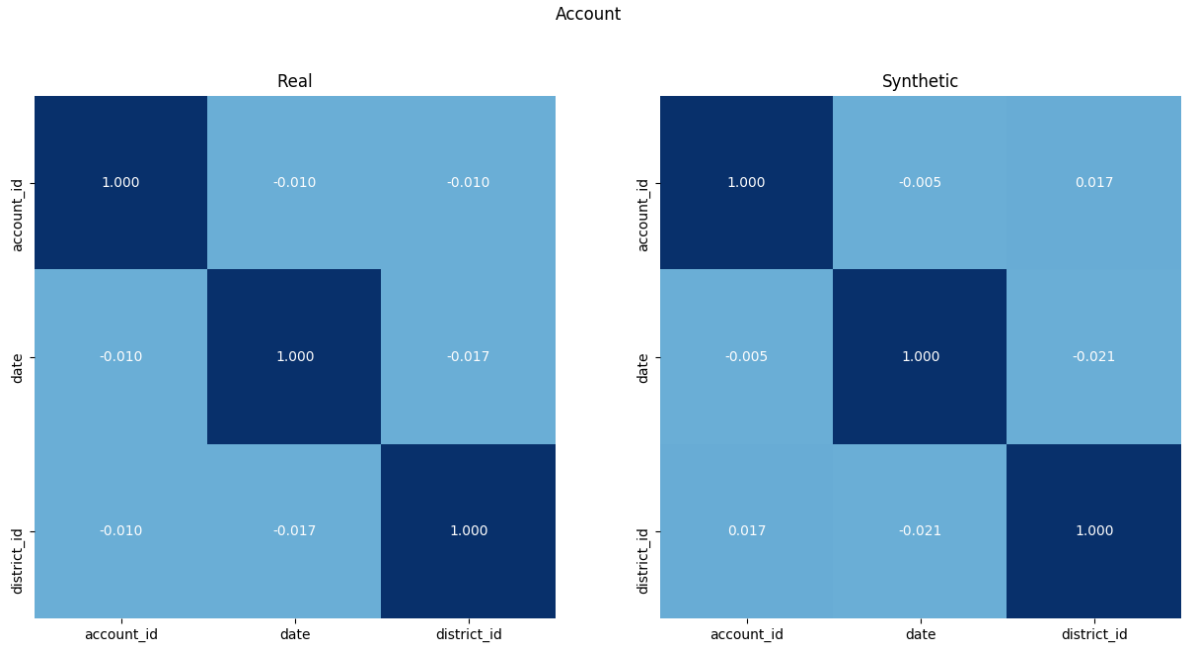


Figure 38 Heat maps for the Account table for the real (left) and synthetic (right) data

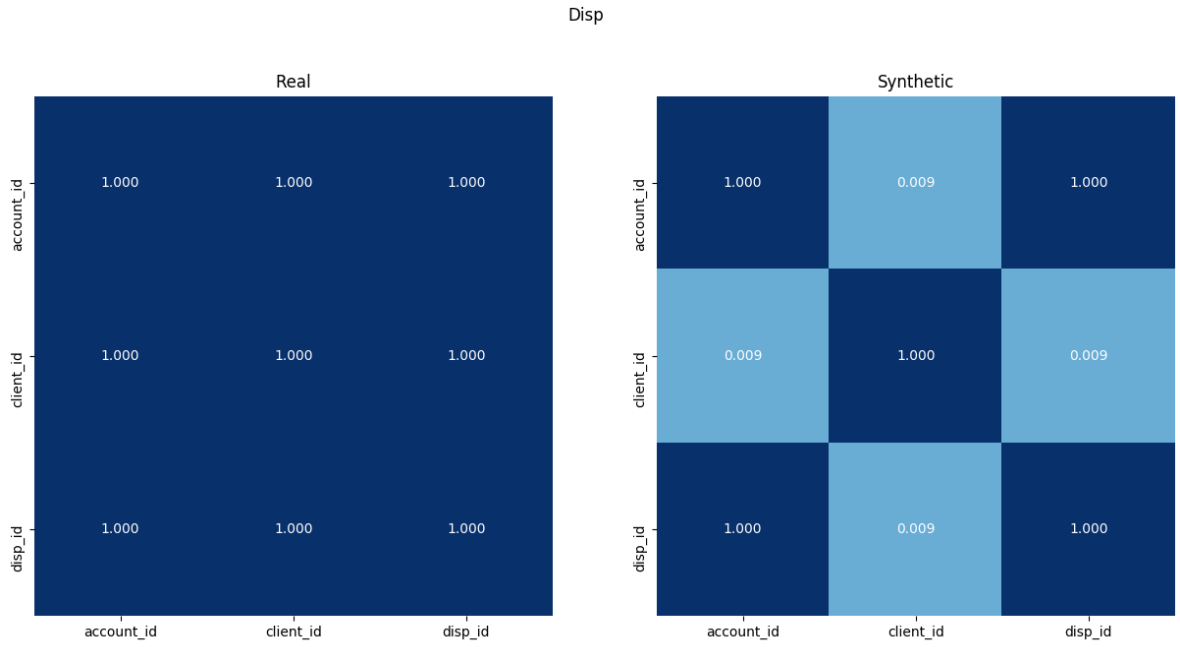


Figure 39 Heat maps for the Disp table for the real (left) and synthetic (right) data

Figure 40 Heat maps for the Order table for the real (left) and synthetic (right) data

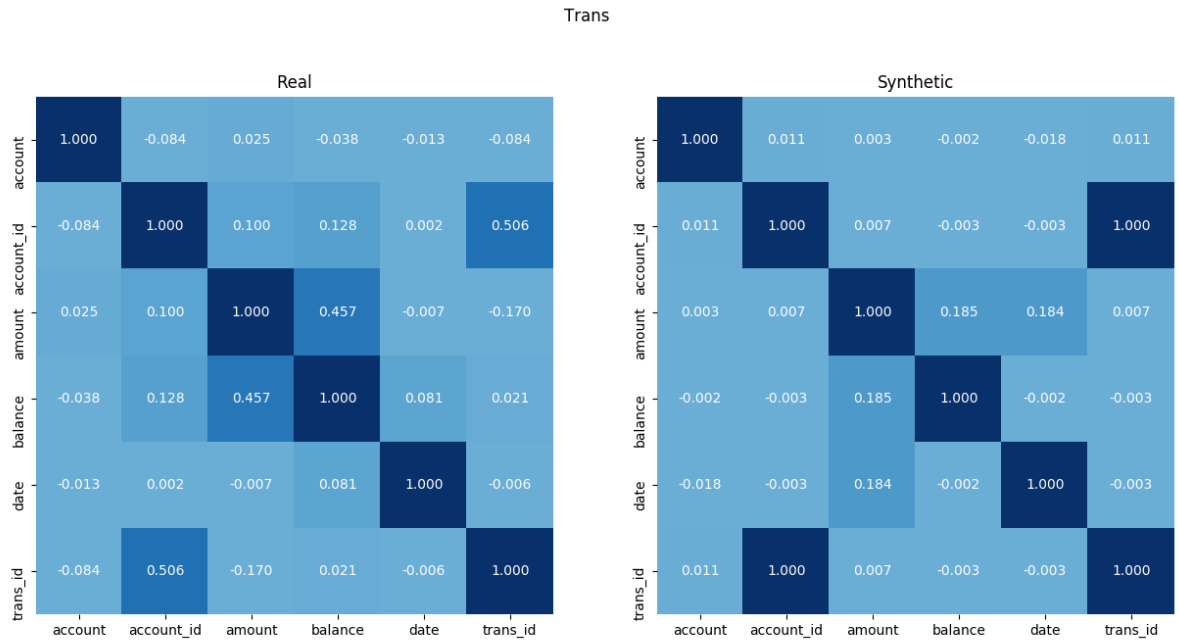


Figure 41 Heat maps for the Trans table for the real (left) and synthetic (right) data

5.3.5 Summary

A summary of the findings are provided here:

- SDV creates tables that contains primary keys that start at 0 and then incrementally increase as data is added. This seams sensible, but it is not clear how well it is coping when the input data does not follow this pattern. Further work should be performed to assess this.
- For the numerical fields SDV has struggled to match the shape and details of the distributions as it is currently limited to Gaussian Distributions. As it is unable to truncate the distributions the synthetic data can include data outside the range of the real data and/or errors that do not appear in the real data (e.g. negative values)
- Correlations between numerical data are preserved to some extent, but when there was a stronger correlation in the real data this was not replicated as strongly in the synthetic data.

5.4 Conclusions

We have evaluated three chosen tools against three datasets. Below is some general observations and thoughts, with an emphasis on mimicing techniques:

- A selection of techniques may need to be used in combination to obscure sensitive data. e.g. Random generation of names along with values generated to mimic reference data.
- Pre-processing of reference data may be required to transform data into a more appropriate form for a chosen technique.

- Strategies are required to address artefacts found in real data sets where these disrupt the performance of the chosen technique. e.g. missing/null fields.

Table 21 lists observations about each technique and their application to the respective dataset.

Tool/Technique	Dataset	Observation
DP-auto-GAN (mimic)	Los Angeles Crime (selected columns)	• Demonstrated some ability to replicate distributions and feature wise correlations. • One-hot encoding of categories consumes significant memory resources when categories are large. • GANs can be difficult to setup and train well. • Shows promise for this type of data. • Current software aimed at replicating experiments would require development for more general application.
SDV (mimic)	Czech Financial	• Able to replicate the distribution of some numerical features. • Replication of numerical distributions is currently limited and therefore applicability is limited to compliant reference data. • Does not keep generated data within bounds of reference data. e.g. an Age feature may be given negative values. • More work required to understand how well database relationships are preserved. • Watch for future development
Presidio (Redaction)	Resumes (subset)	• It comprises a number of technologies to implement a processing pipeline. The analysis engine may be extracted and used separately if the complete system is not appropriate. • Rules based detection commonly applied to this role. Success depends upon the quality of each individual rule. • ML based detection provides context aware entity detection. • High false alarm rates observed for some entity types, assumed to result from poorly developed rules. • Well aligned to this type of text data. Missed and inaccurate detections suggest this would only be suitable as an aid to manual redaction.

Table 21 Tool and data observation

6 Scenario Examples

This section of the report discusses how data obscuration techniques can be used to meet the objectives of scenarios provided by the Authority. The scenarios provided by the Authority includes:

- To protect personal information MOD has a legal obligation for
- To protect commercial sensitive information that could lead to financial damage or be exploited by a threat actor
- To protect strengths and weaknesses that a threat actor can exploit
- To generate pattern of life data for monitoring purpose without compromising personal data

The scenarios developed here are based on our learnings from the literature review and our experience gained in the tool development and testing, but the approaches discussed have not been verified. For the purpose of the discussion, the scenarios have been linked to two datasets identified in section 1.2.

6.1 Pattern of Life

6.1.1 Scenario

Information regarding daily ebbs and flows of people around a geographical area (such as a city) are useful for a number of reasons in defence. Here, the objectives to explore include:

1) Is it possible to create a synthetic pattern of life dataset that is not directly linkable to the original data but still represents trends in movement of people over time?
2) Can one-off and periodic events, if present, be identified in the original dataset and either preserved or represented in the synthetic dataset?

The Chicago Taxi Rides dataset has been selected as a representative dataset to aid this discussion. For reference a summary of the fields in the dataset are provide below. A detailed description of all the fields of this dataset is available under Section 2.1.3.

- Taxi info (Taxi id, company)
- Fare info (Trip duration, distance. Cost (fare, tips, tolls), payment type
- Date (start and end)
- Location (pick-up/drop-off lat/lon, pick-up/drop-off community area)

6.1.2 Discussion

Mimicking techniques are directly relevant to meeting objective 1) as they create datasets with similar characteristics and attempt to ensure that privacy is protected in the synthetic dataset. However, the taxi dataset has correlation between the rows in the form of the taxi id, which are not preserved in the mimicking techniques. Mimicking techniques treat rows/entries independently and for the taxi dataset such techniques are unlikely to generate sensible sequence of rides for an individual taxi. For example, this means that taxis may have multiple rides at the same time and the taxi pick-up location may not have a correlation with the previous drop-off location.

