Høgskolen i Østfold

MASTER'S THESIS

Generating Synthetic Health Data Using Machine Learning GAN Methods

Elaheh Shahmir Shourmasti
October 15, 2022

Master in Applied Computer Science
Faculty of Computer Science

Master's Thesis

Generating Synthetic Health Data Using Machine Learning GAN Methods

Elaheh Shahmir Shourmasti

A thesis presented for the degree of
Master in Applied Computer Science

Faculty of Computer Sciences
Østfold University College
Halden
October 15, 2022

Acknowledgements

I am extremely grateful to my supervisors which supported me during this study. I would like to express my deepest appreciation to Michael Riegler, Celestino Creatore, and Vajira Thambawita for their worthwhile recommendations, much attentions, reviewing my work, and the inspiration for conducting this research. I am also grateful to Hanne Torill Mevik and Sukalpa Chanda for their support.

Abstract

Medical and healthcare researches have benefited greatly from machine learning (ML), but these effects have been definitely slower and more limited than in other application areas. Various concerns, particularly those relating to privacy, may restrict the access and use of electronic health record (EHR) data. This is mostly because patient privacy concerns have prevented data being widely available to the broader ML research community. Medical and healthcare data have been faced with security restrictions. Accessing medical data is hindered by privacy, security, and legal limitations. Organizations are incapable of sharing data due to the sensitivity of the data that involves information such as personally identified information or personally identifiable health information.

Generating synthetic data from real data is a possible approach that can tackle some of the challenges with medical data. A generative adversarial neural network (GAN) is a popular and powerful method to generate synthetic data from noise given a training dataset. Although promising it is not entirely clear how well privacy is preserved when GANs are used.

In this thesis various GAN methods for synthetic data generation have been investigated in terms of their generation and privacy preserving capabilities. Among all of the studied methods, Conditional Tabular GAN (CTGAN), Tabular GAN (TGAN), Wasserstein GAN (WGAN) and anonymization through data synthesis using GAN (ADS-GAN) were the methods that can generate tabular data, handle missing values and produce missing values in the generated sample on the same scale, handle an unbalanced dataset, and protect privacy so that there is minimal risk of data leakage. These methods were applied to the public adult census income dataset and different evaluation metrics were calculated to analyse the quality of the generated data for each method.

The findings show that none of the selected GAN methods is superior for all evaluation metrics. Based on the overall scores, the CTGAN and TGAN were selected to be applied on a privacy sensitive real world dataset provided by Fürst, which is a company that analyses blood samples in Norway. In depth evaluation is performed on the Fürst data and the results revealed that CTGAN can generate a high quality synthetic data that is similar to the real data but does not obfuscate the original data enough to protect privacy. Although in our experiments on the public data, it was concluded that TGAN can generate private synthetic data, it did not obtain a good score for the additional evaluation metrics and it performed the same as CTGAN. Based on our findings, it

is clear that there is a trade-off between the quality of the generated data and how much privacy is preserved. The main finding is that just by using GANs to create synthetic data from real data is not enough to preserve privacy.

Contents

1 Introduction **13**
- 1.1 Background and motivation . . . 13
- 1.2 Problem statement . . . 14
- 1.3 Limitations . . . 14
- 1.4 Contributions . . . 15
- 1.5 Ethical Considerations . . . 15

2 Preliminaries **17**
- 2.1 Synthetic Data . . . 17
- 2.2 Machine Learning . . . 17
- 2.3 Deep learning . . . 19
  - 2.3.1 Adversarial neural networks . . . 19
  - 2.3.2 Probability distribution . . . 20
- 2.4 Generative adversarial network . . . 20
  - 2.4.1 Generator: . . . 21
  - 2.4.2 Discriminator: . . . 22
  - 2.4.3 Challenges with GANs in Tabular Data . . . 22

3 Related works **23**
- 3.1 DataSynthesizer . . . 23
  - 3.1.1 Table-GAN . . . 24
  - 3.1.2 TGAN . . . 24
  - 3.1.3 CTGAN . . . 25
  - 3.1.4 Bayesian Network . . . 25
  - 3.1.5 SynSys . . . 25
  - 3.1.6 HealthGAN . . . 26
  - 3.1.7 medGAN . . . 26
  - 3.1.8 CoreGAN . . . 27
  - 3.1.9 PATE-GAN . . . 27
  - 3.1.10 G-PATE . . . 27
  - 3.1.11 WGAN . . . 28
  - 3.1.12 ADS-GAN . . . 28
- 3.2 Privacy Preserving Methods . . . 29
- 3.3 Data synthesizing Methods Summary . . . 29

4 Proposed methods **32**
4.1 Selected GAN methods for tabular data 32
4.1.1 Dataset: 32
4.2 Generated data quality check 32
4.2.1 Similarity 33
4.2.2 Discriminator and generator losses 40
4.2.3 PRDC evaluation 43
4.2.4 Privacy evaluation 49
4.2.5 Summary 52

5 Experiments on the real data **55**
5.1 Data Description 55
5.2 Exploratory Data Analysis (EDA) 56
5.2.1 Pandas Profiling 56
5.2.2 SweetVIZ 56
5.2.3 AutoViz 57
5.3 Generated data quality check 58
5.3.1 Similarity 58
5.3.2 Synthetic Data Vault (SDV) 66

6 Conclusions **72**
6.1 Summary 72
6.2 Contributions 73

7 Future work: **75**

Appendices **81**

A Additional graphs **82**

List of Tables

4.2 Table evaluator statistical results on CTGAN 37
4.4 Identifiability measure for GAN methods 51
4.1 Adult data set description 53
4.3 Table evaluator Classifier F1scores on CTGAN 54

5.1 Fürst dataset description 56
5.2 Table evaluator Classifier F1scores on CTGAN 67
5.3 Table evaluator Classifier F1scores on TGAN 67
5.4 Table evaluator statistical results on CTGAN and TGAN 69
5.5 SDV statistical metric values on CTGAN and TGAN 69
5.6 SDV Detection metric value on CTGAN and TGAN 71
5.7 Machine Learning Efficacy Metrics value on CTGAN and TGAN 71
5.8 Machine Learning Efficacy Metrics value on CTGAN and TGAN 71

List of Figures

2.1 comparison of AI, machine learning and deep learning 18
2.2 Overview of machine learning algorithms 19
2.3 Overview of the generative adversarial network workflow [1] . . . 21

4.1 Absolute Log mean and STDs of numeric data for CTGAN . . . 37
4.2 cumulative sums for age and work-class columns in the real and fake dataset for CTGAN 37
4.3 cumulative sums for final weight and education columns in the real and fake dataset for CTGAN 38
4.4 cumulative sums for edcation_num and marital_status columns in the real and fake dataset for CTGAN 38
4.5 cumulative sums for occupation and relationship columns in the real and fake dataset for CTGAN 39
4.6 cumulative sums for race and sex columns in the real and fake dataset for CTGAN 39
4.7 cumulative sums for capital_gain and capital_loss columns in the real and fake dataset for CTGAN 40
4.8 cumulative sums for hours_per_week, native_country and income columns in the real and fake dataset for CTGAN 40
4.9 Generator loss and Discriminator loss on real and fake data on VanillaGAN 41
4.10 Generator loss and Discriminator loss on real and fake data on CTGAN 42
4.11 Generator loss and Discriminator loss on real and fake data on WGAN 42
4.12 Generator loss and Discriminator loss on real and fake data on ADS-GAN 42
4.13 Precision versus density [31] 44
4.14 Recall versus coverage [31] 45
4.15 PRDC evaluation on whole adult dataset with K value equal to 5 46
4.16 PRDC evaluation on whole adult dataset with K value equal to 10 46
4.17 PRDC evaluation on whole adult dataset with K value equal to 20 47
4.18 PRDC evaluation on 5000 sub-samples of the adult dataset with K value equal to 5 47

4.19 PRDC evaluation on 5000 sub-samples of the adult dataset with K value equal to 10 48
4.20 PRDC evaluation on 5000 sub-samples of the adult dataset with K value equal to 20 48

5.1 Distribution plot, boxplot and probability plot-skew of T3 . . . 57
5.2 Distribution plot, boxplot and probability plot-skew of T4 . . . 57
5.3 Distribution plot, boxplot and probability plot-skew of VB12 . . 58
5.4 Distribution plot, boxplot and probability plot-skew of Urinstoff 58
5.5 Distribution plot, boxplot and probability plot-skew of TSH . . 58
5.6 Distribution plot, boxplot and probability plot-skew of LDL . . 59
5.7 Distribution plot, boxplot and probability plot-skew of HDL . . 59
5.8 Distribution plot, boxplot and probability plot-skew of Kolesterol 59
5.9 Distribution plot, boxplot and probability plot-skew of Kreatinin 60
5.10 Distribution of the label feature 60
5.11 Bar plot for Kreatinin by label 61
5.12 Bar plot for TSH by label 61
5.13 Bar plot for T4 by label 62
5.14 Bar plot for T3 by label 62
5.15 Bar plot for Urinstoff by label 63
5.16 Bar plot for HDL by label 63
5.17 Bar plot for Kolesterol by label 64
5.18 Bar plot for LDL by label 64
5.19 Bar plot for VB12 by label 65

A.1 Distribution of the label column in Hypothyroidism subset . . . 82
A.2 Distribution of the label column in Hyperthyroidism subset . . . 82

Chapter 1

Introduction

1.1 Background and motivation

There is an increasing demand for the utilization of Machine Learning (ML) technologies for medical applications. The implementation of machine learning on medical records can pave the way for medical trajectories and offer many advantages in terms of accurate and time-saving diagnoses, analyzing medical tests, new insight and knowledge about different diseases etc. Machine-learning techniques can help to predict progressions and treatments for different diseases, the consequences of the medicine tests by analyzing and monitoring patterns among existing data samples.

However, strategies for the analysis of medical data are often faced with major problems of having small and/or flawed datasets, In such cases, synthetic data can support developing and testing machine learning models. Moreover, data privacy is becoming increasingly important for the healthcare domain. Medical data includes personal and health information which is highly sensitive. Therefore, specific machine learning methods would be needed to generate anonymous data. These artificially generated data would help healthcare organizations to share knowledge while preserving the information. Synthetic data generation can maintain the significant characteristics of the main sample which has the most impact for specifying the pattern accurately and it is the modification of the existing data. Accordingly, using synthetic data would benefit healthcare systems to test and analyse medical records without exposing user's data.

In this project, the goal is to implement machine learning algorithms and explore frameworks for generating high-quality synthetic samples based on real patient records collected and stored by Fürst Medisinsk Laboratorium. The data is records of blood tests of patients and the goal is to generate blood test analysis results. In the project we will specifically focus on GAN. GAN is an unsupervised learning method consisting of two neural networks that are competing with each other. It tries to learn the pattern of its input by itself

and generate new samples with similar characteristics to the real dataset. VAE which are another approach for generating synthetic data, describe observations in latent space based on their probability. An encoder would be formulated to report the probability distribution of each latent attribute. The auto-encoder takes the high dimensional input data, runs it through the neural network and tries to compress the data into a smaller representation that has less dimension. Afterwards, it will reconstruct the input and the loss function of the model will compute the reconstruction loss by comparing the input and output.

1.2 Problem statement

Generative adversarial network methods are widely used in generating samples such as images and tabular data. The main objective for this thesis is to study different GAN methods and find the most suitable method(s) that can perfectly generate samples that meet the demanding criteria, both in quality and privacy aspects. Then we use this method(s) to generate the most ideal samples based on the real patient's blood sample. The following research questions (RQs) should be raised:

- ***RQ 1*** *What are the best GAN methods for generating synthetic tabular data?*
- ***RQ 2*** *How well do the proposed GAN methods perform in terms of data privacy and similarity?*
- ***RQ 3*** *How effective are the chosen GAN methods for generating synthetic data from a real world dataset, namely the Fürst dataset?*

This work is divided into three stages to address the aforementioned objectives. The first stage involve studying different GAN methods, their structure and their advantages and disadvantages. The second stage is selecting the methods that are capable of generating high quality and private tabular datasets. The selected benchmarks are analysed to see how good the generated samples are in terms of quality and confidentiality. Various evaluation metrics are used to inspect the results and best model(s) are selected. In the last stage, these outperformed metrics are performed on the real dataset, synthetic data are generated and the quality evaluations are conducted.