If the taxi information is removed, the data still represents people's movements and no longer has correlation between rows. In this case mimicking techniques should work well. The following points should be considered to improve mimicking:

- Pre-processing dates may increase likelihood of preserving temporal relationships. For example creating columns: day of week, hour of day
- Pre-processing location lat/lon into areas (or using community area) may improve relationships in location
- Fields that can be calculated directly from other fields can be excluded from the mimicking process and then subsequently calculated. For example, total cost and perhaps distance and tolls (from a route calculated from pick-up/drop-off location)

To preserve a sensible sequence of rides per taxi a technique similar to 9.3.2.2 (Synthesising Plausible Privacy-Preserving Location Traces [61]) might be suitable. Instead of generating traces for people, traces could be generated for taxis. Converting the lat/lon into areas or using the community area should help in grouping locations and clustering the semantic classes. One-off traces (events) should be preserved provided they are present in the seed traces used to generate the synthetic traces.

6.2 Cloud computing

6.2.1 Scenario

Vehicles are being equipped with an increasing number of sensors to log performance data during routine operation. Analysing the data in a cloud environment could be an efficient way of enabling predictive maintenance. However, this may allow threat actors to infer performance characteristics and limitations of the equipment, which is clearly undesirable. The objective here to explore is:

1) Is it possible to create a dataset of reduced/altered detail (such as locations, times, event types, column headings) which obfuscates the context of the data but preserves, to a reasonable degree:
   a) The distributions of the different factors in the data and
   b) The relationships between them?

The Chicago crime dataset (see Section 2.1.1) can be used as an example.

6.2.2 Discussion

In terms of sensitive data there are two challenges in using cloud computing. The first, assuming the cloud is not secure, or at a higher risk of being compromised, relates to the model or analytics to perform the predictive maintenance. If a threat actor was able to obtain these models then they could infer performance characteristics or failure cases either directly from understanding the model or by passing data into the model and inspecting the results. The second is that a threat actor may be able to intercept data transfers between the equipment and the cloud, thus revealing characteristics and state of the equipment. Both of these can be mitigated by obscuring the input and output data of the model. An overview of this is provided in Figure 42. This shows how transformers can be used to remove private or sensitive information from the raw data before uploading it to the cloud. Transformers are also required to convert the obscured response back into the raw data so the machine or analyst can understand it. The techniques discussed earlier can be used to obscure the data, for example:

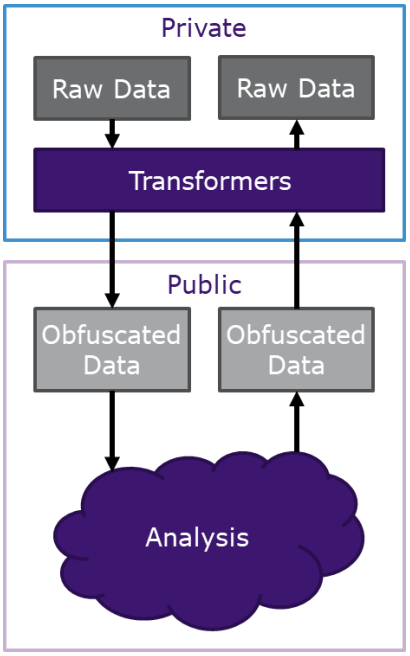


Figure 42 Cloud computing overview

- Replacing/Masking
  - Converting categorical names to numerical numbers, e.g. red, blue, yellow to 1,2,3.
  - Remove column names
  - Map location data to another location
- Coarsening
  - Reduce lat/lon to areas

An alternative to obscuring the data is to encrypt the data. Homomorphic encryption [62] allows calculations to be performed on encrypted data and returns an encrypted result without decrypting the data. There is an active area of research in training and using homomorphic encryption for machine learning purposes. This means that the data can be encrypted before sending it to the cloud, the analysis can be performed in encrypted form and the result (if any) can be return to the equipment or maintenance in encrypted form without ever decrypting the data on the cloud.

7 Conclusions and Recommendations

This report has identified and reviewed state of the art methods in obscuring sensitive information in datasets. Open source datasets have been identified and while not many of the available datasets are directly relevant to LTI they do contain similar characteristics, such as the range of data types.

We selected and assessed three tools/techniques. DP-auto-GAN aimed at mimicking categorical and numeric tabular data, SDV does the same for relational databases and Presidio, a tool for automated redaction of sensitive text. DP-auto-GAN is published to support replication of academic paper results, it provides a guarantee of differential privacy and some hint of what combined auto-encoder GAN based techniques may achieve in future development. SDV demonstrates some techniques of how mimicking can be applied to relational databases. Both mimic techniques have shown useful capability but are limited and not mature. GAN based techniques appear to offer the greatest capability but are more difficult to use and computationally expensive. Note also that these mimic techniques are aimed at protecting sensitive information in individual entries, however they also endeavour to preserve aggregate information which itself may be sensitive. Presidio is a more mature implementation of a complete system, but performance appears limited by its current rules set and does not seem adequate to avoid the need for manual intervention. However, the techniques used within its analysis engine may be useful in combination with more refined rule development.

Data obscuration is a very active area of research and driven primarily by dataset owners who wish to share or release their datasets in order to enhance research but need to protect the sensitive or personal information in their dataset. This is particularly evident in owners of medical records who need to protect the privacy of patients by ensuring records cannot be used to identify individuals. A range of techniques are being developed by researchers but what is clear from the literature review and the tool evaluation is that these techniques are specific to the particular needs being addressed and/or are immature. Therefore, they should not be used blindly on new datasets or for new requirements. It is vitally important that there is a clear understanding of the sensitive information to obscure and to what level. This could be by reducing the precision of the data to acceptable levels, ensuring individuals or specific pieces of equipment cannot be identified or by completely removing items from the data. It is also very important to ensure the resulting dataset still contains the necessary characteristics, such as statistical distributions and relationships, to ensure it can still be used for its particular requirements. There is clearly a conflict between obscuring sensitive information and maintaining the necessary characteristics. The metrics identified in this report help access these criteria, but they are predominantly qualitative rather quantitative metrics meaning that an expert is required to properly assess if the requirements have been met. Given these limitations in the state of the art, a black box data obscuration tool that can take in any dataset and automatically find and remove sensitive information is unlikely; however, developing specific tools for particular datasets and requirements is very possible and has been demonstrated in the literature.

Table 22 provides guidance on what techniques to use based on the datatype and requirements. It is likely that a combination of these will required for a single dataset.

Data Type	Requirements	Guidance
Numeric & Categorical	Obscure values, ranges etc. of specific fields	Redaction, masking and coarsening can be used to obscure values within specific fields (e.g. performance data, such as top speed). Careful consideration is required to ensure a threat actor cannot reverse engineer the original values or determine sensitive relationships between columns that could be exploited.
	Mimic the dataset - preserving the characteristics of the dataset (e.g. statistical distribution and relationships)	Many statistical and GANs (e.g. DP-auto-GAN) techniques exist. Differential Privacy can limit the disclosure of private information. Summary of findings: • Missing values and errors - For categorical data, missing values can be treated as an additional category. Limited approaches for errors or missing data in numerical fields – typically, the data is removed. • Pre-processing the data before mimicking can improve results. For example, if there are trends in the dataset base on the hour of data, then create a new field from the timestamp that contains the hour of day. • Complex distributions and relationships are difficult for the tools to mimic – try to simplify the dataset by pre-processing where possible. • Categories with a large number buckets are more challenging to accurately mimic Types of data: 1. **Uncorrelated rows** - Many statistical and GANs (e.g. DP-auto-GAN) techniques exist 2. **Correlated rows -** Limited support at present. Recurrent GANs have been used on real valued time series data. 3. **Relational Datasets -** Synthetic Data Vault (SDV) is the only method that directly supports relational datasets. Flattening the data and processing it as single dataset is an option.
Text	Obscure values	A range of NLP based tools (e.g. Presidio) exists that automatically identify sensitive information in text documents. Text can then be redacted, masked or encrypted.
Location	Real time obscuration	Large area of research protecting the privacy of user when using location base services. Approaches are to anonymise, obfuscate (e.g. small offsets to location) or encryption
	Consistent Tracks	Privacy preserving traces simulate consistent tracks that match the behaviour of the user.

Table 22 Guidance on using data obscuration techniques

7.1 Recommendations for Future Work

As mentioned above, building a fully functional data obscuration tool that can handle a range of inputs, data obscuration requirements and synthetic dataset usages is not realistic at this time. Instead, the focus should be on developing solutions for specific scenario datasets and requirements, and where possible building generalised components that can be plugged together and extended for use with new datasets and requirements.

Synthetic data techniques for privacy is an active area of research, especially for mimicking techniques for numeric and categorical data, however challenges exist to accurately represent distributions in the data and to handle missing or null values in numeric data. Statistical approaches are still being explored and extended, for example to handle relational data, but GAN based techniques are the more active area of research. GANs may have the potential to represent more complex distributions and relationships than basic statistical methods but they can be difficult to train and training computation requirements and training time can be significant. If the privacy provided by differential privacy is appropriate for the specific scenario requirements, then mimicking techniques should be explored and matured in future work by understanding what pre-processing steps should be performed to improve results, developing approaches to handle errors and missing values, and maturing the GAN training process to make it more reliable.

Metrics used in the literature to evaluate how similar the synthetic dataset is to the real dataset provide broad qualitative measures that assess the dataset or field as a whole. Metrics that provide a more detailed analysis should be developed, such as those that determine if unusual behaviours or edge cases in the real data are represented in the synthetic data.

This project has provided a wide-ranging review of the techniques in the literature and started to understand the capabilities of a subset of these techniques. The focus of future work should be on developing solutions for a few detailed scenarios that specify the characteristics in the data to preserve and the sensitive information to be obscured.

8 Abbreviations & Definitions

ASC	Analysis Support Construct
LTI	Logistics Technology Investigations
GAN	Generative Adversarial Network
VAE	Variational Autoencoder
DP	Differential Privacy
API	Application Programming Interface
SDV	Synthetic Data Vault

9 References

[1] D. Knoors, 'Utility of Differentially Private Synthetic Data Generation for High-Dimensional Databases', Dissertation, 2018

[2] Wang, J., Liu, S. and Li, Y., 2015. A review of differential privacy in individual data release. International Journal of Distributed Sensor Networks, 11(10), p.259682.

[3] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy", Foundations and Trends in Theoretical Computer Science, vol. 9, no. 3, pp. 211–407, 2013

[4] Xie, L., Lin, K., Wang, S., Wang, F. and Zhou, J., 2018. Differentially private generative adversarial network. arXiv preprint arXiv:1802.06739.

[5] Jordon, J., Yoon, J. and van der Schaar, M., 2018. PATE-GAN: Generating synthetic data with differential privacy guarantees.

[6] Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W.F. and Sun, J., 2017. Generating multi-label discrete patient records using generative adversarial networks. arXiv preprint arXiv:1703.06490.

[7] Dandekar, A., Zen, R.A. and Bressan, S., Comparative Evaluation of Synthetic Data Generation Methods.

[8] Alan F Karr, Christine N Kohnen, Anna Oganian, Jerome P Reiter, and Ashish P Sanil. 2006. A framework for evaluating the utility of data altered to protect confidentiality. The American Statistician 60, 3 (2006), 224–232.

[9] Tantipongpipat, U., Waites, C., Boob, D., Siva, A.A. and Cummings, R., 2019. Differentially Private Mixed-Type Data Generation For Unsupervised Learning. arXiv preprint arXiv:1912.03250.

[10] NIST Contest: NIST Differential Privacy, NistDp1https://community.topcoder.com /longcontest/?module=ViewProblemStatement&compid=71282&rd=17319

[11] Page, H., Cabot, C. and Nissim, K., 2018. Differential privacy an introduction for statistical agencies. *NSQR. Government Statistical Service*.

[12] Bolón-Canedo, V., Sánchez-Maroño, N. & Alonso-Betanzos, A. A review of feature selection methods on synthetic data. Knowl Inf Syst 34, 483–519 (2013)

[13] Kaloskampis, I., Pugh, D., Joshi, C. and Nolan, L. (2020). *Synthetic data for public good | Data Science Campus*. [online] Datasciencecampus.ons.gov.uk. Available at: https://datasciencecampus.ons.gov.uk/projects/synthetic-data-for-public-good/ [Accessed 6 Feb. 2020].

[14] Doersch, C., 2016. Tutorial on variational autoencoders. arXiv preprint *arXiv:1606.05908.*

[15] Wan, Z., Zhang, Y. and He, H., 2017, November. Variational autoencoder based synthetic data generation for imbalanced learning. In 2017 IEEE symposium series on computational intelligence (SSCI) (pp. 1-7). IEEE.

[16] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets. *In Advances in neural information processing systems* (pp. 2672-2680).

[17] Salim Jr, A., 2018. Synthetic Patient Generation: A Deep Learning Approach Using Variational Autoencoders. arXiv preprint arXiv:1808.06444.

[18] Xie, L., Lin, K., Wang, S., Wang, F. and Zhou, J., 2018. Differentially private generative adversarial network. arXiv preprint arXiv:1802.06739.

[19] GitHub. (2020). *illidanlab/dpgan*. [online] Available at: https://github.com/illidanlab/dpgan [Accessed 6 Feb. 2020].

[20] Arjovsky, M., Chintala, S. and Bottou, L., 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875*.

[21] Fedus, W., Goodfellow, I. and Dai, A.M., 2018. MaskGAN: better text generation via filling in the_. *arXiv preprint arXiv:1801.07736*.

[22] GitHub. (2020). *models/maskgan*. [online] Available at: https://github.com/tensorflow/models/tree/master/research/maskgan [Accessed 6 Feb. 2020].

[23] Samarati, P. and Sweeney, L., 1998. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression.

[24] Park, N., Mohammadi, M., Gorde, K., Jajodia, S., Park, H. and Kim, Y., 2018. Data synthesis based on generative adversarial networks. *Proceedings of the VLDB Endowment*, *11*(10), pp.1071-1083.

[25] GitHub. (2020). *mahmoodm2/tableGAN*. [online] Available at: https://github.com/mahmoodm2/tableGAN [Accessed 6 Feb. 2020].

[26] Xu, L., Skoularidou, M., Cuesta-Infante, A. and Veeramachaneni, K., 2019. Modeling tabular data using conditional gan. In *Advances in Neural Information Processing Systems* (pp. 7333-7343).

[27] GitHub. (2020). *SDV-dev/CTGAN*. [online] Available at: https://github.com/SDV-dev/CTGAN [Accessed 6 Feb. 2020].

[28] Beaulieu-Jones, B.K., Wu, Z.S., Williams, C. and Green, C.S., 2018. Privacy-preserving generative deep neural networks support clinical data sharing. bioRxiv preprint first posted online Jul. 5, 2017; doi: http://dx. doi. org/10.1101/159756. *Accessed January*, *29*.

[29] GitHub. (2020). greenelab/SPRINT_gan. [online] Available at: https://github.com/greenelab/SPRINT_gan [Accessed 6 Feb. 2020].

[30] Joshi, C., Kaloskampis, I. and Nolan, L. (2020). *Generative adversarial networks (GANs) for synthetic dataset generation with binary classes | Data Science Campus*. [online] Datasciencecampus.ons.gov.uk. Available at: https://datasciencecampus.ons.gov.uk/projects/generative-adversarial-networks-gans-for-synthetic-dataset-generation-with-binary-classes/ [Accessed 6 Feb. 2020].

[31] Arjovsky, M., Chintala, S. and Bottou, L., 2017. Wasserstein gan. arXiv preprint arXiv:1701.07875.

[32] Mirza, M. and Osindero, S., 2014. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

[33] Fabri, C. 2018. Conditional Wasserstein Generative Adversarial Networks. [ebook] Minnesota. Available at: <https://cameronfabbri.github.io/papers/conditionalWGAN.pdf> [Accessed 6 April 2020].

[34] Dukler, Y., Li, W., Tong Lin, A. and Montúfar, G., 2019. Wasserstein of Wasserstein loss for learning generative models.

[35] Xu, C., Ren, J., Zhang, D., Zhang, Y., Qin, Z. and Ren, K., 2019. GANobfuscator: Mitigating information leakage under GAN via differential privacy. *IEEE Transactions on Information Forensics and Security*, *14*(9), pp.2358-2371.

[36] Long, Y., Lin, S., Yang, Z., Gunter, C.A. and Li, B., 2019. Scalable Differentially Private Generative Student Model via PATE. *arXiv preprint arXiv:1906.09338*.

[37] Jordon, J., Yoon, J. and van der Schaar, M., 2018. PATE-GAN: Generating synthetic data with differential privacy guarantees.

[38] Tantipongpipat, U., Waites, C., Boob, D., Siva, A.A. and Cummings, R., 2019. Differentially Private Mixed-Type Data Generation For Unsupervised Learning. *arXiv preprint arXiv:1912.03250*.

[39] GitHub. (2020). *DPautoGAN/DPautoGAN*. [online] Available at: https://github.com/DPautoGAN/DPautoGAN [Accessed 6 Feb. 2020].

[40] Cristóbal Esteban and Stephanie L. Hyland and Gunnar Rätsch, Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs, 2017, *arXiv preprint arXiv:1706.02633*

[41] Patki, N., Wedge, R. and Veeramachaneni, K., 2016, October. The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (pp. 399-410). IEEE.

[42] GitHub. (2020). *SDV-dev/SDV*. [online] Available at: https://github.com/HDI-Project/SDV [Accessed 6 Feb. 2020].

[43] Nowok, B., Raab, G.M. and Dibben, C., 2016. synthpop: Bespoke creation of synthetic data in R. *J Stat Softw*, *74*(11), pp.1-26.

[44] Cran.r-project.org. (2020). *CRAN - Package synthpop*. [online] Available at: https://cran.r-project.org/web/packages/synthpop/index.html [Accessed 6 Feb. 2020].

[45] Dahmen, J. and Cook, D., 2019. SynSys: A synthetic data generation system for healthcare applications. *Sensors*, *19*(5), p.1181.

[46] GitHub. (2020). *jb3dahmen/SynSys-Updated*. [online] Available at: https://github.com/jb3dahmen/SynSys-Updated [Accessed 6 Feb. 2020].

[47] Mostly.ai. (2020). *Mostly GENERATE*. [online] Available at: https://mostly.ai/mostly-generate.html [Accessed 6 Feb. 2020].

[48] MOSTLY AI - ENABLING PRIVACY-PRESERVING BIG DATA. (2020). [online] Available at: https://cdn2.hubspot.net/hubfs/4408323/Synthetic%20Data%20Engine%20-%20White%20Paper.pdf [Accessed 6 Feb. 2020].

[49] Rosette Text Analytics. (2020). *Relax, Your Sensitive Data Is Secure - Rosette Text Analytics*. [online] Available at: https://www.rosette.com/blog/relax-your-sensitive-data-is-secure/ [Accessed 6 Feb. 2020].

[50] SciBite. (2020). *TERMite | SciBite*. [online] Available at: https://www.scibite.com/platform/termite/ [Accessed 6 Feb. 2020].

[51] GitHub. (2020). *rosette-api-community/identity-masker*. [online] Available at: https://github.com/rosette-api-community/identity-masker [Accessed 6 Feb. 2020].

[52] GitHub. (2020). dstl/baleen. [online] Available at: https://github.com/dstl/baleen [Accessed 6 Feb. 2020].

[53] GitHub. (2020). *BitCurator/bitcurator-redact-pdf*. [online] Available at: https://github.com/BitCurator/bitcurator-redact-pdf [Accessed 6 Feb. 2020].

[54] GitHub. (2020). *microsoft/presidio*. [online] Available at: https://github.com/microsoft/presidio [Accessed 6 Feb. 2020].

[55] OpenText. (2020). *OpenText Redact-It*. [online] Available at: https://www.opentext.co.uk/products-and-solutions/products/enterprise-content-management/content-centric-applications/opentext-redact-it [Accessed 6 Feb. 2020].

[56] M. K. Tefera, X. Yang and Q. T. Sun, "A Survey of System Architectures, Privacy Preservation, and Main Research Challenges on Location-Based Services," KSII Transactions on Internet and Information Systems, vol. 13, no. 6, pp. 3199-3218, 2019. DOI: 10.3837/tiis.2019.06.024.

[57] https://en.wikipedia.org/wiki/K-anonymity

[58] P. Zhao, J. Li, F. Zeng, F. Xiao, C. Wang and H. Jiang, "ILLIA: Enabling k -Anonymity-Based Privacy Preserving Against Location Injection Attacks in Continuous LBS Queries," in IEEE Internet of Things Journal, vol. 5, no. 2, pp. 1033-1042, April 2018.

[59] P. Wightman, W. Coronell, D. Jabba, M. Jimeno and M. Labrador, "Evaluation of Location Obfuscation techniques for privacy in location based information systems," 2011 IEEE Third Latin-American Conference on Communications, Belem do Para, 2011, pp. 1-6.

[60] M. Duckham, L. Kulik and A. Birtley, "A Formal Model of Obfuscation and Negotiation for Location Privacy." In Proc. Pervasive 2005. LCNC 3468/2005, pp. 243-251, 2005.

[61] V. Bindschaedler and R. Shokri, "Synthesizing Plausible Privacy-Preserving Location Traces," *2016 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, 2016, pp. 546-563.

[62] https://en.wikipedia.org/wiki/Homomorphic_encryption

[63] https://www.anaconda.com/, Anaconda website accessed March 2020.

A Appendix A Datasets

Dataset	Column details	File format	License	Types of data
US Patent Grant Full Text https://www.kaggle.com/uspto/patent-grant-full-text	Patent number Series code and application number type of patent, filing date, title, issue date, applicant information, inventor information, assignee(s) at time of issue, foreign priority information, related US patent documents, classification information (IPCR, CPC, US), US and foreign references, attorney, agent or firm/legal representative, examiner, citations, Patent Cooperation Treaty (PCT) information, abstract, specification, and claims.	File format: XML Size: 569 MB	Public Domain Mark 1.0	Integers, dates, categorical, free text
Resume Entities for NER https://www.kaggle.com/dataturks/resume-entities-for-ner	Name, College Name, Degree, Graduation Year, Years of Experience, Companies worked at, Designation, Skills, Location, Email Address	File format: JSON	Unknown	Dates, free text
Mock NHS health data https://github.com/theodi/synthetic-data-tutorial	Health Service ID, Age, Time in A&E (mins), Hospital, Arrival Time, Treatment, Gender, Postcode	File format: CSV Size: 1 MB	MIT	Dates, integers, times, Booleans, free text

New York City bus data https://www.kaggle.com/stoney71/new-york-city-transport-statistics	RecordedAtTime DirectionRef PublishedLineName OriginName, OriginLat, OriginLong, DestinationName, DestinationLat, DestinationLong, VehicleRef, VehicleLocation.Latitude, VehicleLocation.Longitude, NextStopPointName, ArrivalProximityText, DistanceFromStop, ExpectedArrivalTime, ScheduledArrivalTime.	File format: CSV Size: 5GB	Unknown	Dates, time, floats, free text, categorical
New York City Airbnb Open Data https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data	id-listing ID name-name of the listing host_id- host ID host_name- name of the host neighbourhood_group- location neighbourhood- area latitude latitude- coordinates longitude longitude- coordinates room_type- listing space type price- price in dollars minimum_nights- amount of nights minimum number_of_reviews- number of reviews last_review- latest review reviews_per_month- number of reviews per month calculated_host_listings_count- amount of listing per host availability_365- number of days when listing is available for booking	File format: CSV Size: 7 MB	CC0: Public Domain	Dates, times, free text, categorical, floats, integers, some gaps

Crime in Los Angeles https://www.kaggle.com/cityofLA/crime-in-los-angeles	DR Number Date Reported Date Occurred Time Occurred Area ID Area Name Reporting District Crime Code Crime Code Description MO Codes Victim Age Victim Sex Victim Descent Premise Code Premise Description Weapon Used Code Weapon Description Status Code Status Description Crime Code 1 Crime Code 2 Crime Code 3 Crime Code 4 Address Cross Street Location	File format: CSV Size: 360 MB	CC0: Public Domain	Dates, times, free text, Booleans, categorical

Missing Migrants Project https://www.kaggle.com/snocco/missing-migrants-project	Region of Incident Reported Date Reported Year Reported Month Number Dead Minimum Estimated Number of Missing Total Dead and Missing Number of Survivors Number of Females Number of Males Number of Children Cause of Death Location Source Location Migration Route URL UNSD Geographical Grouping Source Quality	File format: CSV Size: 1.41 MB	Attribution 4.0 International (CC BY 4.0)	Dates, integers, floats, categorical
TWA Flight 800 FDR Dataset http://www.stat.ucla.edu/~rosario/datasets/twa800/tw800case.html	Second, Altitude, Airspeed, PitchAngle, ElevPosition, Heading, RollAngle, RudderPos, AngleAttack, Engine1, Engine2, Engine3, Engine4, LongAccel, VertAccel, PitchTrimStabPos, PressureAlt	File format: Rdata and CSV Size: 60 KB	Unknown	Integers, floats