1.3 Limitations

In order to make reliable conclusions from a study, the sample needs to be valid and large enough. It will be difficult to find meaningful links in the data if your sample size is too small or includes a lot of missing values. Hence, the selected method should be capable of handling missing values and generating data that compensates for the lack of inadequate data.

Moreover, working with sensitive data faces the problem of having limited access. This research involves medical data of people, which includes sensitive information that needs to be kept private. Being incapable of sharing the dataset may cause bottleneck. The in-depth data exploration is not feasible. However, as long as no individual information involves, the results of the study can be shared to do comparisons and further studies.

1.4 Contributions

This thesis is focused on study of performance of generative adversarial network (GAN) methods on generating synthetic health data. Throughout this thesis, we learned that some GAN models can generate high quality synthetic tabular data. However, in healthcare concepts, the importance is not only about producing data with equivalent statistical properties to the real data, but also generating synthetic data that can preserve privacy. Moreover, a paper for this thesis will also be published.

The main contributions of this thesis are the following:

1. We research different GAN methods and study their layout and use cases to find out what are the most relevant methods for generating synthetic tabular data.

2. We compare the performance of the selected GAN methods by conducting experiments on a public tabular dataset and evaluate the performance of each GAN method in terms of similarity and privacy. The results show none of the GAN methods can outperform the other in all the evaluation metrics. Based on the overall performance score, CTGAN and TGAN are the two final selected methods.

3. The final selected methods are performed on the private Fürst Medisinsk Laboratorium dataset to evaluate the performance of the selected methods. More in depth evaluations are conducted to compare the results of the two methods.

4. Based on our findings, CTGAN can produce very similar and high quality data. However, it can not gain a high margin in terms of privacy. Even though the TGAN performs the best among the other selected methods in the initial examination in terms of privacy, it does not outperform the CTGAN in the additional evaluations. Therefore, GANs should be used with caution when dealing with highly sensitive data because they do not protect privacy by nature.

1.5 Ethical Considerations

Since people make unconscious decisions while choosing the data that seems most appropriate for a dataset, biases exist in all datasets. This issue might be

made worse by synthetic data. Synthetic data will not produce a "fair sample" of the real data it represents; instead, it is more likely to concentrate on and accentuate certain biases and patterns that exist in the real world. A synthetic dataset will only ever be a "snapshot in time", whereas, real data will constantly change and develop. The AI models generate a limited sets of predictions since they were trained on the biased and repetitive datasets. There are many cases that the synthetic data is not able to accurately model reality and these are the challenges that people are needed to be aware of to choose whether they need to face the complexity of collecting real data or they can settle for synthetic data for that problem.

The dataset has been kept confidential and anonymous to avoid any re-identification or disclosures of the data. All the experiment have been performed on a specific machine provided by Simula research laboratory in Fürst Medisinsk Laboratorium to preserve data and observe confidentiality and the research has benefited from the Experimental Infrastructure for Exploration of Exascale Computing (eX3), which is financially supported by the Research Council of Norway.

Chapter 2

Preliminaries

Before going deep into generating synthetic data, it is needed to define some basic concepts for a better understanding. This chapter tries to clarify some related concepts about machine learning, specifically Generative Adversarial Networks(GANs) methods.

2.1 Synthetic Data

Inadequate data is causing many problems for AI industries. Synthetic data is an approach to overcome a series of pitfalls with these problems which could be generated either by data manipulation or producing artificial data. Artificial data does not exactly look like the real data, but it tries to mimic the characteristics of it. Many applications of synthetic data are being used in the fields of social sciences, healthcare, and economics. mostly, the main concern of researchers in healthcare area that leads them to the synthetic data is privacy issues rather than lack of data [32, 4].

2.2 Machine Learning

A branch of computing algorithms called machine learning, a subset of artificial intelligence, is constantly developing and aims to mimic human intelligence by learning from the environment (Figure 2.1). The typical stages of the learning process starts with a given collection of labeled examples. Labeled data means that values or categories are assigned to the examples. The data will be separated into a training sample, a validation sample, and a test sample randomly. Afterward, practical relevant features will be associated with the example so that the learning process can be trained. At this step values of the hyperparameter (the values that are not determined by the learning algorithm) will be tuned by selecting different hypotheses for each value. At last, with the best-performed hypothesis, the labels of the example in the test sample will be predicted [30, 16].

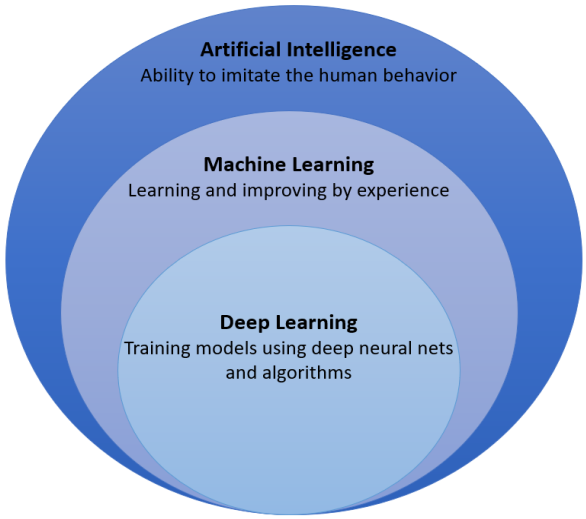


Figure 2.1: comparison of AI, machine learning and deep learning

Training data, the order and method of capturing data procedure, and the test data which is used for assessing determine the types of learning algorithms (Figure 2.2). The following learning scenarios are the most common ones:

Supervised learning: In this scenario, a function will be developed by the algorithm which gets a set of labeled data as training input and maps it to the output which is the prediction or classification for the unseen data. In supervised learning, the agent (software entity that obtains information from the environment) tries to learn from the training set and minimize the difference between the predicted and the expected values with the received information. Meanwhile, it also tries to avoid over-fitting which is simply memorizing the training set instead of learning the classification technique. Supervised machine learning problems is been distinguished based on the type of the output.This could be regression problem with numerical output or classification with categorical ones [3, 6].

Unsupervised learning: In unsupervised learning there is no available labeled example for the learner. This method is basically about modeling the probability density of the input. The algorithm learns to extract information from data distribution, finds patterns and best presentation of the data. Clustering similar groups and dimensionality reduction are two simple classic examples of unsupervised learning. Moreover, Object segmentation, similarity detection and automatic labeling are some of the common unsupervised application areas [19, 23].

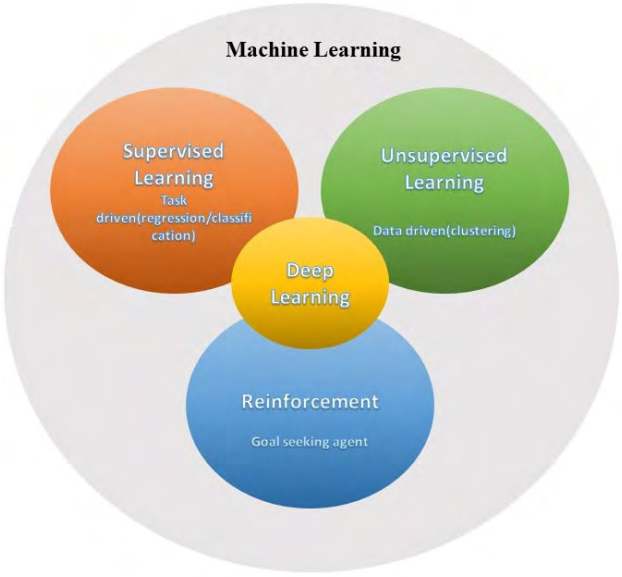


Figure 2.2: Overview of machine learning algorithms

Reinforcement learning: Learning process in this method is based on the information that the environment gives to the learner as a feedback which also called reward. The agent will learn from direct interaction with the environment, and that is what differentiates reinforcement learning from other computational approaches. The learner tries the actions to meet the ones that results in the highest reward. Each action has impact on each current state's reward and subsequently the following rewards. In other words, the agent can trade off short-term rewards for long-term rewards. Therefore, trial and error, and delayed reward are the most important features in this method of learning [6, 43].

2.3 Deep learning

2.3.1 Adversarial neural networks

In a generative network, the generative model is set against a rival. The generator observes the real sample distribution, attempts to generate synthetic data that looks similar to the real data, and fools the discriminator so that it can not differentiate between real and fake data. In contrast, the discriminator model learns to detect the real data from the fake ones [21].

2.3.2 Probability distribution

The probability distribution is the likelihood of each possible state that is being taken on by a random variable or sets of random variables and the definition of the probability is based on the variables being discrete or continuous.

Probability mass function (PMF) is used over discrete variables. The state of a random variable will be mapped to the probability of that random variable acquire that state. Probability density function (PDF) is used to describe probability distribution for continuous variables. PDF conveys how likely the variable lies in a very small zone with volume δx. [19].

Joint probability distribution: A probability distribution that can serve many variables simultaneously, is referred to as a joint probability distribution.

2.4 Generative adversarial network

Generative Adversarial Network was first introduced by Ian Goodfellow (2014) [21]. This framework is simulated as a two player game in which two models contend with each other. The generative indicates creating new data and adversarial refers to the competitive dynamic between the two models. A generative model's objective is to analyze a set of training examples and discover the probability distribution that produced them. The estimated probability distribution is then used by Generative Adversarial Networks (GANs) to produce more instances.

The generative model (G) attempts to capture the distribution of the real sample of the training and create non-distinguishable fake data from real data. Meanwhile, the discriminator's (D) objective is to distinguish whether the data comes from the real samples or the fake examples that the generative produced (Figure 2.3). By way of explanation, D is getting trained to maximize the likelihood that training examples and samples from G will receive the right label. In parallel, G is being trained to reduce the value $[log(1 - D(G(z)))]$ which means generating realistic data so it can fool the discriminator. Accordingly, the objective of GAN can be described as follows [20]:

$$\underset{G}{min}\underset{D}{max}V\left(G,D\right)=\mathbb{E}_{x\sim p_{data}(x)}\left[logD(x)\right)]+\mathbb{E}_{z\sim p_{z}(z)}\left[log(1-D(G(z)))\right] \quad (2.1)$$

Although GANs are widely used for image generation, but they can also generate tabular data. Simply put, tabular GANs are GANs that produce datasets with a tabular format. The generator and discriminator in the initial GAN architectures were both fully connected neural networks [19]. In tabular data generation , each row in table (T) is a vector (C), and the table itself includes

n-c continuous variables and n-d discrete (categorical) variables with unknown joint distribution (P). A generative model (M) is being trained. M must create a new T-synth synthetic table with P-like distribution [2].

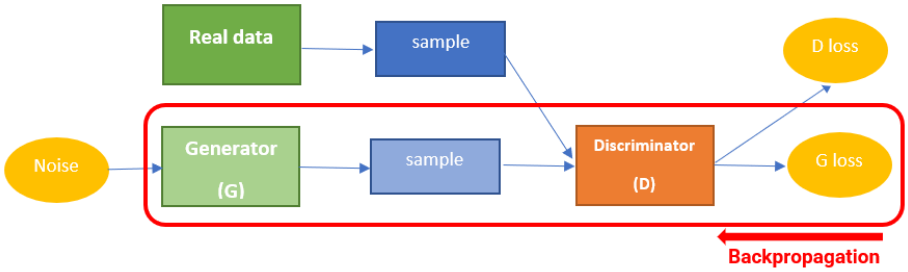


Figure 2.3: Overview of the generative adversarial network workflow [1]

Numerical variables Tanh activation function enable neural networks to efficiently produce values with a distribution centered around (-1, 1). However, in multi modal data, the networks can not make proper data. Therefore, they train a Gaussian Mixture Model (GMM) with m (m=5) components for each of C to cluster a numerical variable. At the end, Finally, C is normalized to produce V using GMM. In addition, they generate a vector U representing the likelihood that C will come from each of the m Gaussian distributions.[46, 2]

categorical variables Using softmax activation function, the probability distribution can be directly produced. However, category values must be transformed into binary variables using a one-hot encoding representation with noise. It will be prepossessed with n_c + n_d columns and then transform them into V, U, and D vectors. This vector serves as both the discriminator's input and the generator's output in a GAN. GMM parameters are inaccessible to GAN [2].

2.4.1 Generator:

In the beginning of the training, the generator produces samples that are easily distinguishable by the discriminator. The generator takes the noise vector as input. The benefit of using noise vector is to make sure it does not generate the same data. The noise will be generated by random numbers from a normal distribution.

Training examples (x) in generative modeling are taken from an unknown distribution $p_{data}(x)$. Writing a function called $p_{model}(x;\theta)$ that is explicitly controlled by parameters θ and then looking for the value of the parameters

[1] https://developers.google.com/machine-learning/gan/gan_structure

that makes p_{data} and p_{model} as close as possible are two simple ways to learn an approximation of p_{data}. The generator G which is a differentiable function represented by a multilayer perceptron with parameters θ_g, is defined by a prior distribution $p(z)$ over a vector z that is used as input to the generator function $G(z; \theta^G)$ [20].

2.4.2 Discriminator:

The discriminator model predicts a binary class label of real or fake based on an input example from the domain (either real or generated data). The training dataset contains the actual example and the generator model outputs the created examples.

2.4.3 Challenges with GANs in Tabular Data

A GAN model's design is challenging because of some unique properties of tabular data [48].

imbalanced categorical columns Many of the tabular datasets are imbalance. Therefore, training opportunities for minor classes would be limited due to having imbalanced data. This can cause intense mode collapse.

Mixed data types Tabular data in the real world contains a variety of types. GANs must use both softmax and tanh on the output to produce a mixture of discrete and continuous columns.

Multimodal distributions If a tabular dataset has multiple modes, it may be challenging to model the multimodal distribution of continuous columns.

Non-Gaussian distributions Unlike the pixel values in image which follow a gaussian-like distribution and can be normalized to [-1,1], tabular data usually continuous non-Gaussian values which can lead to vanishing gradient problem [48].

Chapter 3

Related works

In This section, we discuss the methods of generating synthetic data and their advantages and disadvantages, and compare them with each other. We studied these methods in detail.

3.1 DataSynthesizer

Dankar et al. [11] investigated the impact of different synthetic data generations on the utility of the generated synthetic data. They centered the investigation on four domains.

- How data pre-processing affect the utility of the synthetic data generation
- Applying tuning on the synthetic dataset during the generation supervised machine learning models
- The importance of sharing the primary machine learning results for data generation model's improvement
- To what extend propensity score can predict the accuracy of the model in real-life usage.

They used propensity score and 4 classification algorithms to assess synthetic data utility: Logistic regression (LR), support vector machines (SVM), Random forest (RF) and decision trees (DT).

The authors used four data synthesizer for this evaluation. DataSynthesizer or DS builds a Bayesian network to comprehend the hidden relationship structure of the different attributes. DS supports various types of data such as numeric, categorical and non-categorical, date, string, and missing values [38]. The Synthetic Data Vault (SDV) uses latent Gaussian copula to predict the common distribution of the population by considering all the attributes being numeric. In case that this assumption did not run, the pre-processing is needed to be applied to the dataset to reconstruct the attributes into a 0 to 1 range

of numerical values which indicates the repetition of the values in the dataset. Similarly, the DateTime can be converted to numerical values by counting the number of seconds. Moreover, in SDV missing values are considered null values and they count as important information for the model [37]. Synthpop or SP uses conditional distribution estimation to generate a synthetic dataset and it uses two methods for that. SP-np method works with nonparametric CART algorithm (Classification and Regression Trees) and SP-p uses logistic regression and linear regression [33].

The results from Dankar et al. [11] experiments on 15 different tabular public datasets showed that according to propensity score, generating synthetic data from raw real dataset tended to have better performance compared to pre-processed real data. Furthermore, importing the real data tuning setting granted better accuracy across all ML algorithms and for all synthesizers, rather than tuning the synthetic data independently. Besides, there was insignificant difference between tuned and non-tuned synthetic dataset which was generated from raw unprocessed data. Therefore, there is no confirmation to tune the synthetic dataset. Ultimately, employing propensity scores raise the prediction accuracy when synthetic data is generated using SP-p and DS synthesizers.

3.1.1 Table-GAN

A study has been conducted in order to compare the techniques of preserving privacy while sharing or releasing data in the public [34]. Four real-world datasets from four different domains were employed to compare anonymization, perturbation, and generation techniques.

Anonymization techniques remove sensitive attributes. However, the identification of records can be retrieved if adversaries have that knowledge. Data perturbation is about adding noise or altering the values. Still, these modifications can have an adverse influence on data usability. Park et al. [34] proposed table-GAN method to generate synthetic tables. Their model adopts the deep convolutional GAN (DCGAN) [40] architecture with an additional neural network called classifier. Accordingly, the model has a generator neural network G to produce synthetic records, a real and fake records identifier called discriminator (D) and, a classifier neural network (C) to predict synthetic records' labels. Having a classifier will help to keep the consistency of values in the generated records. According to the results, table-GAN presents the best balance between privacy level and model compatibility.

3.1.2 TGAN

A GAN-based synthetic data generator called TGAN for tabular data is developed by Xu et al. [49]. They selected three tabular datasets from the UCI Machine Learning Repository and used LSTM (Long short-term memory) in their model and they intended to produce mixed variable types like multinomial, discrete, and continuous. The generated synthetic data by TGAN was statistically evaluated. The authors claimed that training the machine learning

model with data generated from TGAN presented a better performance and generates high-quality synthetic data compared to three other data synthesizers (GC, BN-Id, and BN-Co) that rely on multivariate probabilistic graphical models. However, the presented model only supports a single table with numerical and categorical features.

3.1.3 CTGAN

Conditional Tabular GAN (CTGAN) [48] is one of the numerous extensions of GAN that models tabular data distribution. Generating tabular data with GAN has been faced with some challenges. Continuous values in tabular data are usually non-Gaussian. Through normalizing, these values face vanishing gradient problem. Additionally, modeling the multimodal distribution of continuous columns could be challenging. To tackle these dilemmas, Xu et al. [48] design the mode-specific normalization in which each column is processed independently, each value is represented as a one-hot vector indicating the mode and a scalar indicating the value within the mode. Therefore, it turns continuous values into a limited vector which is proper for neural networks. Moreover, they design conditional generators and training-by-sampling to deal with the imbalanced data issue.

3.1.4 Bayesian Network

Benedetti et al. [12] investigate some facets of issues that can be faced while using synthetic data as a substitute for real care data. "Handling the complexities of real-world data to transparently capture realistic distributions and relationships, modeling time, and minimizing the matching of real patients to synthetic data points" were issues that the authors mentioned and they believed using the suitable modeling approach would lead to a secure and transparent synthetic dataset. They employed two real datasets, MIMIC III dataset[1] which is the records of the stay of a patients and Clinical Practice Research Datalink (CPRD Aurum Database) which contains fully-coded patient electronic health records[2], and used Bayesian network to obtain the main characteristics of data and generated samples imitating those features. Moreover, the extended approach has successfully captured the key characteristics of blood pressure data by using dynamic Bayesian networks. Lastly, the nearest neighbor analysis was conducted to check the risk of the real and fake data equivalence. It concluded that the risk is unlikely to occur, and it is remarkably difficult to reach sensitive information.

3.1.5 SynSys

Machine learning-based synthetic data generation method, called SynSys, was introduced by Dahmen et al [10] to deal with complexity and realism limitations

[1] https://physionet.org/content/mimiciii/1.4/
[2] https://cprd.com/primary-care-data-public-health-research

that happen in the existing synthetic data generating methods. The generated synthetic time-series data was made of nested sequences using hidden Markov models and regression models. This model was tested on a real annotated smart home dataset which can be considered as a part of healthcare application. Smart home data represent the activities occurring in a smart home with a time series structure containing sensor events ordered by time. The first step in SynSys is to train the HMM (Hidden Markov Models library)[3] to generate a realistic sequence of activities. The second HMM is trained to generate the sequence of sensor events. Each activity of the sequence of activities that are generated by the first-level HMM is expanded into the corresponding sequence of sensor events generated by the second-level HMM. The last step is training the regression learners for creating the timestamps that calculate the time gap within sensor events and the duration of each activity. They compared SynSys and an alternative synthetic data generation method that does not use combinations of HMM's and regression algorithms. According to the results, SynSys algorithm generates more realistic synthetic data.

3.1.6 HealthGAN

Yale et al. [50] proposed HealthGAN that generates privacy-preserving synthetic health data which is a GAN-based method for generating mixed continuous and categorical data. The workflow of HealthGAN is to train the model inside a secure sandboxed environment using real data and export the model outside of the data-secure environment to generate the synthetic data. The authors implement the HealthGAN on MIMIC data (Medical Information Mart for Intensive Care) which consists of de-identified ICU data from 2001 to 2012. To assess the resemblance and privacy of synthetic data, nearest neighbor adversarial accuracy was used. They compared HealthGAN to other data generative methods including Gaussian Multivariate [13], Parzen Windows [35], Additive Noise Model (ANM) [24], Differential Privacy-preserving data obfuscation (DP) [14], and Copy the real data (CP). According to the results, HealthGAN was the only effective method in privacy maintenance, and that allowed model export.

3.1.7 medGAN

Choi et al. [8] focused on generating high-dimensional discrete variables to undertake the problem of aggregating discrete features derived from longitudinal EHRs (electronic health records) which is time-consuming. The authors proposed a neural network model that generates high dimensional multi-label discrete variables named medGAN. The design of the medGAN mode is a combination of an autoencoder and the adversarial framework. The dataset was derived from the multi-label discrete electronic health records (EHR). The authors declared that medGAN had an impressive results for both binary variables and count variables and the attribute disclosure had limited risk in this presented model. Yet, it is only capable of generating discrete data.

[3] https://ghmm.sourceforge.net/

3.1.8 CoreGAN

Correlation capturing Generative Adversarial Network (CorGAN) is proposed by Torfi and Fox [45] to generate synthetic healthcare records. This framework is composed of Convolutional GANs and Convolutional Autoencoders (CAs) and it is able to generate both discrete and continuous synthetic records. Due to the privacy assessment, this method provides an acceptable level of privacy. The selected datasets were MIMIC-III for binary discrete variables experiment and UCI Epileptic Seizure Recognition dataset (an unbalanced dataset) which contains brain activities characterizes for continuous variables experiment. The effectiveness of this method was measured by comparing it to Stacked Deep Boltzmann Machines (DBMs), Variational Autoencoder (VAE) and medGAN. According to their analysis, in CorGAN, the generated synthetic data represents a similar performance to the real data in different Machine Learning settings such as classification and prediction.