Global terrorism database, 1970-2017 https://www.kaggle.com/START-UMD/gtd	Eventid year month day approxdate extended1 (duration of incident) resolution country region subnational administrative region city latitude longitude summary of attack etc (+ many more columns)	File format: CSV Size: 155 MB	University of Maryland (permission needed for commercial use)	Integers, floats, years, Booleans, categorical, dates, free text - contains gaps
Chicago crime https://www.kaggle.com/chicago/chicago-crime	unique_key, case_number, date, block, iucr code, primary_type, description, location_description, arrest, domesticbeatdistrictward, community_area, fbi_code, x_coordinate, y_coordinate, year, updated_on, latitude, longitude, location	File format: BigQuery, CSV, XML Size: 1.5 GB	CC0: Public Domain	Integers, floats, categorical, dates, free text

Chicago taxi rides https://www.kaggle.com/chicago/chicago-taxi-rides-2016	Taxi IDs, trip start and end timestamps, trip duration, trip distance, pickup and dropoff community areas, fares, tips, tolls, extras, trip total, payment type, company, pickup and dropoff longitudes/latitudes	File format: CSV Size: 2 GB	Public (by City of Chicago). Indemnity is onerous.	Dates/timestamps integers, floats, Booleans - data contains gaps
News articles and online reviews: -popular blog posts, news articles, negative and positive company reviews, negative and positive hotel reviews, negative and positive movie reviews etc https://webhose.io/free-datasets/	Time, name, location, date, url, organisation, text	File format: JSON Size: Each dataset is 130MB to 1GB	Unknown	Free text, dates, categorical

Amazon fine food reviews https://www.kaggle.com/snap/amazon-fine-food-reviews	ProductId UserId ProfileName HelpfulnessNumerator HelpfulnessDenominator Score Time Summary Text	File format: CSVs, SQLite Size: CSV (287 MB) or SQLite (642 MB)	CC0: Public Domain	Free text, integers
Amazon reviews https://nijianmo.github.io/amazon/index.html	Product ID, product title, product price, review user ID, review profile name, review helpfulness, review score, review time, review summary	File format: .txt.gz Size: 11 GB, but subsets available	Unknown	Free text, alphanumerical codes, integers, floats

Bike journeys https://www.lyft.com/bikes/bay-wheels/system-data	Trip Duration (seconds) Start Time and Date End Time and Date Start Station ID Start Station Name Start Station Latitude Start Station Longitude End Station ID End Station Name End Station Latitude End Station Longitude Bike ID User Type (Subscriber or Customer – "Subscriber" = Member or "Customer" = Casual) Member Year of Birth Member Gender	File format: CSVs Size: 112 MB	Bay Wheels License Agreement	Integers, floats, free text, categorical, dates, Booleans - contains gaps
US military base locations http://osav-usdot.opendata.arcgis.com/datasets/d163fcde26de4d21aa06aa141ce3a662_0	COMPONENT SITE_NAME JOINT_BASE STATE_TERR COUNTRY OPER_STAT PERIMETER AREA Shape_Length Shape_Area	File format: CSV, KML Size: 100 KB	Unknown	Categorical, integers, floats
US census data https://www.kaggle.com/muonneutrino/us-census-demographic-data	Census tract ID, state, county, total population, men, women, ethnic group percentages, percentages of poverty, incomes etc	File format: CSV Size: 29 MB	CC0: Public Domain	Integers, floats, categorical, free text

Amtrak stations in the US http://osav-usdot.opendata.arcgis.com/datasets/3e9daf681b154fb19372044f4d52941a_0	X Y OBJECTID STNCODE STNNAME CITY2 STATE STFIPS	File format: CSV, KML Size: 45 KB	'No access and use constraints'	Floats, categorical, integers
Resumes dataset https://github.com/JAIJANYANI/Automated-Resume-Screening-System	Name, contact number, email, work experience, qualifications, extra-curricular activities, awards/achievements, skills, other	File format: DOCX, PDFs Size: 3 MB	MIT License	Integers, free text
Stack Exchange data dump https://archive.org/details/stackexchange	Anonymized dump of all user-contributed content on the Stack Exchange network	File format: XML Size: 8 MB to 24.4 GB	CC BY-SA 4.0	Dates, free text
Bosch production line performance https://www.kaggle.com/c/bosch-production-line-performance/data	Production line, the station on the line, and a feature number, measurements of parts, time data	File format: CSV Size: 820 MB	Only for competition use	Dates, free text, integers, floats, Booleans

Air pressure failure system of Scania trucks https://www.kaggle.com/uciml/aps-failure-at-scania-trucks-data-set	Cost, component classes of sensor readings	File format: CSV Size: 178 MB	GPL 2	Integers, floats, contains gaps and n/a values
Relational Dataset Repository https://relational.fit.cvut.cz/	A large collection of relational datasets	Stored on a MariaDB. Can be exported (e.g. to CSV or SQL dump)	No license provided. Permission was granted for our purposes	Integers, floats, Boolean, Strings

Key	
Fits all requirements in terms of data types or license requested	
Fits most requirements in terms of data types or license requested	
Fits some requirements in terms of data types or license requested	
License unknown or contains few types of data	
License or data types do not fit requirements well	

B Appendix B Scoping Methods for Data Obscuration Methods

9.1 Overview

There are a wide range of data obscuration techniques, but typically they can be summarised as falling into the following categories:

- Redaction – completely removing data from the dataset
- Replacing/Masking – replacing parts of the dataset, e.g. using: hashing, encryption, tokenising or lookup datasets
- Coarsening – reducing the precision of the data, e.g. reducing number of decimal places in lat/lon, remove last 3 digits of postcode
- Mimicking – generate a dataset that closely matches the real dataset but does not contain exactly the same entries
- Simulation – generating part or all of the dataset that is similar in essential ways to the real data but is different with regard to sensitive information.

The actual techniques to perform these obscurations will vary depending on the type of dataset. This section elaborates on the tools mentioned in Section 3 and provides futher detail.

9.2 Mimicking Methods

9.2.1 Variational Autoencoders (VAEs)

9.2.1.1 VAE Use on an Imbalanced Image Dataset (December 2017) [15]

Figure 43 Original images from MNIST dataset (top) and the generated images (bottom)

In this paper the VAE method is applied to the MNIST dataset (Figure 43), which contains a large number of images showing handwritten digits between 0 and 9. An imbalanced dataset was created where the numbers 0-4 had 50 samples compared to 2500 for numbers 5-9. In this paper, the authors aimed to generate handwritten digits which looked similar to those in the MNIST dataset but were still different (in other words, of different handwriting).

The images produced (Figure 43), were found to be similar to the other members of the dataset whilst also being clear and sharp unlike traditional autoencoder methods. However, this method was focussed on producing accurate synthetic images, and had no privacy-based focus. This can be an issue since there was no guarantee in this method that the network was not memorising training samples and simply reproducing these.

9.2.1.2 VAE Use on Patient Records (August 2018)[17]

The authors experimented with a few approaches such as Gaussian mixture models and adversarial networks, and decided to use a VAE for their task at hand due to its simplicity and good performance in identifying relationships in large amounts of unlabelled data. The VAE developed in this paper aimed to mimic the statistical patterns present within a dataset containing a series of records of patient visits. The patterns being mimicked included those present in the following fields: patient symptoms, age, gender, time of year, differential diagnosis, tests ordered, test results, diagnosis and treatments. The mechanism involved encoding the input data into lower dimensional space (a Gaussian distribution density) and then decoding a sample from this distribution back to the original input.

The dataset contained a mixture of numerical and categorical information and a short training time (90 epochs) was sufficient when developing the model. This dataset covered many common ailments and how to deal with them but did not contains records of some rare conditions. This can be seen as an advantage as there is no need to learn rare illnesses that may not be present in a given hospital; however, this also means the model cannot necessarily be used in hospitals in other regions where there may be other distributions, unless retrained with a new dataset relevant to that area.

A subjective metric was used whereby the real patient data was firstly mixed with the synthetic data, and the medical doctors were then asked if they could identify which data points were synthetic. Some of the synthetic results produced are shown in Figure 44. Results showed that 20% of the synthetic data produced was identified as synthetic, 23.3% of the real data was identified as synthetic and 80% of the synthetic data was identified as real.

Malaria

Patient No.	Gender	Age (yrs)	Month	Symptoms
1	Female	39.1	April	body weakness, fever, headaches, vomiting
2	Male	47.0	April	body weakness, fever, headaches, joint pain
3	Female	33.5	February	body pain, fever, headaches
4	Male	84.4	April	body weakness, convulsion, fever, sleepiness
5	Female	29.8	February	fever, headaches

Figure 44 Table shows examples of synthetic data produced for malaria patients

9.2.2 Mimicking Methods - GAN Methods

9.2.2.1 Differentially Private GAN (DPGAN) (February 2018) [18]

Traditional GANs often learn distributions of training data points and therefore can end up "remembering" the training samples before reproducing them– to address this issue, DPGAN adds noise into the gradients during training which also provides a rigorous privacy guarantee. The Github code is available [19]

In the paper, DPGAN was trained on the MNIST dataset in order to produce synthetic images. Figure 45 shows the results for four different epsilons (ranging from low privacy (high epsilon) on the left to high privacy (low epsilon) on the right). Within each result, the left hand column shows the generated synthetic images and the remaining three columns show the nearest neighbours from the real dataset. The nearest neighbours illustrate that the generated data is not memorising the real data and that privacy is preserved. In effect, a differential privacy guarantee will aim to hide the handwriting of the numbers and as the MNIST dataset has a large collection of handwriting styles, this tool will aim to produce numbers which are in a handwriting style not present anywhere in this dataset.

The level of noise added during training is specified by changing the epsilon-factor which directly determines the the level of differential privacy in the generated dataset; the smaller the value of epsilon, the blurrier the images become (and the better the privacy protection due to more added noise). An infinite epsilon value would indicate that there is no differential privacy protection and the images would therefore be completely clear.

The report concluded by saying that the model generated 'high quality data points at a reasonable privacy level'. For instance, the synthetic images produced when epsilon=14.0 (shown in Figure 45) was considered to be a reasonably low epsilon value by the authors and produced a good balance between accuracy and privacy. The algorithm was also successful with another dataset containing numerical codes from medical records. In addition, the Wasserstein distance [20] was used as a metric and as a method of training the data; this showed that DPGAN was effective for both noisy and limited training data (when assuming a reasonably sized privacy budget). A drawback of DPGAN compared to other privacy-focussed GANs is that DPGAN does not adopt an efficient optimisation technique, preventing any improvement with the training stability or convergence speed. Despite DPGAN's strong privacy measures, it encounters a significant utility loss on the synthetic data when a large amount of it is produced or when the privacy budget is reduced by a certain amount.

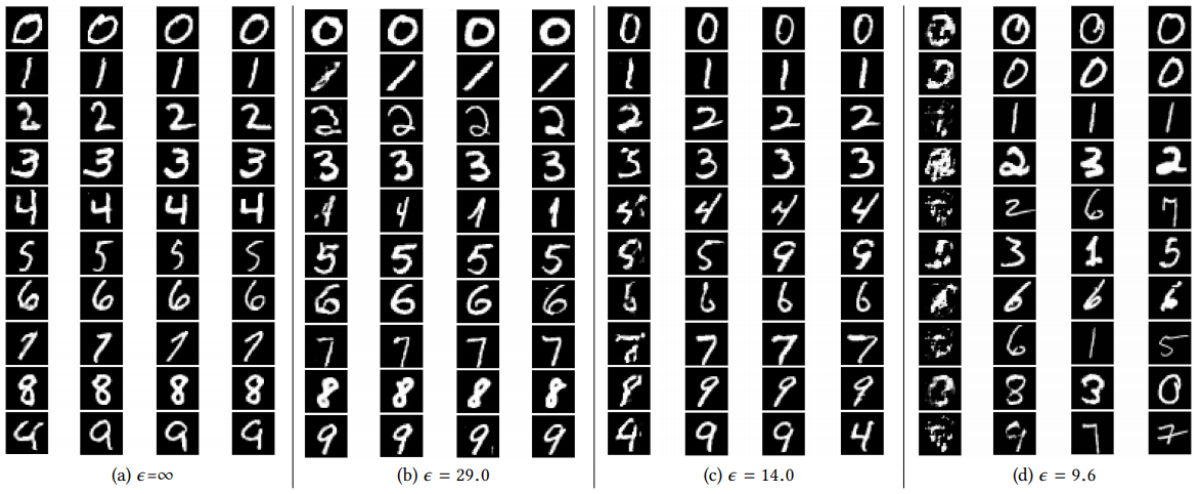


Figure 45 Generated images (left column) and the three nearest neighbours from the real dataset for four different epsilons on the MNIST dataset

9.2.2.2 MaskGAN (March 2018) [21]

The paper introduces a conditional GAN known as MaskGAN which fills in missing text based on its surrounding context. GANs have been used principally for images, they have not been used as

frequently for text generation due to difficulty arising from instability during training; in order to address this, the authors used reinforcement learning to train the generator. This GAN is focused on producing text and hence, does not work with numerical data. The Github code [19] is available [22] .