3.1.9 PATE-GAN

PATE-GAN [53] methodology is for generating differentially private synthetic data. Besides producing high-quality synthetic data, it provides differential privacy guarantees. Thi method is structured by modifying the version of the Private Aggregation of Teacher Ensembles (PATE) framework for the training procedure of the discriminator so it could be differentially private. The Credit card fraud detection real-world dataset from Kaggle, The Meta-analysis Global Group in Chronic Heart Failure (MAGGIC) dataset, The United Network for Organ Transplantation (UNOS) dataset , Kaggle cervical cancer dataset , UCI ISOLET dataset and UCI Epileptic Seizure Recognition dataset were the author's selected datasets. Two methods were used for evaluating the similarity of the synthetic datasets with a real dataset: "comparing the predictive performance of models trained on the synthetic datasets and tested on the real dataset", and "comparing the performance rankings of predictive models on the synthetic datasets with their performance rankings on the real dataset". The authors stated that "using PATE to enforce differential privacy results in higher quality synthetic data than DPGAN" and this method operated better compare to the state-of-the-art method [53].

3.1.10 G-PATE

Long et al. [28] believed that "It is not necessary to ensure differential privacy for the discriminator in order to train a differentially private generator". Therefore, they represented a new approach for training differentially private data generator G-PATE, by combining GAN framework with the PATE mechanism in which it ensures privacy property on the information flow from the discriminator to generator. In this model the generator is differentially private, not the entire GAN, since the generator is the only part that is published for data generation. Kaggle credit card fraud detection dataset, MNIST and Fashion-MNIST

image datasets were chosen for this experiment. Due to the results, this model beats DP-GAN and PATE-GAN in terms of data utility. The results showed better performance compare to the prior works image and non-image datasets and works well on more complex image which DP-GAN is almost incapable of performing it well.

3.1.11 WGAN

In Wasserstein GAN the authors focused on a number of techniques to measure how closely the model distribution matches the real distribution. WGAN increases the model's stability during training and offers a loss function that correlates with the quality of the generated data. This method Provide realistic learning curves useful for debugging and hyperparameter searches, eliminate problems like mode collapse and make learning more stable. In WGAN, instead of discriminator, a critic is introduced. The Wasserstein distance is predicted by the critic network, which has a similar design to a discriminator network but is optimizes to find the value to maximize the Wasserstein distance. The training process is more reliable and less sensitive to model architecture and selection of hyperparameter configurations with the WGAN. The loss of the discriminator appears to be related to the generator's ability to provide high-quality data, which is maybe most significant. [1]

3.1.12 ADS-GAN

To mitigate the risk of breaching patient confidentiality, Yoon et al. [51] proposed ADS-GAN (anonymization through data synthesis using generative adversarial networks) which is the modification of the conditional GAN framework. They performed a quantifiable, mathematical definition for "identifiability" to discuss the sufficient anonymization of the data. ADS-GAN generates the components based on optimizing the conditioning set for each patient and in this model the conditioning variables are not pre-determined. This step makes the improvement of the data quality as well as preserving the patient's identity by making sure that no combination of features can reveal those identities. The Meta-analysis Global Group in Chronic Heart Failure (MAGGIC) and The United Network for Organ Transplantation (UNOS) dataset[4] were selected for this experiment. They evaluated the marginal distribution mismatch for each feature as well as how well the generated synthetic data preserves the joint distribution of the original data. This framework was compared to PATE-GAN, DP-GAN, MedGAN, and WGAN-GP and the authors claimed that ADS-GAN outperformed all these models and it is the best solution to open data sharing of EHR-type datasets.

[4]https://unos.org/data/

3.2 Privacy Preserving Methods

There are some privacy-preserving methods that are currently using broadly. Anonymization techniques remove the sensitive attributes. K-anonymity [42] which is still used in the healthcare world, is a method in which the QID(attributes like ZIP code, gender, age, etc.) is modified with the same modified QIDs and equivalence class with respect to the concept of "equivalence class of records" that says one record is similar to at least k-1 other records in the same equivalence class concerning their QIDs. However, in this method, other sensitive attributes can be recovered and it is open for homogeneity and background knowledge attacks (attribute disclosure).

To tackle this problem, l-diversity [29] was introduced to present more guarantee to preserve the existing leakages in K-anonymity. Yet, this method is effective in protecting categorical attributes and it is not developed for continuous sensitive attributes. moreover, the sensitive attributes can breach if the global distribution of it is identified. Li et al. [26] proposed a novel approach called t-closeness such that "it requires that the distribution of a sensitive attribute in any equivalence class is close to the distribution of the attribute in the overall table "(i.e., the distance between the two distributions should be no more than a threshold t)". It overcomes the possibility of re-identification attacks by constructing equivalence classes of sensitive values. Also, data perturbation is about adding noise or altering the values. It can perturb continuous and categorical values.

3.3 Data synthesizing Methods Summary

The effectiveness of the medGAN method was assessed by comparing it with different versions of the medGAN and several other generative methods. The different versions of the medGAN that were assessed are GAN (the same architecture as medGAN with the standard training strategy, but do not pre-train the autoencoder), GANP (pre-train the autoencoder and use minibatch discrimination), GANPA (pre-train the autoencoder and use minibatch averaging) and the generative methods were Random Noise, Independent Sampling, Stacked RBM, and Variational Autoencoder. In medGAN they pre-trained the autoencoder, used minibatch averaging, used batch normalization, and a shortcut connection for the generator G. To generate synthetic patients records, medGAN was capable to generate high-dimensional discrete variables better compared to other tested methods. Still medGAN does not generate multimodal continues variables and the similarity of the real data and generated data had some flaws [8].

Table-GAN tried to solve the problem of generating synthetic data for a tabular dataset. In Park et al. experiment the table-GAN method using deep learning techniques was employed on the Health dataset which consists of various information (such as blood test results, questionnaire survey, diabetes, and so on). Comparing to state-of-the-art anonymization, perturbation, and generation

techniques, table-GAN only showed a consistent balance between privacy level and model compatibility since the model compatibility is crucial and as it stated, table-GAN showed the better performance accomplishing that. Yet, it does not support generating other data types such as strings.

Tabular data includes discrete and continuous columns. The multi-modal values within each continuous column and the imbalance of categorical columns made it hard to model this type of data. Conditional Tabular GAN (CTGAN) tackled these challenges and it outperformed MedGAN, VeeGAN, Table-GAN.

PATE-GAN framework ensures the (differential) privacy of the generator of the Generative Adversarial Nets (GAN). As it is stated by the authors, this model "can be used for generating synthetic data on which algorithms can be trained and validated, and on which competitions can be conducted, without compromising the privacy of the original dataset." Their experiment was on various types of data including sound classification. For the authors, Further research is to examine with Extending PATE to the regression setting, whether Wasserstein GAN can be used instead or not.

G-PATE is an approach for training a differentially private data generator based on the Private Aggregation of Teacher Ensembles (PATE) framework. In this mode the student generator is connected with an ensemble of teacher discriminators, and a private gradient aggregation mechanism is used for securing the differential privacy of all the flowing information from teacher discriminators to the student generator. The privileges of using G-PATE over PATE-GAN and DP-GAN are improvement of using privacy budget by using it in part of the model that needs to published for data generation and, training the discriminator on real data.

The structure of Tabular GAN (TGAN) is close to table-GAN with some few fundamental differences. Table-GAN uses convolutional neural networks while TGAN uses recurrent networks. TGAN generates synthetic tables while simultaneously generating discrete and continuous variables. TGAN has been limited to generating only a single table with numerical and categorical features and needs to be developed for modeling sequential data and multiple tables. SynSys method was formed to generate synthetic time-series data that is composed of nested sequences having only a small amount of ground truth data.

The healthGAN method is based on the medGAN architecture idea with the combination of Wasserstein GAN gradient penalty (WGAN-GP) to solve the existing issues with medGAN which are generating only binary and unrealistic data. According to the comparison of 5 other data generation methods which are Gaussian Multivariate, Parzen Windows, Additive Noise Model (ANM), Differential Privacy-preserving data obfuscation (DP), and Copy the real data (CP), the HeathGAN method claimed as a novel approach with better performance among these other aforementioned methods. It improved the medGAN algorithm and performed a better metrics for evaluating the quality of synthetic health data. But this version of the HealthGAN does not provide time-series data.

CorGAN which utilizes the convolutional generative adversarial networks was successful in generating both discrete and continuous synthetic values and

it outperformed DBMs, VAE and medGAN methods. From what has been discussed we can achieve that medGAN has some flaws which current methods tried to resolve. The TGAN is the improved version of table-GAN. This method plus CTGAN outperforms the table-GAN method and both methods are successful in generating discrete and continuous variables. HealthGAN and CorGAN are other useful methods in generating discrete and continuous synthetic values. SynSys is useful in generating synthetic time-series data.PATE-GAN focuses on the privacy of the generator of GAN whilst generating synthetic data and G-GAN has improved the PATE-GAN and DP-GAN model and works well for both image and none-image data.

Among different approaches that can be utilized to generate synthetic data, generative adversarial networks are one of the most successful methods in generating high quality samples. Exploring these studies helped us to get familiar with the layout and functionality of each method, and discover the advantages and disadvantages of each generative model. After these investigations, we can decide what are the most relevant and applicable methods for generating realistic synthetic patient records

Chapter 4

Proposed methods

4.1 Selected GAN methods for tabular data

Among all the studied GAN models, we needed to select the models which meet the requirements for our data type. The choices were narrowed down to the models in which they fit tabular data the best, generate numerical and categorical data with similar distribution as the real data, to handle missing values and to produce missing value in the same scale in the generated sample, manage an imbalanced dataset, as well as preserving privacy in a way that revealing the information from the sample won't be feasible. After the models comparison in section 3.3, we picked the models which outperformed the others in order of generating the tabular samples. The models for our experiments are TGAN, WGAN, ADS-GAN and, CTGAN, as well as VanillaGAN. The reason of choosing VanillaGAN is that we want to have the basic model of GAN so that the other models can be relatively compared to the base model.

4.1.1 Dataset:

To evaluate the selected GAN models, a real dataset was used to set-up a benchmarking system. The adult cencus income provided by Kaggle dataset repository[1].

The dataset consists of 32561 rows and 15 columns. Attribute characteristics involve ordinal, categorical, and numerical values. The aim of this dataset is to predict whether the person's income exceeds $50,000 a year based on its attributes. The detail of the dataset's features are described in the table 4.1.

4.2 Generated data quality check

After deciding which GAN models best fit the type and model of our dataset, it is needed to investigate the performance of all these selected models to realize

[1] https://www.kaggle.com/datasets/uciml/adult-census-incom

which one(s) are the perfect match for our data and generate data which meet high quality and privacy aspects. Many evaluation have been conducted to address these matters.

4.2.1 Similarity

Checking the similarity between the synthetic data and the real data is one of the quality check steps. **TableEvaluator**[2] is the chosen method for similarity evaluation of the real and the fake data. TableEvaluator is a python library in which it measure how real the fake data is. The given information in TableEvaluator consist of both plotting visual evaluation metrics and similarity score. The statistics that are calculated by TableEvaluater are as follows:

Mean Absolute Error (MAE): Mean Absolute Error evaluates the average of the amount of error in the measurement. This measurement is a useful method for comparing anticipated DATA with their actual outcomes. Hence, the smaller the value, the closer the real and fake data points are [9].

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}. \tag{4.1}$$

Where:
n = total number of data points,
y_i is the predicted value and x_i is the true value.

Euclidean distance: The Euclidean distance between two points measures the distance between those two points. Here, it measures the distance between the real values and the predicted values [9].

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{4.2}$$

Where:
p, q = two points in Euclidean n-space
q_i, p_i = Euclidean vectors
n = n-space

Root Mean Square Error (RMSE): This metric calculates the average distance between the predicted value and the actual value. In other words, how far the data points from the regression line (line of the best fit) are. Lower RMSE represents a better model that fits a dataset [9].