For more difficult/long sentences, the algorithm considers short segments at a time before increasing the length of segment considered in order to improve performance. To compare performance between MaskGAN, a maximum-likelihood trained model was used for comparison, which the paper referred to as MaskMLE [18].

The outputs generated from MaskGAN were assessed to be producing superior samples compared to MaskMLE when using the reviews from the IMDb dataset. To evaluate the results, assessors were selected using Amazon Mechanical Turk and were given 100 reviews (each 40 words long) for a pair of extracts sampled from three sets: MaskGAN, MaskMLE and real samples. They were asked whether if extract 1 was higher quality, extract 2 or neither between a given pair and were instructed to focus on grammar, topicality and overall quality.

They found that 44% preferred MaskGAN compared to the 18% that preferred MaskMLE and that 62% preferred real samples compared to 17% who preferred MaskGAN. Examples of outputs from MaskGAN, along with comparisons with MaskMLE, are shown in Figure 46.

Ground Truth	**the next day 's show <eos> interactive telephone technology has taken a new leap in <unk> and television programmers are**
MaskGAN	the next day 's show <eos> interactive telephone technology has taken a new leap in its retail business <eos> a
MaskMLE	the next day 's show <eos> interactive telephone technology has taken a new leap in the complicate case of the
Ground Truth	**Pitch Black was a complete shock to me when I first saw it back in 2000 In the previous years I**
MaskGAN	Pitch Black was a complete shock to me when I first saw it back in 1979 I was really looking forward
MaskMLE	Black was a complete shock to me when I first saw it back in 1969 I live in New Zealand

Figure 46 Table shows conditional samples of which the second one is from the IMDB dataset.

9.2.2.3 TableGAN (October 2018) [24]

TableGAN is an implementation of a Deep Convolutional Generative Adversarial Network (DCGAN) for comma separated value (CSV) style data with a well-known and widely adopted privacy model known as k-anonymity [23] being adopted. The authors decided to use GANs due to the flexibility offered in modelling distributions and wanted to produce synthetic data that was protected against information leakage.

The example datasets this was tested on contained both numerical and categorical values, with the datasets used for testing being characteristically similar and referred to as the LACity, Adult, Health and Airline datasets.

The proposed method was demonstrated to be effective against re-identification attack (where the obfuscated record is correctly linked with a person from the real dataset), membership attack (where the actual training data is determined) and attribute disclosure (where information is inferred depending on what values are shared between points). Generated records for the LACity dataset are shown in Figure 47, along with some of the rows from the original dataset.

Privacy was tested by using the 'distance to the closest record' along with performing membership attacks; this method was overall found to be effective in preserving privacy when using these metrics. The authors also claimed that this was the first method to use deep learning for general relational databases; these are a set of datasets, such as tables, which have defined relationships between them.

The paper wished to extend the method to other data types such as strings, and the datasets used along with its corresponding code [21] have been made available on GitHub [25]

Year	Salary	Q1	Q2	Q3	Dept	Job
2014	70386.48	16129.89	17829.78	17678.24	98	1230
2013	52450.56	11331	13859.93	11968.32	70	2214
2013	89303.76	20036.32	23479.2	21153.6	70	2214
2013	60028.96	15793.88	18560.38	16471.18	42	3184
2014	64553.13	14700	17313.1	15257.17	82	1368
2014	65959.92	26530.26	32978.41	25697.5	98	3181

Year	Salary	Q1	Q2	Q3	Dept	Job
2013	72005.93	11747.34	17186.00	19557.64	50	1451
2013	59747.90	4369.88	13377.60	22311.95	73	1248
2013	85600.46	17993.01	25420.13	27127.87	46	2025
2013	65156.87	11011.99	20201.47	23563.72	67	1887
2014	68638.75	9642.26	13674.69	15680.99	51	998
2014	73140.91	14474.15	28872.33	30307.91	71	2279

Figure 47 Example records from the original LACity dataset (left) and the synthesised table using tableGAN (right) using a low-privacy setting.

One of the drawbacks of tableGAN is that it can suffer from mode collapse (which is a common issue amongst many GANs) where the model stops learning as the discriminator becomes too successful, and so this results in only limited varieties of data points being produced. However, a new model was proposed for tabular data in October 2019 known as Conditional Tabular GAN (CTGAN) [26] which was shown to have better performance when you had many categories of data. The Github code is available for CTGAN [27]

9.2.2.4 AC-GAN for SPRINT Trial (December 2018) [28]

The Auxiliary Classifier GAN (AC-GAN) incorporates differential privacy measures, and was created to simulate participants of the SPRINT clinical trial. The parameters required to ensure a good balance of accuracy whilst maintaining a reasonable privacy budget were identified in this study for the patients they had. The code has been made available on Github [29]

The SPRINT trial data consisted of a table with values on systolic blood pressure, diastolic blood pressure and the number of medications for patients for over multiple study visits. A standard, non-private AC-GAN as well as a private AC-GAN (trained with differential privacy) were compared upon training with the data. Different epsilon values for differential privacy were chosen and results showed that a value of 2 was able to produce useful synthetic data. An epsilon value of 0 indicates maximum privacy protection and infinity indicates no privacy, so a value of 2 indicates a high level of privacy.

During the training process, the authors saved the models produced after each epoch and then selected the best models for further evaluation. The best models were determined by training a classifier (logistic regression or random forest) on the synthetic data produced by the model and

testing these models on the training set from the real dataset. The models that performed the best where selected. To evaluate whether the synthetic data generated from the best models was similar to the real data, correlations between each study (for systolic, diastolic blood pressure and medication count) were determined by using Pearson correlation. The correlation matrices between the real SPRINT data and the non-private GAN data were highly correlated with a coefficient of 0.96. Adding differential privacy with the private GAN produced a slightly lower correlation value of 0.89. In addition, three physicians were asked to determine whether the participants (of which 50 were real and 50 were from private GAN) looked real. The clinicians looked for any inconsistencies (e.g. where blood pressure was in the normal region, but medicine was prescribed anyway) and rated each record a "realism score" between 0 and 10. They found that the mean realism score for synthetic patients was 5.01 and for real patients was 5.16. This suggests that it was difficult to distinguish between the synthetic and real patients by the clinicians.

9.2.2.5 GANs for Synthetic Dataset Generation with Binary Classes (February 2019) [30]

In this article, US census data was used in order to predict whether a person earns over $50,000 or not. The dataset was not balanced with only 25% of the dataset having a class label of over $50,000. In addition, the other columns contained a mixture of discrete and continuous features including age, working class, education, marital status, race, sex, relationship and hours worked each week.

The GAN proposed learns the structure and distributions of the real dataset and then produces an augmented, synthetic dataset. First, the real dataset was split into a training and testing set. Classifiers were then trained on this synthetic data produced by the GAN (using the training set) to classify between the >50k and ≤50k classes. The classifiers were then subsequently tested on the real testing set and it was found an average of 80% was obtained for the prediction accuracy.

Four different GAN architectures – vanilla (normal) GAN[16] , Wasserstein GAN (WGAN)[31] , Conditional GAN (CGAN)[32] and Wasserstein Conditional GAN (WCGAN) [33] were compared (using the Wasserstein metric [34])in this study. . Wasserstein GANs (WGANs) displayed the optimum performance and the structure of this GAN type was able to minimise the occurrence of some of the common issues involved with training GANs, such as mode collapse.

The current work presented deals with having only two classes (≤$50k and >$50k) but datasets with more than two classes are now being investigated by the authors with suggestions for possible improvements:

- Modifying network hyperparameters by increasing the number of steps and hidden layers in the network
- Improving the quality of the GAN architecture by using approaches such as feature matching, minibatch discrimination, historical averaging, one-sided label smoothing, virtual batch normalisation and adding noise.

9.2.2.6 GANobfuscator (February 2019) [35]

This GAN uses the technique of bounding the gradients during training. This results in a GAN which is better at handling differential privacy with larger amounts of synthetic data generation. The labelled

MNIST dataset (handwritten digit images), LSUN (images of scenery) and CelebA (images of celebrities) were used to evaluate the model. The results are shown in Figure 48.

The paper provides proof of the privacy guarantee of the model, and the method incorporates privacy-enhancing mechanisms which improve the stability and scalability of the training procedure. In addition, membership attacks were used to demonstrate effectiveness against privacy attacks.

The paper discusses how some authors enforce differential privacy by injecting noise into objective functions of deep autoencoders at every layer and training step, though this approach can lack strict differential privacy guarantees. Conversely GANobfuscator incorporates noise more directly into the training procedure. This technique was found to be stable during training and did not suffer from mode collapse or gradient vanishing, which meant there was excellent scalability for model training, assuming a 'reasonable' privacy budget.

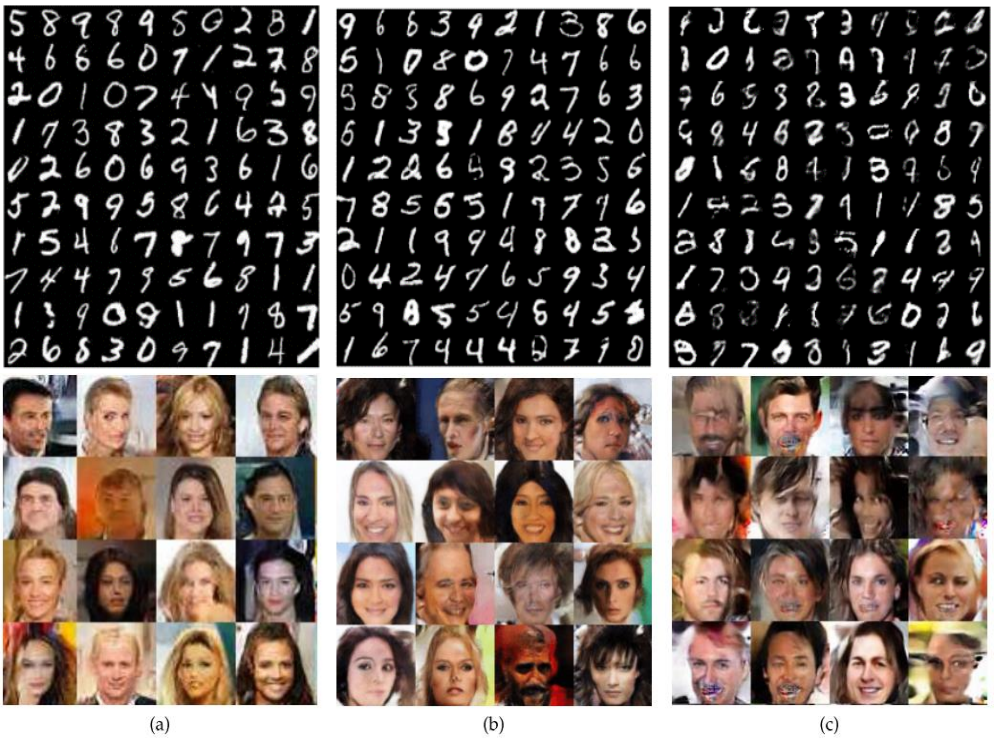


Figure 48 Synthetic samples produced for the MNIST dataset (top row) and CelebA (bottom row). Column (a) shows the results with the least privacy protection (epsilon=8), followed by Column (b) where epsilon=4, and then Column (c) which has the highest amount (epsilon=2).

GANobfuscator was found to be superior to DPGAN [19] and vanilla GAN [16] (with no privacy-based modifications) in the ability to protect against membership privacy attacks. The output synthetic images incorporated unrealistic details due to the noise added in in the model, which meant real images were not replicated.

9.2.2.7 G-PATE (June 2019) [36]

G-PATE, loosely based on the DPGAN architecture, trains using the Private Aggregation of Teacher Ensembles (PATE) framework. Here, a student discriminator is trained with an ensemble of teacher discriminators whereby the student discriminator is not allowed to see any of the real data. The

MNIST and Fashion-MNIST image datasets and the fully numerical tabular Credit Card Fraud Detection dataset were examined.