[2]The GitHub repository is available here: https://github.com/Baukebrenninkmeijer/table-evaluator

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} (\hat{x}_i - x_i)^2}{N}} \tag{4.3}$$

Where:
n = the number of observations
$\hat{x}_i$ = the predicted value
x_i = the actual value

Cosine similarity: Cosine similarity defines the similarity of the data objects irrespective of their size. In this metric data objects will be assumed as vectors and the cosine similarity is the cosine of the angle between them. Unlike Euclidean distance which is restricted by the size of the document, the two similar documents may increase similarity score in cosine similarity even if it is measured as being far apart by Euclidean distance. Smaller angle leads to the higher cosine similarity [22].

$$\text{cosine Similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4.4}$$

Where:
A_i and B_i are components of vector A and B.

Column correlation: The column correlation returns mean correlation between all columns of the two datasets. In tableEvaluater, the correlation of the categorical columns are evaluated by Theil's U and Cramér's V, and numeric columns are evaluated by Pearson correlation coefficient.

Cramér's V measures how deeply the relationship between the two categorical variables is. The range of Cramér's varies between 0 (no association) and 1 (complete association) [17].

$$V = \sqrt{\frac{\varphi^2}{\min(k-1, r-1)}} = \sqrt{\frac{\chi^2/n}{\min(k-1, r-1)}} \tag{4.5}$$

Where:
φ = the phi coefficient
χ^2 = Pearson's chi-squared test

n = the sample size
k = number of columns
r = number of rows

Theil's U or uncertainty coefficient measures the degree of nominal association between two variables. It is calculated from conditional distribution of joint distribution of the two variables. It indicates if the forecasting model is better than naive forecasting. Naive forecasting is a method that uses previous period to predict the next period. If the Theil's value is less than 1, the model is better than the naive forecasting, and if it is greater than 1, the model is worse. This statistic is helpful to remove methods with large error with magnifying the errors [44].

$$U(X|Y) = \frac{H(X) - H(X|Y)}{H(X)} \tag{4.6}$$

where:
$H(X)$ = entropy of a single distribution
$H(X|Y)$ = conditional entropy

Pearson correlation coefficient: Pearson's r measures the linear correlation between two datasets. The value varies between -1 and 1. 0 indicates no relation between two datasets and -1 or 1 denotes the same correlation [5].

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{4.7}$$

where:
x_i and y_i are the sample points from x and y vector
$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ = mean of vector x
$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ = mean of vector y
n = sample size

Jensen-Shannon Distance: The disparity between two probability distribution P and Q is calculated by this metric and it varies between 0 and 1, having 0 means the two distributions are the same. The probability distribution shows the likelihood of a random variable taking different possible outcomes. Q can be referred to as the fake samples distribution and, P as the real sample distribution. The J-S distance is derived by Kullback–Leibler divergence. unlike K-L divergence, The J-S distance value is symmetric and finite. Kullback–Leibler divergence is also measures the difference in two probability distributions [27]. It is calculated as follows:

$$D_{\mathrm{KL}}(P \parallel Q) = -\sum_{x \in \mathcal{X}} P(x) \log\left(\frac{Q(x)}{P(x)}\right) \tag{4.8}$$

Therefore, The J-S distance would be:

$$\mathrm{JSD}(P \parallel Q) = \frac{1}{2} D(P \parallel M) + \frac{1}{2} D(Q \parallel M) \tag{4.9}$$

Where:
$M = \frac{1}{2}(P + Q)$ is the mean of the P and Q distribution

Kolmogorov–Smirnov test: This metric compares the distribution of the fake and real samples by quantifying the distance between the empirical distribution functions of two samples and checks if they have equal underlying distributions [21].

Table evaluator calculated the F1 score on different machine learning models including Logistic Regression, RandomForestClassifier, DecisionTreeClassifier, and, MLPClassifier. Thus, the model with the highest F1 score will be the best model that makes the best predictions. For that reason, the evaluate method uses real dataset, split it into 80% train and 20% test dataset, train multiple machine learning models and calculate the F1 score on the 20% of the test data as well as the fake data. This procedure take place for the fake data as well. The fake data will be split into test and training sets, the models will be trained on the training set of the fake data and the F1 score will be calculated on the test set of the fake data and real data.

The similar and close results of F1 score for the fake and real data indicates that they have similar distribution and behaviour. Furthermore, the higher score represents that the machine learning model is performing better among others. Although a high value means a perfect score for similarity aspects, having a very similar generated data may raise questions in terms of the privacy and identifiability.

In the visual evaluation graphs of the table evaluators, CTGAN plots illustrated a fairly close distribution of the real and fake datasets for each column (Figures 4.1 till 4.8). The result of a F1 score shows that CTGAN has the best performance compared to the others. For instance, after training a RandomForest model on a generated sample, the accuracy on the real data is 0.82 which is also close to the accuracy of the generated test sample. The similarity score between real and generated samples is 0.93 (Table 4.3, 4.2).

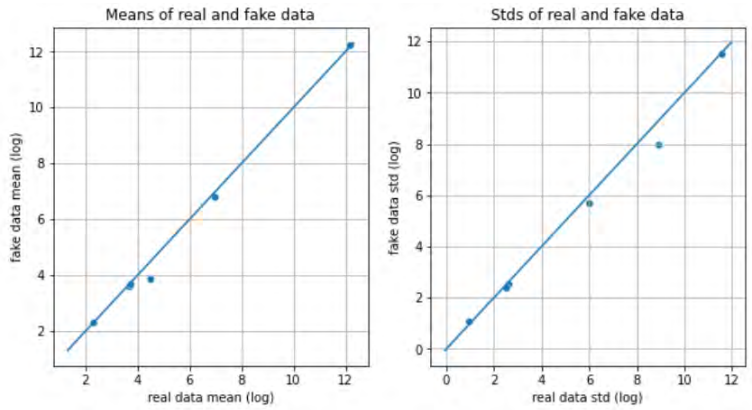


Figure 4.1: Absolute Log mean and STDs of numeric data for CTGAN

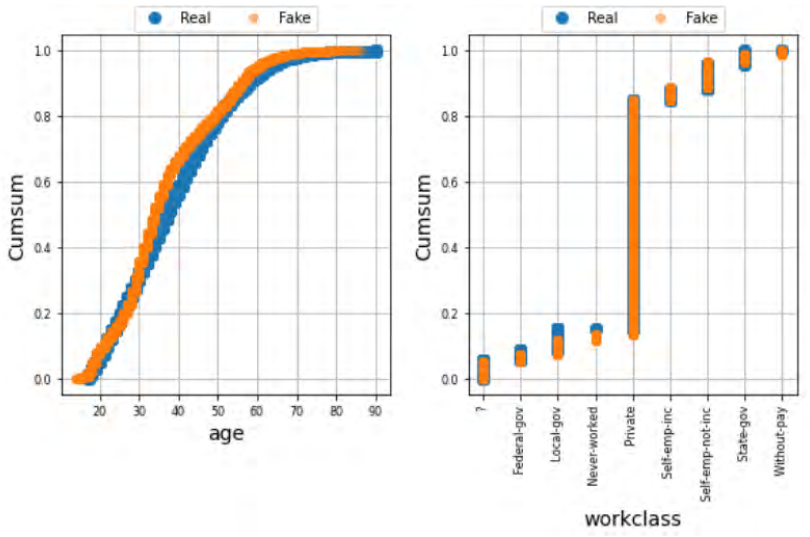


Figure 4.2: cumulative sums for age and work-class columns in the real and fake dataset for CTGAN

	CTGAN
Similarity Score	0.93
Column Correlation Distance RMSE	0.05
column correlation distance MAE	0.03

Table 4.2: Table evaluator statistical results on CTGAN

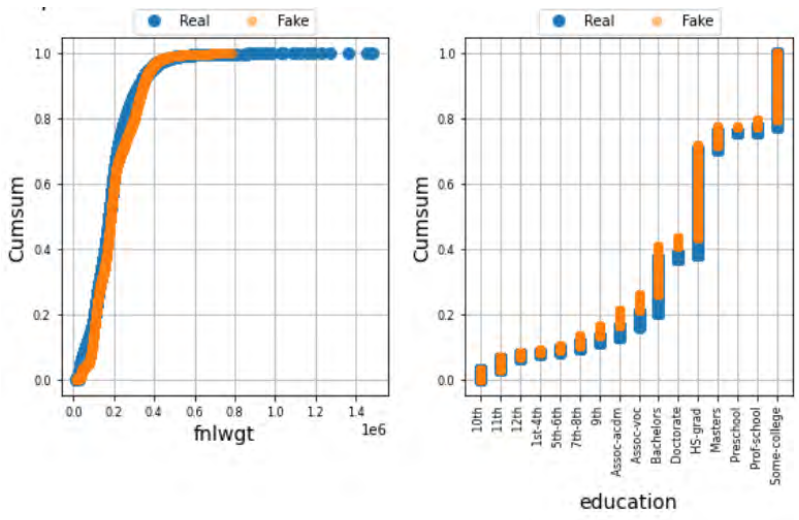


Figure 4.3: cumulative sums for final weight and education columns in the real and fake dataset for CTGAN

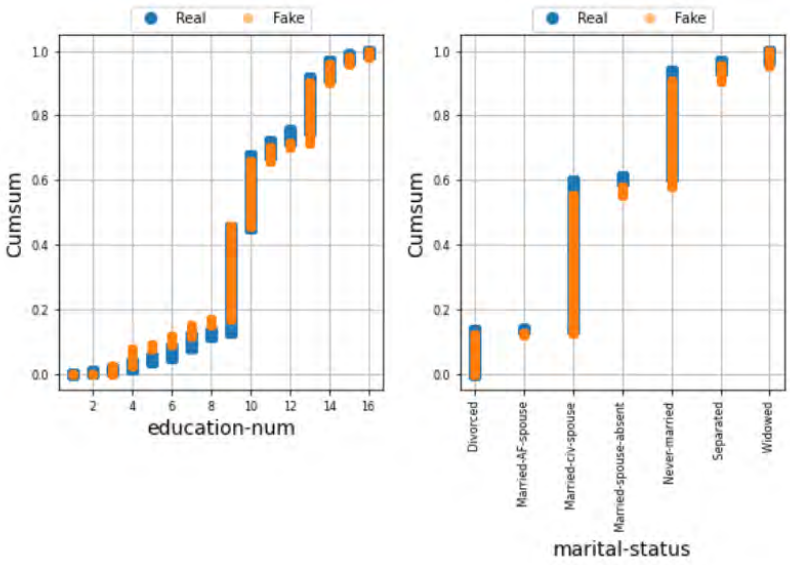


Figure 4.4: cumulative sums for edcation_num and marital_status columns in the real and fake dataset for CTGAN

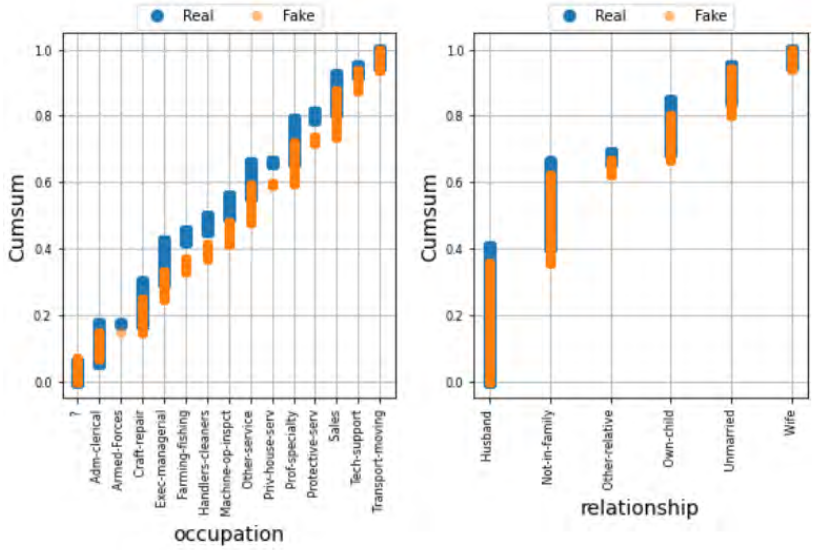


Figure 4.5: cumulative sums for occupation and relationship columns in the real and fake dataset for CTGAN

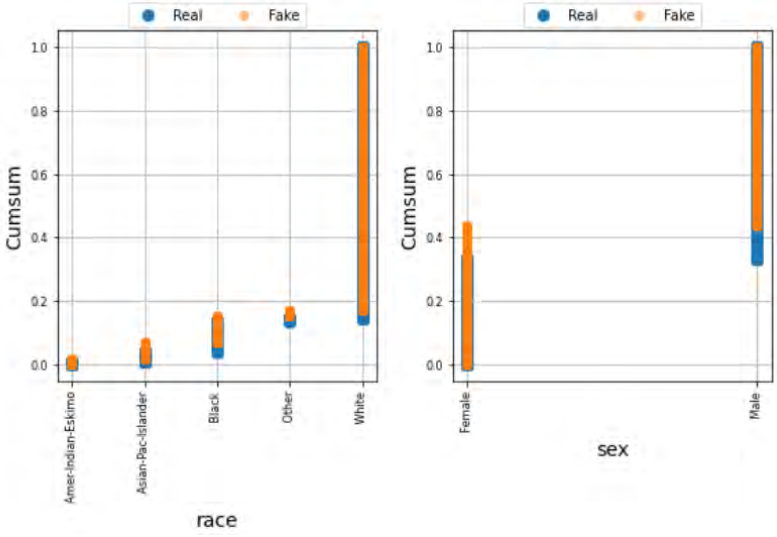


Figure 4.6: cumulative sums for race and sex columns in the real and fake dataset for CTGAN

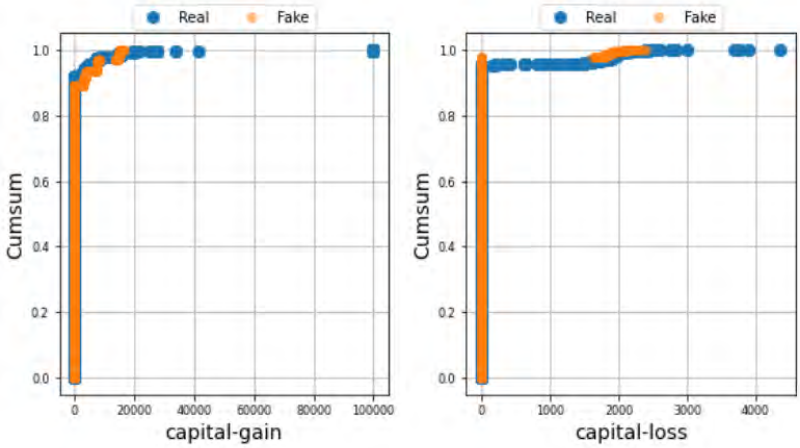


Figure 4.7: cumulative sums for capital_gain and capital_loss columns in the real and fake dataset for CTGAN

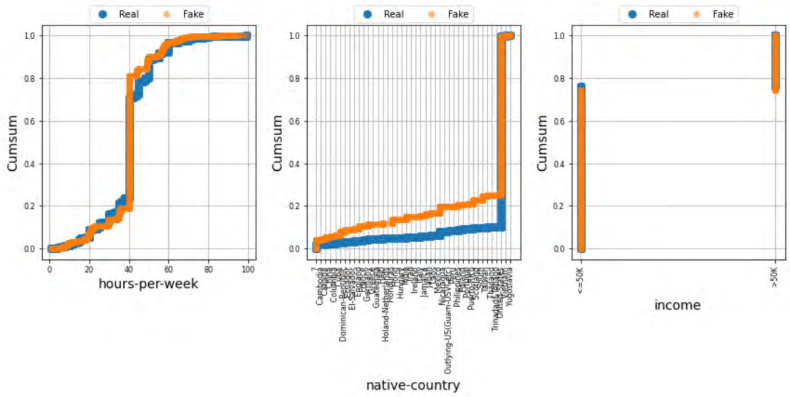


Figure 4.8: cumulative sums for hours_per_week, native_country and income columns in the real and fake dataset for CTGAN

4.2.2 Discriminator and generator losses

The difficulty of training the GAN models is due to the concurrent training of generator and discriminator in a way that progress in one model leads to cost for another model. Therefore, it is advantageous to monitor the possible failures in training GAN models. By tracking the discriminator loss for both real and the fake samples and, loss for the generator, the line plot of loss at the end of the training phase can be plotted. Evaluating the performance of the GAN, can

be done by tracking the performance of the generated samples and assess them [7, 18, 21].

This evaluation method was integrated into each GAN model's library. During the training process, the loss of the discriminator for real and fake samples and, loss of the generator have been reported each iteration for each model, the results have been saved and, used for creating the line plots of loss. Although it was attempted to implement the loss plot in the TGAN library as well, the output did not generate for this model.

The losses plots shows the performance of the GAN models during their training process (figure 4.9 till figure 4.12). These patterns' absolute values and time scales (such as the number of training epochs or iterations) will vary depending on the problems and the type of GAN model.

The Vanillagan loss plots (figure 4.9) shows many fluctuations in discriminator and generator losses during the training process and this is due to the bad performance of this method in generating the samples. There is no clear pattern and the loss for the generator oscillate over time which represents mode collapse, meaning there are many identical generated examples.

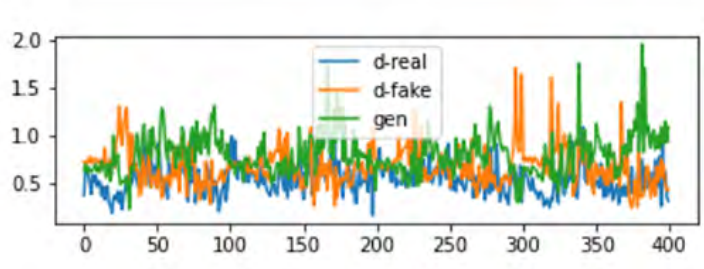


Figure 4.9: Generator loss and Discriminator loss on real and fake data on VanillaGAN

CTGAN plot shows a good trend since when the training initiates, the loss for the discriminator is low and the generator loss is high because the initial phases of data generation produce low-quality data. As it progresses, the discriminator loss increases as the loss of the generator decreases. This trend occurs in CTGAN loss plot figure 4.10.

In WGAN the losses start at zero. The discriminator loss of the real data is stable, but the variance between discriminator loss for fake data and generator loss is increasing (Figure 4.11).

In ADS-GAN after epochs 200, the losses start to stabilize, although the

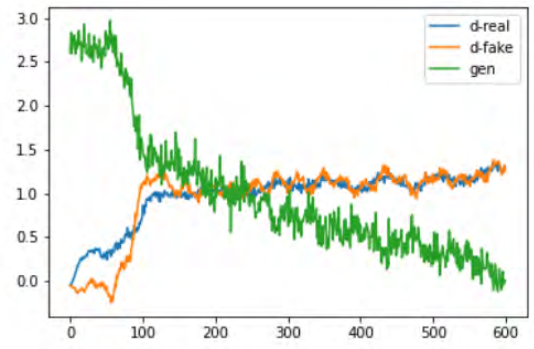


Figure 4.10: Generator loss and Discriminator loss on real and fake data on CTGAN

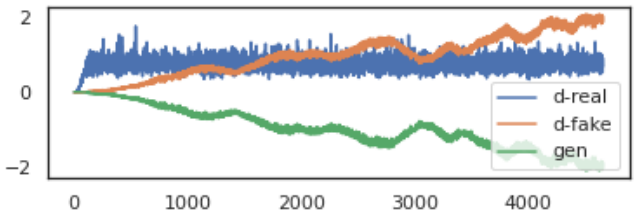


Figure 4.11: Generator loss and Discriminator loss on real and fake data on WGAN

variance increases between the losses (Figure 4.12).

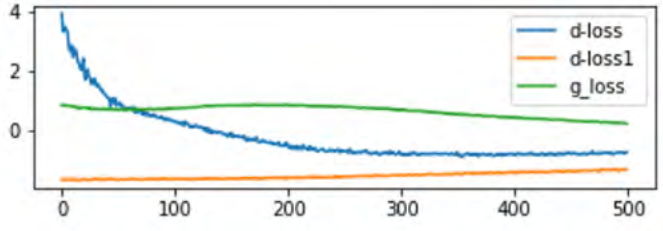


Figure 4.12: Generator loss and Discriminator loss on real and fake data on ADS-GAN

4.2.3 PRDC evaluation

Precision and recall are evaluation metrics for estimating the quality and coverage of the generated samples by generative model [41]. If we consider the real distribution as P(X), and the generative model as Q(Y), the portion of Q(Y) that can be generated by P(X) would be defined as Precision, and the portion of P(X) that can be generated by Q(Y) is considered as recall.

The improved version of precision and recall addresses the drawbacks of the previous version. Hence in this version, The expected likelihood of fake samples compared to the real manifold is measured by precision, and the expected likelihood of real samples compared to the real fake manifold is measured by recall. In the equations 4.10 and 4.11, N and M are the number of real and fake data and $1_{(.)}$ is the indicator function [25].

$$precision := \frac{1}{M} \sum_{j=1}^{M} 1_{Y_j \in manifold(X_1,...,X_N)} \tag{4.10}$$

$$recall := \frac{1}{N} \sum_{i=1}^{N} 1_{X_i \in manifold(Y_1,...,Y_M)} \tag{4.11}$$

The manifolds are defined as:

$$manifold\left(X_1, ..., X_N\right) := \bigcup_{i=1}^{N} B\left(X_i, NND_k\left(X_i\right)\right) \tag{4.12}$$

Where $B\left(x, r\right)$ represents the sphere in $\mathbb{R}^D$ around x with radius r, and $NND_k\left(X_i\right)$ represents the distance between X_i and its k^{th} nearest neighbor among $\{X_i\}$ excluding itself.

Nonetheless, the improved precision and recall still have some flaws including failing to detect the match between two identical distributions, being weak against outliers and, randomly selection of the hyper parameters evaluation. By using k-nearest neighbor distances instead of the k-means and the uniform-density assumption, the probability density functions are computed in this metric.Density counts the number of real-sample neighbourhood spheres contain fake data Y_j (equation 4.13), where k is the k-nearest neighbourhoods. The highest the density the better it is.

$$density := \frac{1}{kM} \sum_{j=1}^{M} \sum_{i=1}^{N} 1_{Y_j \in B(X_i, NND_k(X_i))} \tag{4.13}$$

Coverage counts the percentage of real samples in areas where at least one fake sample is present. The range of coverage is 0 to 1, the higher the value the better coverage it has. (equation 4.14).

$$coverage := \frac{1}{N} \sum_{i=1}^{N} 1_{\exists\ j\ s.t.\ Y_j \in B(X_i, NND_k(X_i))} \tag{4.14}$$

Figure 4.13 and 4.14 demonstrate the advantage of using density over precision and coverage over recall. [31]

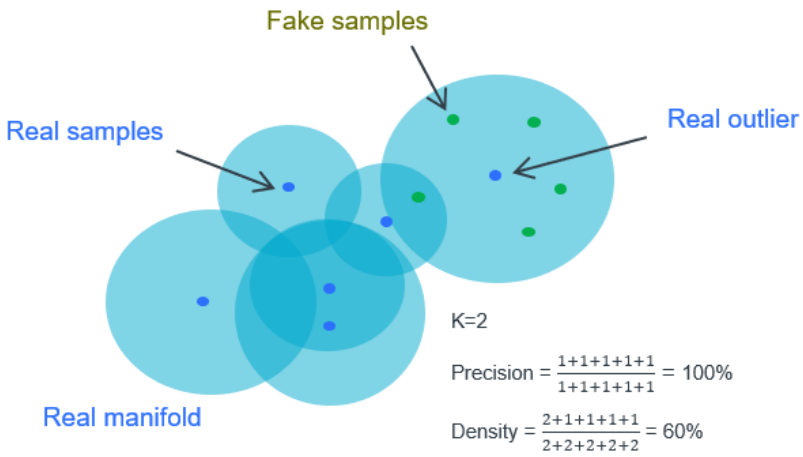


Figure 4.13: Precision versus density [31]