The paper observes that differential privacy needs to be guaranteed only on the information that flows from discriminator to generator, rather than the entire network. This means that the discriminator can be trained using the real data itself; hence, the privacy budget is improved (compared to DPGAN) whist incurring lower utility loss.

G-PATE showed better performance compared to vanilla GAN [16] , DPGAN [19] and PATE-GAN [37] in maintaining both a low privacy budget (i.e. high privacy protection) and low utility loss; DPGAN had also failed to converge on some of the more complex image datasets, unlike G-PATE. Even though the results shown in Figure 49 still look quite poor, partial features are still preserved from the real images so the synthetic images can still be useful. However, no further analysis or evaluation is conducted to prove this.

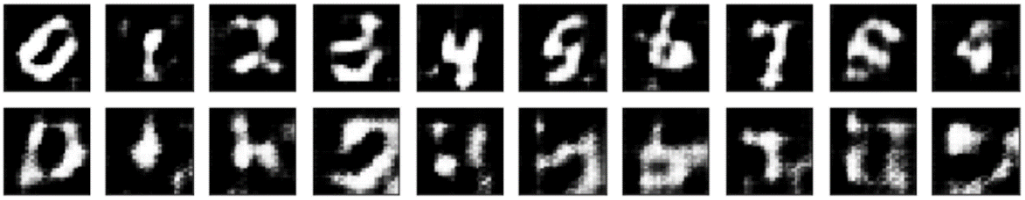


Figure 49 Synthetic samples produced with G-PATE with low privacy protection (top row) and high privacy protection (bottom row)

9.2.2.8 DP-auto-GAN (October 2019) [38]

The DP-auto-GAN model is applicable to mixed-type data which includes binary, categorical and real-valued features. In this paper, the MIMIC-III medical dataset and UCI ADULT Census dataset were used.

The results showed that the DP-auto-GAN algorithm had significantly better performance with small privacy budgets compared to prior work (such as that by Xie et al [6] and Choi et al [4] indicating a strong privacy guarantee. In other words, DP-auto-GAN was able to produce superior results with a high level of privacy protection compared to the aforementioned prior work, which often needed to sacrifice privacy protection in order to obtain comparable accuracies to that of DP-auto-GAN.

The GAN is incorporated into an autoencoder framework with gradients being clipped and Gaussian noise added during training. The autoencoder reduces dimensionality of the data before it is fed into the GAN with the noise only being added into the decoder part of the autoencoder (resulting in less noise being added unlike previous models, whilst maintaining the same privacy guarantee). The source code is available on Github [39] and this tool was explored in detail in Section 5

9.2.2.9 Real-valued (Medical) time series generation with recurrent conditional GANs [40]

This paper investigates the use of Recurrent GANs and recurrent conditional GANs to mimic real value multi-dimensional time series. The work is based on using the Phillips eICU dataset (https://eicu-crd.mit.edu/) and focusses on generating the regularly sampled real valued variables (e.g. heart rate) measured by monitors. As in previous mimicking methods discussed here, differential privacy is used to help ensure training data points are not represented in the output

dataset. Comparisons between the performance of a random forest classifier trained on the real and synthetic dataset is used to evaluate the synthetic dataset. The random forest classifier trained on the synthetic data and tested on the real data shows a small to moderate drop in performance compared to the classifier trained and tested on the real data.

9.2.3 Mimicking Methods - Statistical & Other Deep Learning Methods

9.2.3.1 Synthetic Data Vault (July 2018) [41]

The Python package 'SDV', as available on Github [42] is a statistical tool which can be used to build generative models of relational databases. It allows users to model an entire multi-table, relational dataset – users can then use the statistical model to generate a synthetic dataset. SDV can be used to model numerical, categorical and date/time data. For each column in each table, it will calculate covariances and standard deviations of the values, which are then used to model a distribution. The SDV python package comes with an example relational dataset, shown in Figure 50, which can be used as an input to the SDV tool.

In addition, SDV is automated such that it will identify the type of data in a column (e.g. numerical or categorical) and will perform any required pre-processing to deal with missing values. Metadata must also be defined by the user when the tables are loaded in. These may be specified through the use of primary and foreign keys.

The tool is currently limited to modelling a Gaussian distribution but the paper suggests a truncated Gaussian, uniform, Beta and Exponential distribution would also be possible. The paper ran SDV on five datasets: Biodegrability, Mutagenesis, Airbnb, Rossmann and Telstra. Synthetic tables were synthesised for each of these datasets with different versions: no noise, noise added into the covariance values and noise added into random primary-foreign key relations. Freelance data scientists were given the synthetic copies for a given data set, were told to write feature scripts to assess the accuracy of the predictions.

The authors found that for half of the analyses conducted, there was no significant difference between the real and synthetic dataset accuracy though the data scientists mentioned facing issues

such as unrealistic values for ages. The privacy concerns are dealt with by being able to control the level of noise input into the model, but no rigorous privacy guarantee proof provided otherwise. This tool is explored in greater depth in Section 5.

```
{
   'users':
          user_id country gender  age
        0        0     USA      M   34
        1        1      UK      F   23
        2        2      ES   None   44
        3        3      UK      M   22
        4        4     USA      F   54
        5        5      DE      M   57
        6        6      BG      F   45
        7        7      ES   None   41
        8        8      FR      F   23
        9        9      UK   None   30,
  'sessions':
        session_id  user_id  device      os
        0           0        0  mobile  android
        1           1        1  tablet      ios
        2           2        1  tablet  android
        3           3        2  mobile  android
        4           4        4  mobile      ios
        5           5        5  mobile  android
        6           6        6  mobile      ios
        7           7        6  tablet      ios
        8           8        6  mobile      ios
        9           9        8  tablet      ios,
  'transactions':
        transaction_id  session_id           timestamp  amount  approved
        0               0           0 2019-01-01 12:34:32   100.0      True
        1               1           0 2019-01-01 12:42:21    55.3      True
        2               2           1 2019-01-07 17:23:11    79.5      True
        3               3           3 2019-01-10 11:08:57   112.1     False
        4               4           5 2019-01-10 21:54:08   110.0     False
        5               5           5 2019-01-11 11:21:20    76.3      True
        6               6           7 2019-01-22 14:44:10    89.5      True
        7               7           8 2019-01-23 10:14:09   132.1     False
        8               8           9 2019-01-27 16:09:17    68.0      True
        9               9           9 2019-01-29 12:10:48    99.9      True
}
```

Figure 50 Example input relational dataset consisting of 3 tables which can be used by the SDV model

9.2.3.2 SynthPop (December 2018) [43]

This R package, which is readily available [44] uses a combination of hidden Markov models and regression algorithms to produce synthetic data and works with numerical, categorical and Boolean values enclosed in a CSV file. However, although the paper mentions that the tool can be used to produce synthetic versions of data containing confidential information, no rigorous differential privacy method or proof was provided or tested.

9.2.3.3 SynSys (March 2019) [45]

This model uses hidden Markov models initially trained on real datasets. A time-series distance measure was used to determine how realistic the synthetic dataset was. SynSys is used to generate time series data composed of nested sequences using hidden Markov models and regression models, which are initially trained on real smart home data. This data consists information on date and time, sensor name, sensor reading and activity performed. This is represented as date-time, categorical and Boolean data. Github code is available [46]

When comparing the time-series distance measure, SynSys generates more realistic data for a particular home with a smart home device, compared to just using real data collected from another home, data from another time period or random data points. This makes sense as the patterns found within one home in their smart home device usage will differ to that of another home, or the time of day the device was used. The distance metric found that the synthetic data produced by SynSys had the shortest Euclidean distance compared with the real distance (and hence was most similar) whereas the distance value between randomly generated data and the real data was largest, as one would expect.

The synthetic data when combined with the real dataset was used to test whether activity recognition accuracy would improve (with a semi-supervised learning algorithm) and the results showed that an improvement of around 10% was obtained when training using SynSys real and synthetic data combination, compared to just real data It should be noted that SynSys was not focussed on ensuring a privacy guarantee, and only on accurate synthetic data generation.

9.2.3.4 Mostly GENERATE (September 2019) [47]

This is a commercial tool available to produce synthetic data by using generative deep neural networks, with a white paper available describing the project [48] . The tool has been awarded the European ePrivacy Seal and a free demo version is available for trial, though this version is limited to CSVs with a maximum of 500 rows and 50 columns.

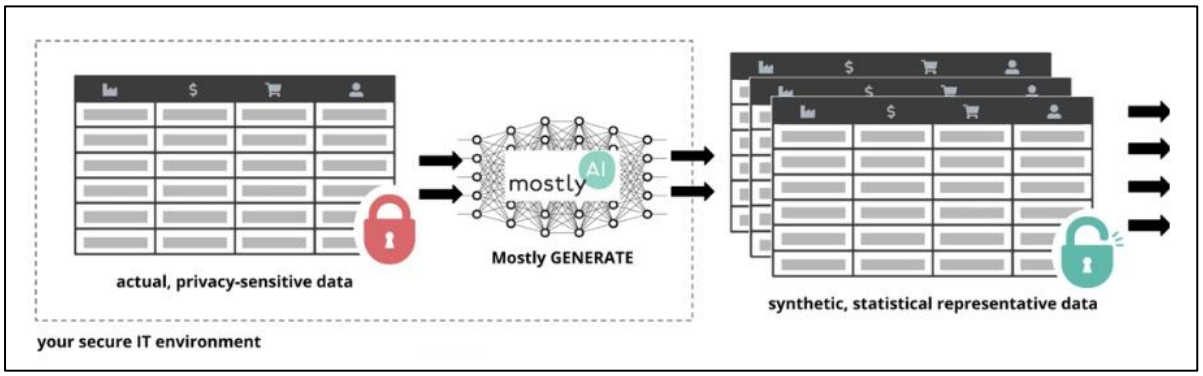


Figure 51 Mostly GENERATE Overview

9.3 Redaction, Replacing/Masking, Coarsening and Simulation

9.3.1 Text-based Methods

9.3.1.1 Rosette Text (April 2019) [49] and SciBite TERMite (May 2018) [50]

Rosette Text is a commercial tool in which a Rosette Cloud script is able to mask identities in order to redact personally identifiable information from documents. It is a regex-based tool and has been used by large organisations such as Microsoft, Amazon, Oracle and Airbnb.

For example, the text "*John Smith is accused of stealing $1,000,000. Jane Smith was John's accomplice*" becomes "*PERSON 1 is accused of stealing IDENTIFIER:MONEY. PERSON2 was PERSON1's accomplice*" when the script is run.

A 30 day free trial is offered as well as instructions on use of the tool on Github [51] could potentially be used in conjunction with the tool 'Baleen' (DSTL) [52] to extract text from unstructured or semi-structured data.

A similar product offered by competitor SciBite is TERMite which uses named entity recognition to find specific information. This can be tuned to find sensitive details which can then be redacted. The

tool also uses the context of the information to identify whether it should be removed and can output the new document such as HTML, Word, XML and JSON.

9.3.1.2 Bitcurator-Redact (October 2018) [53]

This is a Java-based PDF redaction tool (with a GUI) that employs statistical named entity recognition. It is a desktop application which can be designed to help effectively remove Personally Identifiable Information (PIIs) from PDF files and can work individually or on many at once. It highlights sensitive information on the page and asks if you want to remove it or not. The tool will then redact PDF text areas completely and replaces the text characters with empty space whilst putting a black border around the relevant page.

9.3.1.3 Presidio API (October 2018) [54]

Presidio is an open-source data protection and anonymization tool which has been pre-trained using machine learning. It can be used to remove sensitive text such as credit card numbers, names, locations, social security numbers, bitcoin wallets, US phone numbers and financial data. An example of the anonymization performed is shown below in Figure 52 and it can work with both structured and unstructured text, as well as images (though the latter is still experimental).