PRDC evaluation has been performed on three outperformed GAN models, CTGAN, TGAN and, ADG-GAN, as well as the basic model, VanillaGAN. This evaluation starts with smaller sample of the Adult dataset. The first 5,000 samples have been selected for all the methods and PRDC evaluation has been performed on them. This experiment was conducted with three different

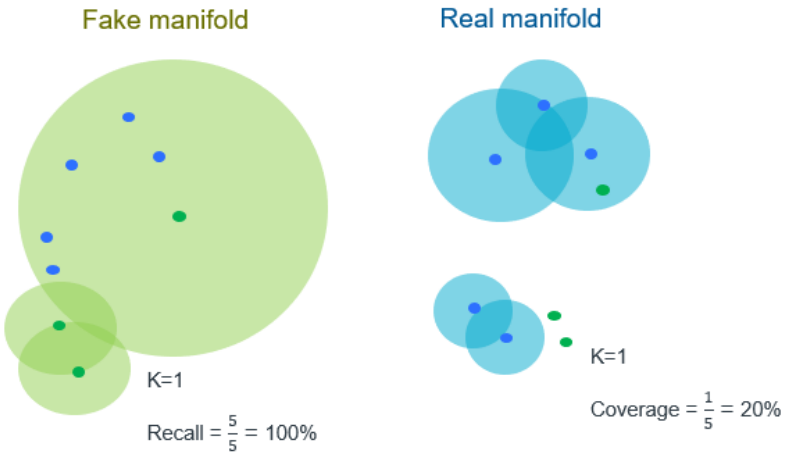
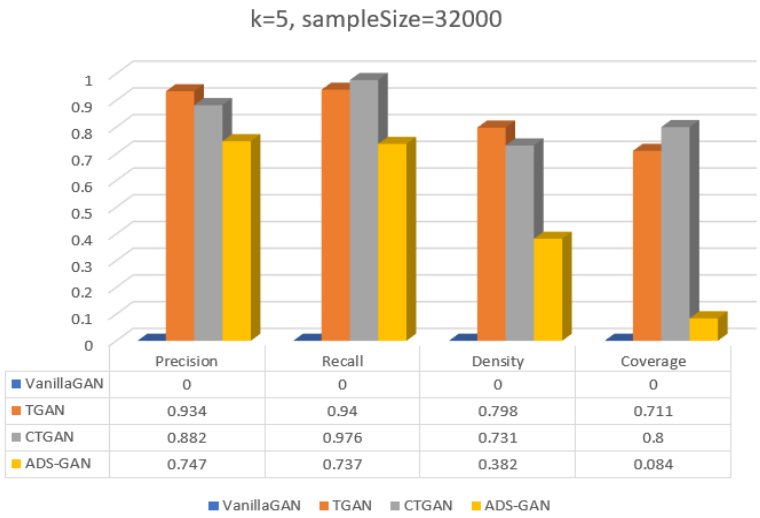


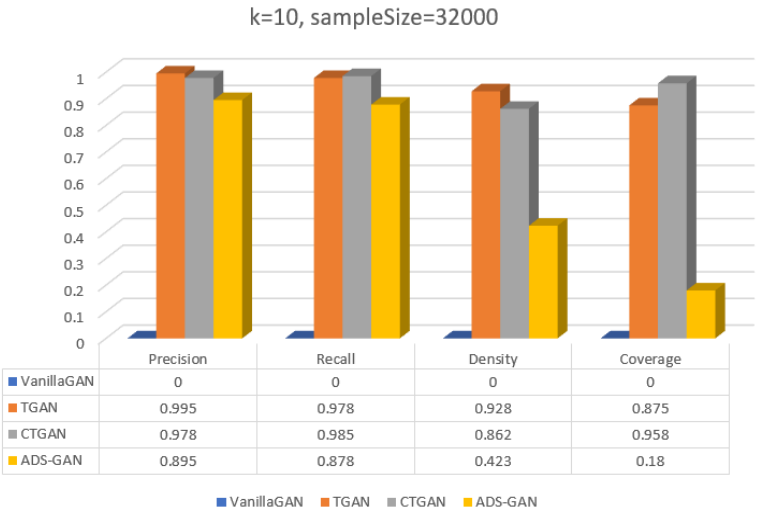
Figure 4.14: Recall versus coverage [31]

nearest neighbour values; 5, 10 and 20 (see figures 4.15 till 4.20). In almost all of the experiments, besides having a high precision and recall value, CTGAN has the highest density and coverage overall and after that TGAN has the closest best result. For instance, in figure 4.18, the density and coverage for CTGAN is 96.6% and 91.2% respectively, and for TGAN it is 87.3% density and 81% coverage.



	Precision	Recall	Density	Coverage
VanillaGAN	0	0	0	0
TGAN	0.934	0.94	0.798	0.711
CTGAN	0.882	0.976	0.731	0.8
ADS-GAN	0.747	0.737	0.382	0.084

Figure 4.15: PRDC evaluation on whole adult dataset with K value equal to 5



	Precision	Recall	Density	Coverage
VanillaGAN	0	0	0	0
TGAN	0.995	0.978	0.928	0.875
CTGAN	0.978	0.985	0.862	0.958
ADS-GAN	0.895	0.878	0.423	0.18

Figure 4.16: PRDC evaluation on whole adult dataset with K value equal to 10

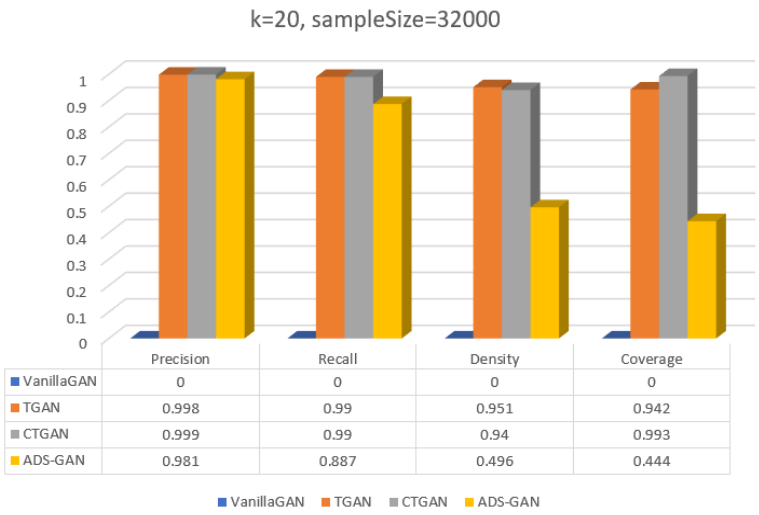


Figure 4.17: PRDC evaluation on whole adult dataset with K value equal to 20

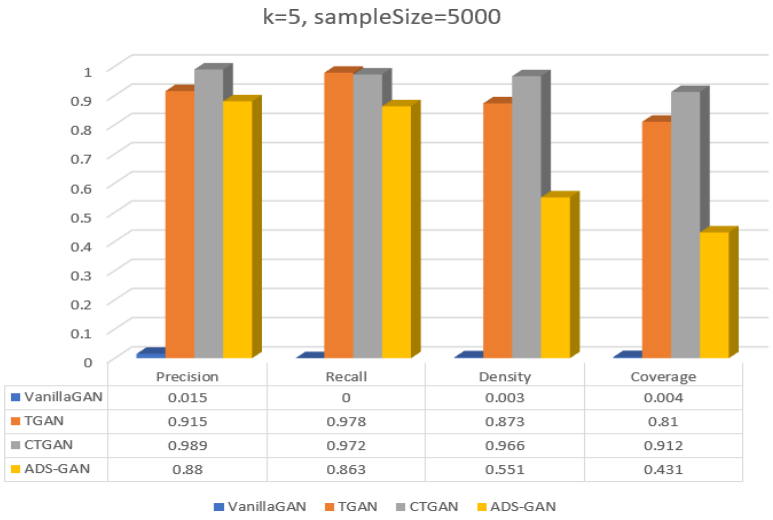


Figure 4.18: PRDC evaluation on 5000 sub-samples of the adult dataset with K value equal to 5

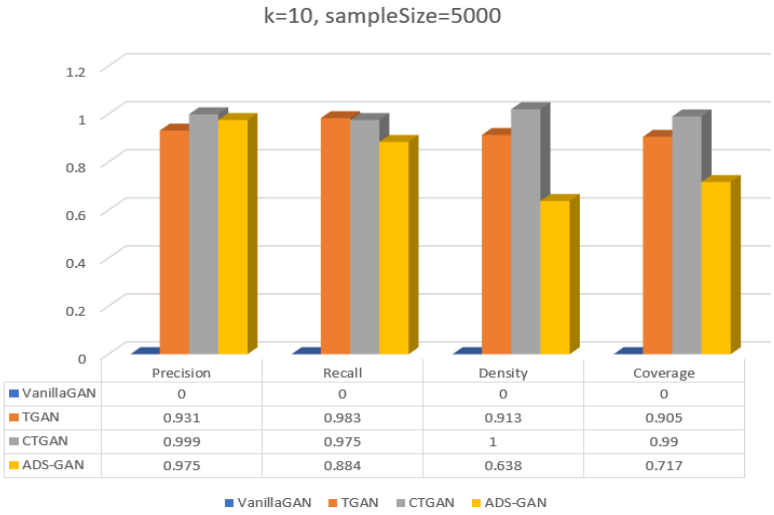


Figure 4.19: PRDC evaluation on 5000 sub-samples of the adult dataset with K value equal to 10

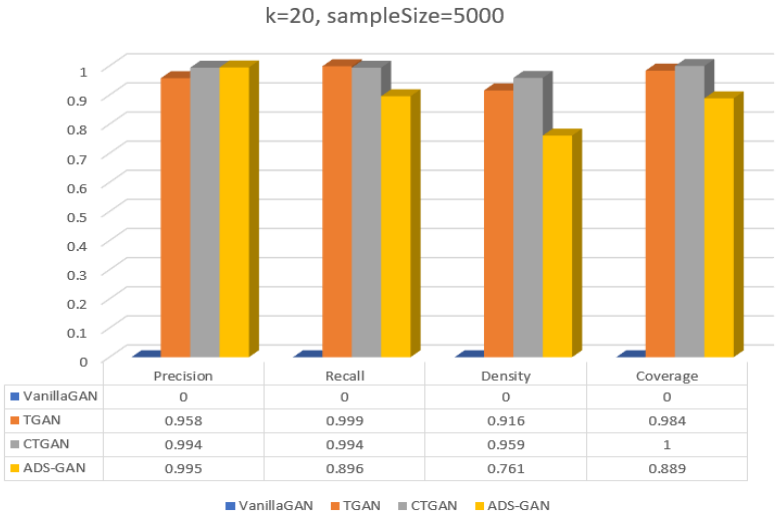


Figure 4.20: PRDC evaluation on 5000 sub-samples of the adult dataset with K value equal to 20

4.2.4 Privacy evaluation

The privacy aspects of the information in the health industries has significant value since the information involves a lot of private clinical information of the patients. Therefore, examining the generated data for the privacy sake is crucial.

The General data protection regulation (GDPR) defined the "personal data" as "any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person" [3].

There are different types of data identifiability. Direct identifiers include critical information such as name or unique individual numbers. Indirect identifiers contains less evident information. Potentially identifiable includes information which by having extra data will lead to the re-identification of an individual [39, 51].

Differential privacy (DP) has been able to address many issues regarding the confidentiality and privacy preserving in computer science. Differential privacy which consists of statistical and machine learning analysis is a criterion of privacy protection.