Presidio can also return the data field of the text as well as the confidence score of the corresponding text that was redacted. For images, it can use optical character recognition (OCR) to redact text from images. In addition, different types of PIIs can be programmed into the tool to look for different features so the tool is highly customizable. The code is available on Github [54]

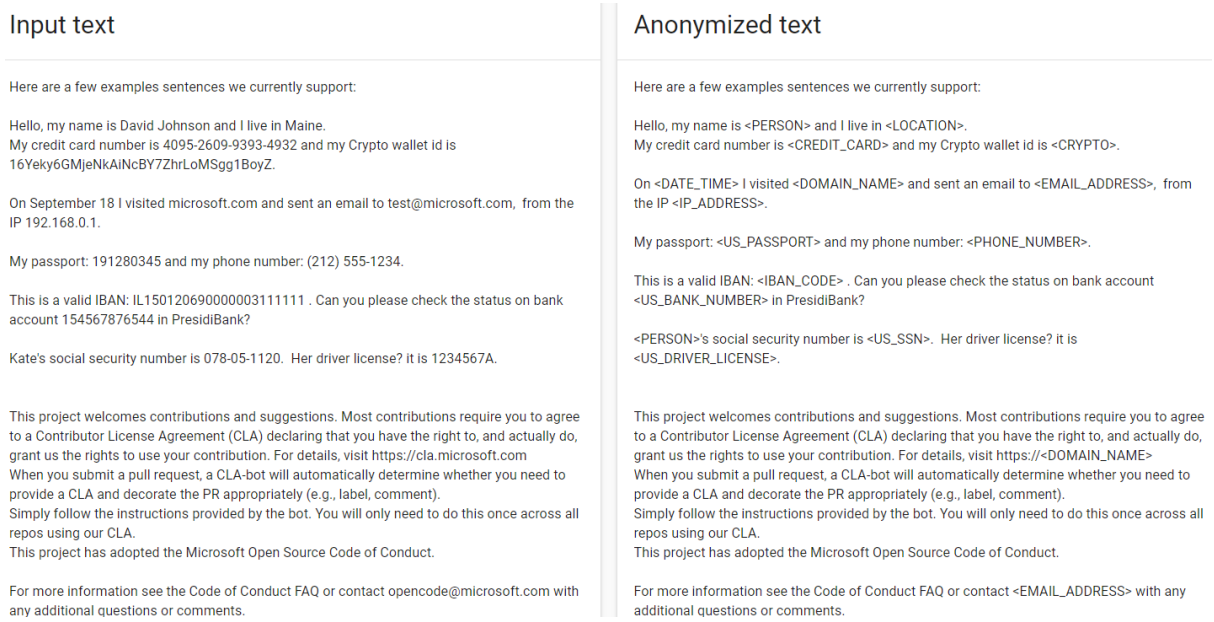


Figure 52 An example of the text redaction performed by Presidio

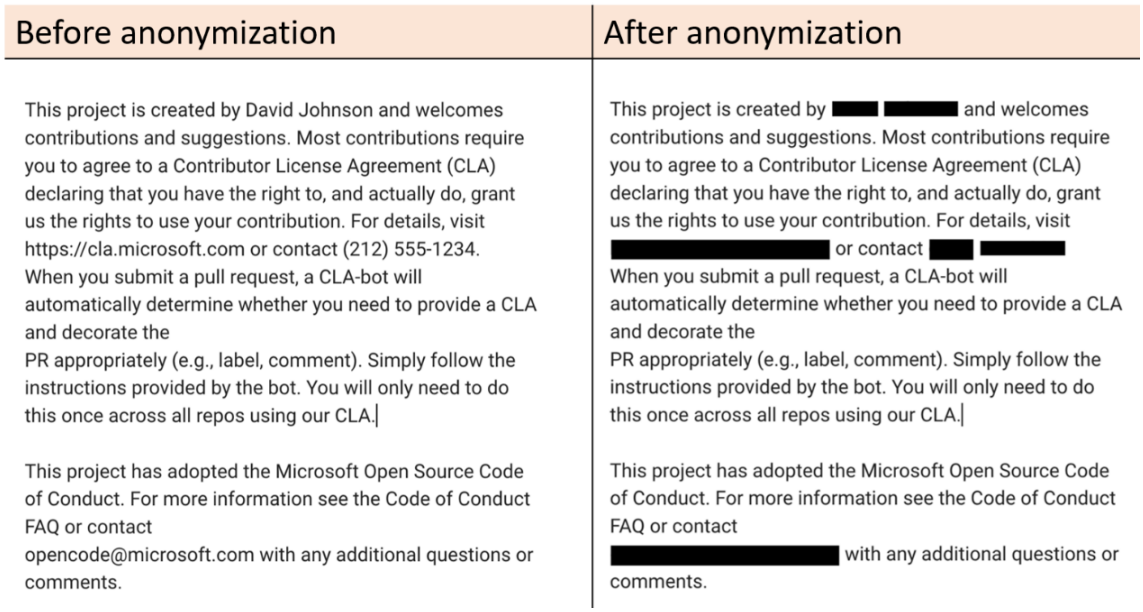


Figure 53 An example of Presidio using optical character recognition to mask out sensitive information in an image containing text.

Findings

Presidio Analysis

Field Type	Score	Text	Start:End
DATE_TIME	0.85	September 19th	39:53
PERSON	0.85	Aarav Navuluri	66:80
CREDIT_CARD	1	4095-2609-9393-4932	110:129
EMAIL_ADDRESS	1	aarav@presidio.site	143:162
DOMAIN_NAME	1	presidio.site	149:162
LOCATION	0.85	Amherst	177:184

Figure 54 Presidio will perform an analysis of the features it has redacted including the data type, the confidence score and the associated text.

9.3.1.4 OpenText Redact-It (December 2018) [55]

Redact-It is a regex-based commercial tool (as part of the OpenText Blazon software) which completely removes sensitive information from both structured and unstructured document/forms in a range of file formats including PDFs, Word documents and scanned images., but does not replace it with anything else (or produce synthetic data). The tool has built-in features to find sensitive information, such as phone numbers and credit card numbers, but you can add your own custom algorithms using regular expressions. There is a 15-day free trial available.

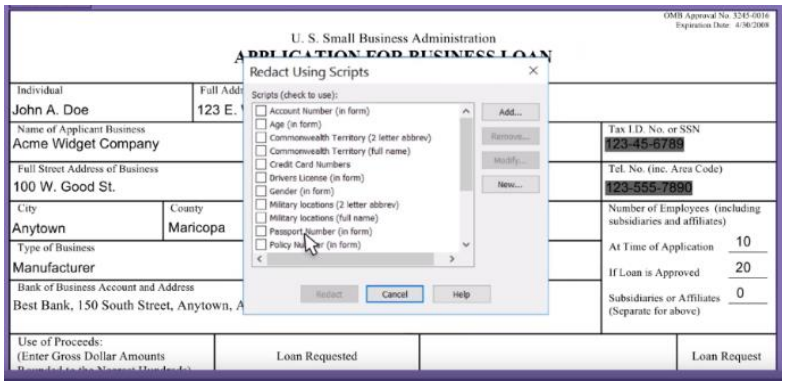


Figure 55 OpenText Redact-It software showing examples of features which can be redacted (top) and an example of a PDF upon which redaction scripts have been run (right).

9.3.2 Location-based Methods

9.3.2.1 A Survey of System Architectures, Privacy Preservation, and Main Research Challenges on Location-Based Services (June 2019) [56]

This paper provides a good overview of the problem area and challenges of protecting the privacy of users when using location (GPS) based services (LBS). The paper summarises three state of the art techniques to protect a user's privacy:

- **Anonymisation** – the process of anonymising the user from the LBS. This is done by grouping requests from multiple individuals and generalising the queries such that the user cannot be individually identified by the LBS.

  K-anonymity [57] [58] , as shown in Figure 56, is a common approach to ensure that data sent to the LBS cannot be used to identify less than k users. However this approach does require the users to completely trust the anonymiser as this has access to raw data from the user.

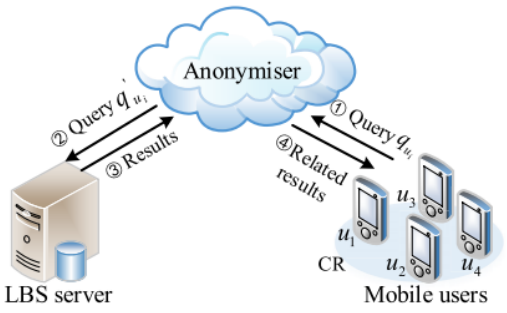


Figure 56 K-anonymity base privacy preserving system [58]

- **Obfuscation** - "the means of deliberately degrading the quality of information about an individual's location in order to protect that individual's location privacy." [60] . One such technique is to alter the location by selecting another location within a radius r of the real location. Examples of different techniques are shown in below in Figure 57.

  Encryption - make the user's LBS query invisible to the LBS server. Unlikely to be of interest for data obscuration.

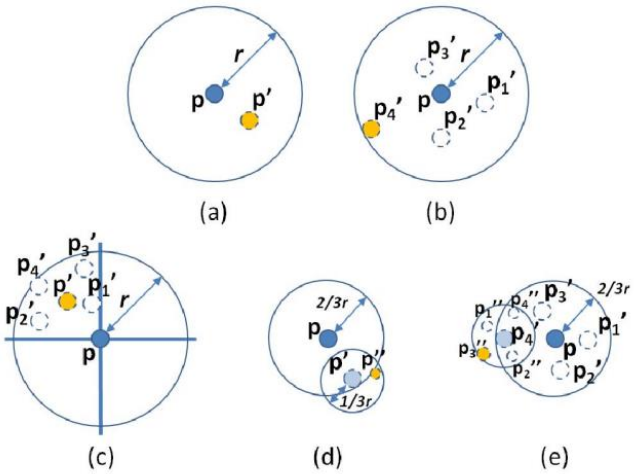


Figure 57 Example of the obfuscated point selection Randomize (a), N-Rand (b), N-Mix (c), Dispersion (d) and N-Dispersion (e) [59]

9.3.2.2 Synthesising Plausible Privacy-Preserving Location Traces [61]

This paper aims to create fake location traces (tracks) for individuals that represents their behaviour but cannot be used to identify the individual. A trace is an ordered sequence of locations that an individual has visited over time. A generator is trained using a corpus of real location traces and is used to synthesise fake traces that matches an individual's behaviour (i.e. visits similar places at similar times) and has realistic mobility profile for the new locations visited.

A flowchart which demonstrates the procedure is shown in Figure 58. For each trace in the corpus a mobility model is created that represents the visiting probability of each location (including time of visit and duration), and the probability of transitioning from one location to another. The locations are then clustered based on their semantic similarity, which aims to group together locations that are used in similar ways regardless of their geolocation. For example, if Alice and Bob spend all day at their respected work locations and all night at their home locations then these locations are likely to be clustered into two classes (home and work) even if though the entities in each class are different places (i.e. have different geolocations). A new synthetic trace is generated based on a real (seed) trace. The locations in the real trace are mapped to semantic classes (e.g. home and work) and then new locations are selected from each class taking into account the mobility of individuals in the area (e.g. speed of moving between locations, duration at location based on time of day) to

ensure the locations selected and movement patterns form a realistic trace. This information is used to create the synthesised trace.

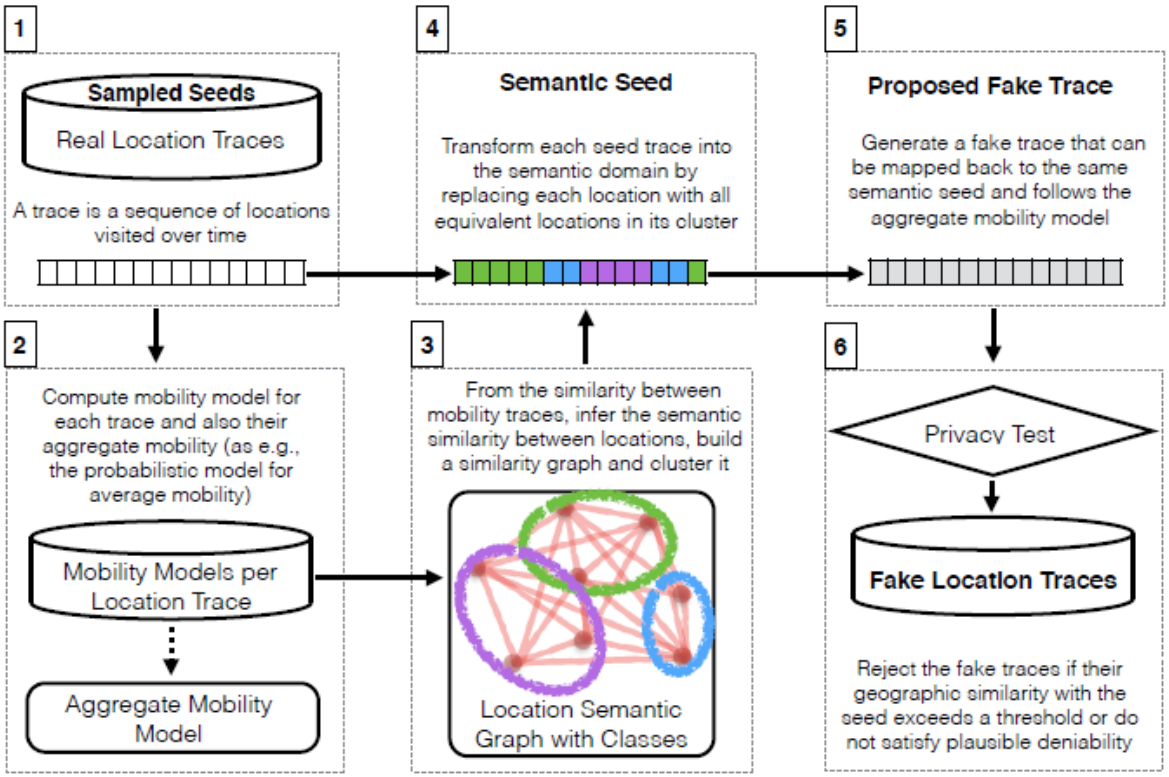
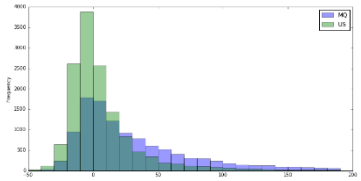
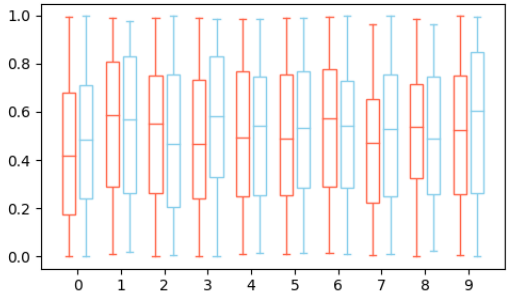
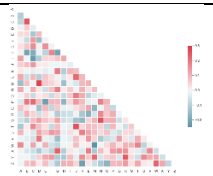


Figure 58 Summary of approach to generating privacy preserving traces [61]

C Appendix C Metric Tools

		Original Dataset		
Create a number of datasets with varying privacy budgets and evaluate each against the original dataset using the qualitative and quantitative methods identified below		DS1 (eg. ε = 200)	DS2 (eg. ε = 1)	DS3 (eg. ε =0.001)
Evaluating the utility of the dataset	**1. Qualitative evaluation of the statistical characteristics of each feature**	**1. Feature-wise histograms** Qualitative evaluation of the histogram of each feature in the original dataset against the same in each synthetic dataset Consider using seaborn (https://seaborn.pydata.org/tutorial/distributions.html) or matplotlib (https://matplotlib.org/3.1.3/api/_as_gen/matplotlib.pyplot.hist.html) package  **2. Box-plots**		

		 Code: https://stackoverflow.com/questions/43612687/python-matplotlib-box-plot-two-data-sets-side-by-side
	2. Quantify the distributional deviations	Use either Kullback-Leibler (KL) Divergence (https://machinelearningmastery.com/divergence-between-probability-distributions/) Or KS Distance or Wasserstein distance Explanation : https://www.datadoghq.com/blog/engineering/robust-statistical-distances-for-machine-learning/ Code: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wasserstein_distance.html Please note that you need to apply above functions on empirical cumulative distribution functions (CDF) and not on original data
	3. Evaluate Cross-correlation through heat maps (works only with numerical variables)	 Code: https://seaborn.pydata.org/examples/many_pairwise_correlations.html
	4. ML Based Techniques	e.g. compute feature-wise prediciton score using the random forest algorithm as discussed in DP Auto GAN paper and compre the feature-wise predition scores of each dataset – an indication of how data utility may suffer as the privacy budget is increases

Evaluating the disclosure Risk		https://hummedia.manchester.ac.uk/institutes/cmist/archive-publications/reports/2015-02%20-Report%20on%20disclosure%20risk%20analysis%20of%20synthpop%20synthetic%20versions%20of%20LCF_%20final.pdf Consider using the disclosure risk assessment methodology framework presented in page 3 of the document above. Consider implementing the KNN methods discusses below. https://ieeexplore.ieee.org/document/9005476

D Appendix D DPautoGAN python packages

List of packages and their versions used in ASC259.dpautogan conda environment.

Package name	Version
_libgcc_mutex	0.1
_pytorch_select	0.2
attrs	19.3.0
backcall	0.1.0
blas	1
bleach	3.1.0
ca-certificates	2019.11.27
certifi	2019.11.28
cffi	1.13.2
cudatoolkit	10.0.130
cudnn	7.6.4
cycler	0.10.0
dbus	1.13.12
decorator	4.4.1
defusedxml	0.6.0
entrypoints	0.3
expat	2.2.6
fontconfig	2.13.0
freetype	2.9.1
glib	2.63.1
gmp	6.1.2
gst-plugins-base	1.14.0
gstreamer	1.14.0
icu	58.2
importlib_metadata	1.3.0
intel-openmp	2019.4
ipykernel	5.1.3
ipython	7.10.2
ipython_genutils	0.2.0
ipywidgets	7.5.1
jedi	0.15.1
jinja2	2.10.3
joblib	0.14.1
jpeg	9b
jsonschema	3.2.0
jupyter	1.0.0
jupyter_client	5.3.4
jupyter_console	5.2.0
jupyter_core	4.6.1

Package name	Version
kiwisolver	1.1.0
libedit	3.1.20181209
libffi	3.2.1
libgcc-ng	9.1.0
libgfortran-ng	7.3.0
libpng	1.6.37
libsodium	1.0.16
libstdcxx-ng	9.1.0
libuuid	1.0.3
libxcb	1.13
libxml2	2.9.9
markupsafe	1.1.1
matplotlib	3.1.1
mistune	0.8.4
mkl	2019.4
mkl-service	2.3.0
mkl_fft	1.0.15
mkl_random	1.1.0
more-itertools	8.0.2
nbconvert	5.6.1
nbformat	4.4.0
ncurses	6.1
ninja	1.9.0
notebook	6.0.2
numpy	1.17.4
numpy-base	1.17.4
openssl	1.1.1d
pandas	0.25.3
pandoc	2.2.3.2
pandocfilters	1.4.2
parso	0.5.2
pcre	8.43
pexpect	4.7.0
pickleshare	0.7.5
pip	19.3.1
prometheus_client	0.7.1
prompt_toolkit	3.0.2
ptyprocess	0.6.0
pycparser	2.19
pygments	2.5.2
pyparsing	2.4.5
pyqt	5.9.2
pyrsistent	0.15.6

Package name	Version
python	3.6.9
python-dateutil	2.8.1
pytorch	1.3.1
pytz	2019.3
pyzmq	18.1.0
qt	5.9.7
qtconsole	4.6.0
readline	7
scikit-learn	0.22
scipy	1.3.2
send2trash	1.5.0
setuptools	42.0.2
sip	4.19.8
six	1.13.0
sqlite	3.30.1
terminado	0.8.3
testpath	0.4.4
tk	8.6.8
tornado	6.0.3
traitlets	4.3.3
wcwidth	0.1.7
webencodings	0.5.1
wheel	0.33.6
widgetsnbextension	3.5.1
xz	5.2.4
zeromq	4.3.1
zipp	0.6.0
zlib	1.2.11

E Appendix D SDV (Financial Dataset)

Permanent Order	Record Identifier	Account Identifier	Recipient Bank	Recipient Account	Order Amount	Type of Payment
	order_id	*account_id*	*bank_to*	*account_to*	*amount*	*k_symbol*
ORDER	29401	1	YZ	87144583	2452.00	SIPO
	29402	2	ST	89597016	3372.00	UVER
	29403	2	QR	13943797	7266.00	SIPO
	29406	3	AB	59972357	3539.00	POJISTNE
	29415	10	QR	93182509	1344.00	LEASING
	29433	25	AB	79838293	1110.00	

Transactions	Trans Identifier	Account Identifier	Trans Date	Trans Type	Mode of Trans	Amount	Balance After	Type of Trans	Partner bank	Partner account
	trans_id	*account_id*	*date*	*type*	*operation*	*amount*	*balance*	*k_symbol*	*bank*	*account*
TRANS	1117247	3818	930101	PRIJEM	VKLAD	600	600.00			
	637742	2177	93015	PRIJEM	PREVOD Z UCTA	5123	5923.00	DUCHOD	YZ	62457513
	695560	2378	930131	VYDAJ	VYBER	34700	59453.70			
	2349940	7753	940103	VYDAJ	VYBER KARTOU	3600	61068.10			
	3215628	10670	94013	PRIJEM	VKLAD	31607	101624.10			
	1042445	3566	940105	VYDAJ	PREVOD NA UCET	294	26767.50	SIPO	QR	38624727
	2699732	8934	940105	PRIJEM	PREVOD Z UCTA	14918	37466.70		MN	79855632

Disposition	Record Identifier	Client Identifier	Account Identifier	Type of Disposition
	disp_id	*client_id*	*account_id*	*type*
DISP	1	1	1	OWNER
	2	2	2	OWNER
	3	3	2	DISPONENT

Accounts	Account Identifier	Branch Identifier	Frequency of State Issuance	Date of Account Creation
	account_id	*district_id*	*frequency*	*date*
ACCOUNT	1972	77	POPLATEK MESICNE	930102
	1539	1	POPLATEK PO OBRATU	930103
	2087	7	POPLATEK TYDNE	930108

	Keyword	Translation
Order	SIPO	Household/Form of Direct Debit
	UVER	Loan Payment
	POJISTNE	Insurance Payment
Trans	PRIJEM	Credit
	VYDAJ	Withdrawal
	VYBER KARTOU	Credit Card Withdrawal
	VKLAD	Credit in Cash
	PREVOD Z UCTU	Collection from Another Bank
	VYBER	Withdrawal in Cash
	PREVOD NA UCET	Remittance to Another Bank
	POJISTNE	Insurance Payment
	SLUZBY	Payment on Statement
	UROK	Interest Credited
	SANKC. UROK	Sanction Interest
	SIPO	Household
	DUCHOD	Old-age Pension
	UVER	Loan Payment
Disp	DISPONENT	User
Account	POPLATEK MESICNE	Monthly Issuance
	POPLATEK PO OBRATU	Issuance after Transaction
	POPLATEK TYDNE	Weekly Issuance

Report Documentation Page v5.0

* Denotes a mandatory field

1a.	**Report number: *** ASC, 0259 D3 V1.1		**1b.**	**Version number:** v1.1
2	**Date of publication: *** 02/06/2020		**3.**	**Number of pages:**
4a.	**Report UK protective marking: ***	OFFICIAL		
4b.	**Report national caveats: ***	NONE		
4c.	**Report descriptor: ***	NONE		
5a.	**Title: *** LTI Synthetic Data			
5b.	**Title UK protective marking: ***	OFFICIAL		
5c.	**Title national caveats: ***	NONE		
5d.	**Title descriptor: ***	NONE		
6.	**Authors: *** Brijesh Patel, Gary Francis, Indika Wanninayake, Alan Pilgrim and Ben Upton (all BAE Systems Applied Intelligence Labs)			
7a.	**Abstract: *** This project has reviewed the state of the art techniques to create synthetic datasets that mimic the characteristics of the real dataset as closely as possible, but remove or obscure any private or sensitive information.			
7b.	**Abstract UK protective marking: ***	OFFICIAL		
7c.	**Abstract national caveats: ***	NONE		
7d.	**Abstract descriptor: ***	NONE		
8.	**Keywords:** LTI, Synthetic Data, Privacy, GANs			

Please note: Unclassified, Restricted and Confidential markings can only be used where the report is prepared on behalf of an international defence organisation and the appropriate prefix (e.g. NATO) included in the marking.

* Denotes a mandatory field

9.	**Name and address of publisher: *** BAE Systems Applied Intelligence Limited Chelmsford Office & Technology Park Great Baddow Chelmsford Essex CM2 8HN		**10.**	**Name and address of funding source:**
11.	**Funding source contract:**			
12.	**Dstl project number:**			
13.	**Programme:**			
14.	**Other report numbers:**			
15a.	**Contract start date:**	30/10/2019	**15b.**	**Contract end date:** 15/04/2020
16.	**IP conditions for report: ***	DEFCON 703 Third Party Rights		
17a.	**Patents:**	No		
17b.	**Application number:**			
18.	**Release authority role:**			

Guidance on completing the report documentation page can be found on the Gov.UK website.

20150701_Guidance
_Document_Vers_5_F

This page is intentionally blank