Under the view of DP, information is categorized in two classes, general information and, private information. The statistical analysis of DP ensures the privacy protection against a wide range of privacy attacks. It also guarantees the problem of "composition" which is about reaching the conclusion in data through the results of multiple analyses using information about the same individual.

Yet, DP can only assure the privacy preserving when general information and private information are determined. It only protects the private information and, if the general information includes any confidential information, then it will not be protected anymore. Therefore, DP fails to handle critical concepts in medical and health research [15, 47, 51].

ADS−GAN [52] privacy measurements checks that the patient's identity can not be exposed from any combination of the features in the synthetic data and with the combination of all the data on any individual, it defines the identifiability based on the likelihood of the re-identification of these data. In the original data, each of the two different observations are distinguishable since they are different individual, so they are "different enough". Accordingly, the minimum distance between the original observations is used as a measurement for "different enough" between the synthetic and original data.

To calculate the minimum distance between two observations in terms of identifiability, weighted Euclidean distance is used instead of simply using Eu-

[3] https://edps.europa.eu/data-protection/dataprotection/glossary/p_en

clidean distance, since some features are not as frequent as others, thus they are more identifiable. The calculation of the term "different enough" for the observation x_i and other original observations in D is as follows:

$$i = min_{x_j \in D/x_i} ||w(x_i - x_j)|| \tag{4.15}$$

Where:
D/x_i = dataset D without x_i
w = weight vector

Similarly the minimum distance between x_i and the generated observations in $\hat{D}$ is as follows:

$$\hat{r}_i = min_{\hat{x}_j \in \hat{D}} ||w(x_i - \hat{x}_j)|| \tag{4.16}$$

Having r_i and $\hat{r}_i$, the ϵ-identifiability definition can be described as "the synthetic data are different enough from the real data when synthetic data do not reveal the $\geq 1 - \epsilon$ ratio of the real data" [52]. Thus, a completely non-identifiable dataset would be equivalent to 0-identifiability and a perfectly identifiable dataset to 1-identifiability. It can be concluded that the $\hat{D}$ is ϵ-identifiable from D when:

$$\mathcal{I}\left(D, \hat{D}\right) = \frac{1}{N} \left[\mathbb{I}(\hat{r}_i < r_i)\right] < \epsilon \tag{4.17}$$

Where:
$\mathbb{I}$ = Identity function

The outliers are more subjected to be identified. Therefore, more restrictions has been applied by adding more noise. To define the weight vector for the definition of weighted Euclidean distance, discrete entropy has been employed. The discrete entropy of the i-th feature is:

$$H\left(X^{(i)}\right) = -\sum_{x^{(i)} \in \mathcal{X}^{(i)}} P\left(X^{(i)} = x^{(i)}\right) \log(P\left(X^{(i)} = x^{(i)}\right)) \tag{4.18}$$

The weight of the identifiability is the inverse of the entropy since the entropy denotes the uncertainty of the feature and higher uncertainty results in smaller identifiability.

$$\mathbf{w} = (w_1, ..., w_d) = \left(\frac{1}{H\left(X^{(1)}\right)}, ..., \frac{1}{H\left(X^{(d)}\right)} \right) \tag{4.19}$$

The generated samples need to meet the requirements of the identifiability constraints since calculating the maximum value of the loss (LD) which measures the Wasserstein distance between the joint distributions of original and synthetic datasets itself does not guarantee the identifiability constraints. Thus, the identifiability loss has been described as:

$$\mathcal{L}_{\mathcal{I}} = \mathbb{E}_{x,\hat{x}|x} \left[-||\mathbf{w}.\left(\mathbf{x} - \hat{\mathbf{x}}\right)|| \right] \tag{4.20}$$

With the condition that the generated sample $\hat{x}$ is "different enough" from the original sample, this equation (4.21) depicts the generator's attempt to maximize the weighted Euclidean distance between the two samples. Thus, the generator tries to minimize the loss ($max_D[\mathcal{L}_{\mathcal{D}}]$) and identifiability loss ($\mathcal{L}_{\mathcal{I}}$). Hence, the final optimization problem is:

$$\begin{aligned} min_G = \lfloor max_D \lfloor \mathcal{L}_{\mathcal{D}} \rfloor + \lambda \mathcal{L}_{\mathcal{I}} \rfloor \\ = min_G \lfloor max_D \mathbb{E}_{\mathbf{x}\sim\mathcal{P}_x, \mathbf{z}\sim\mathcal{P}_z} \lfloor D(\mathbf{x}) - D\left(G\left(\mathbf{x},\mathbf{z}\right)\right) \\ -\eta \left(||\nabla_{\hat{x}} D\left(\hat{\mathbf{x}}\right)||_2 - 1 \right)^2 - \lambda \mathbb{E}_{\mathbf{x},\hat{\mathbf{x}}} \left[||\mathbf{w}.\left(\mathbf{x} - \hat{\mathbf{x}}\right)|| \right] \rfloor \end{aligned} \tag{4.21}$$

$\lambda > 0$ is a hyperparameter which straighten out how much information can be presented in the way that it would be sufficient as well as not being too much so they can be identifiable.

These metrics are applied to the selected GAN methods to calculate the identifiability value of each of these methods. λ was set to 0.1, the mini-batch size to 128 and, the process ran for 1000 iterations. At the first trial the identifiability value for VanillaGAN was 0. This was due to the poor performance of the vanillaGAN in a way that the generated data does not have the similar distribution of the real samples, After normalizing the real and the generated samples this value changed to 3.11e-5, which still denoted a low performance. Among other GAN extensions, TGAN with the lowest value of 0.009 outperformed the other methods in term of the identifiability. Nevertheless, the CTGAN and ADS-GAN had considerable performances as well (Table 4.4).

GAN Method	VanillaGAN	ADS-GAN	TGAN	CTGAN
Value	3.11e-5	0.019	0.009	0.245

Table 4.4: Identifiability measure for GAN methods

4.2.5 Summary

In summary, after evaluating different GAN methods, CTGAN, TGAN, WGAN and ADS-GAN were selected as the most suitable methods that can generate synthetic tabular data preserving privacy of the real samples and be able to handle the imbalanced samples and those containing null values. Afterward, the synthetic data were analysed by different evaluation metrics. These experiments were conducted on the adult census dataset available on the Kaggle data repository.

According to the F1 score and the table evaluator plots, CTGAN performs the best in terms of the similarity of the synthetic data and the real data. The PRDC evaluation results showed that CTAG and thereafter TGAN have the best performance among others. In ϵ-identifiability metric from ADS-GAN the TGAN performed the best compared to CTGAN and ADS-GAN. In discriminator and generator losses plots the ADS-GAN losses stabilized after 300 epochs.

We cannot declare for sure which GAN model outperformed the best among others. Accordingly, the methods in which they have the best performance in terms of similarity and privacy were selected to be performed on the real dataset. Therefore, the CTGAN was chosen since it has the best similarity performance among others, and after that TGAN was selected because it also performed good in terms of similarity and privacy.

Attribute	**Data Type**	**Description (possible values)**
Age	Continuous	Age of the person
Work Class	Categorical	Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked
Final Weight	Continuous	"Number of units in the target population that the responding unit represents"
Education	Categorical	Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool
Education Number of Years	Continuous	Number of Years of education in total
Marital-status	Categorical	Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse
Occupation	Categorical	Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspect, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces
Relationship	Categorical	Role in the family (Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried)
Race	Categorical	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black
Sex	Categorical	Female, Male
Capital-gain	Continuous	income from investment sources other than wage/salary
Capital-loss	Continuous	income from investment sources other than wage/salary
Hours-per-week	Continuous	Hours of work in every week
Native-country	Categorical	United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinidad Tobago, Peru, Hong, Holland-Netherlands

Table 4.1: Adult data set description

	Real	Fake
real_data_LogisticRegression_F1	0.80	0.78
real_data_RandomForestClassifier_F1	0.84	0.81
real_data_DecisionTreeClassifier_F1	0.80	0.77
real_data_MLPClassifier_F1	0.77	0.77
fake_data_LogisticRegression_F1	0.77	0.80
fake_data_RandomForestClassifier_F1	0.82	0.84
fake_data_DecisionTreeClassifier_F1	0.75	0.78
fake_data_MLPClassifier_F1	0.80	0.77

Table 4.3: Table evaluator Classifier F1scores on CTGAN

Chapter 5

Experiments on the real data

This chapter presents the experiments on the real data set and their results.

5.1 Data Description

The dataset has been provided by Fürst Medical laboratory which is results of patient's blood test. Permission to use these data for research purposes has been granted to Fürst by REK (the regional committee for medical and healthcare research ethic) under the project *Anonymous and accurate health data synthesis using deep learning*, project number 259224. The data are stored in a secured server located at Fürst, and made accessible for analysis only to the participants in the REK-approved project. In this thesis , I have shown only descriptive and aggregate data, which do not disclose individual patient records.

The dataset has 952685 rows and 26 columns. Each row represents a requisition (a set of analysis/tests ordered by a doctor and performed at the laboratory), reporting the gender, the results of 11 blood tests and whether such results are pathological or not. There is also a label/class which categorizes each patient as healthy ("Euthyroid") or as affected by a pathology/condition ("Hyperthyroidism" and "Hypothyroidism" are the main ones).

The main test results are those reporting the values of three hormones which regulate the function of the thyroid gland: TSH, T4, T3 (columns 4-6). The values of eight other blood markers are also reported.

In this dataset, 14 features out of 26 were chosen for this experiment, the other features were explanations and flags of these 14 main features. This dataset contains 2 categorical and 12 numerical features. Although the ID did not contain any private information on it, the ID column was also dropped for the further experiments. The data description are represented in detail in table 5.1.

Column	Col Name	Description
1	ID	a generic ID (string)
2	gender	0 Male, 1 Female (integer)
3	age	age of the patient (integer)
4	TSH	Reporting the result of the test measuring the level of TSH in blood (numeric)
5	T4	Control in the treatment of thyroid diseases (numeric)
6	T3	Free triiodothyronine, Assessment of thyroid function (numeric)
7	Kreatinin	Measurement of how concentrated the urine is (numeric)
8	HDL	High density lipoprotein, assessment of risk for cardiovascular disease (numeric)
9	LDL	Low density lipoprotein, assessment of risk for cardiovascular disease (numeric)
10	Kolesterol	Assessment of risk for cardiovascular disease (numeric)
11	Urinstoff	Assessment of protein and amino acid turnover. Fluid balance. Kidney function. Assessment of the degree of toxicity in uraemia (numeric)
12	VB12	Vitamin B12 (numeric)
13	Alat	Liver diagnostics, toxic liver damage (numeric)
14	Label	8 classes-conditions/pathologies (string)

Table 5.1: Fürst dataset description

5.2 Exploratory Data Analysis (EDA)

In this chapter, we want to explor the data analysis and inspect the distribution of the features and check how balanced the label feature is. Multiple libraries have been used to do the AutoEDA.

5.2.1 Pandas Profiling

The pandas profiling[1] is an open source library in python which is used for data analysis and represent different information of univariate and multivariate report of the dataset such as mean, medaian, correlation matrix, etc.

5.2.2 SweetVIZ

SweetViz[2] is also an open source python library which illustrates the target characteristics, analyzes the train and test data, correlation of the variables and

[1] https://pandas-profiling.ydata.ai/docs/master/index.html
[2] https://pypi.org/project/sweetviz/

the target value, etc.

5.2.3 AutoViz

AutoViz[3] is another python open source library that visualize the information of the dataset shape in heat map, bar chart, pair plot etc.

In the plots from Autoviz library, the distribution and skewness of each feature are represented. See figures 5.1 till 5.9. These graphs demonstrate uneven distributions of the dataset features. Moreover, the label feature figure 5.10 illustrated a very imbalanced distribution for different classes.

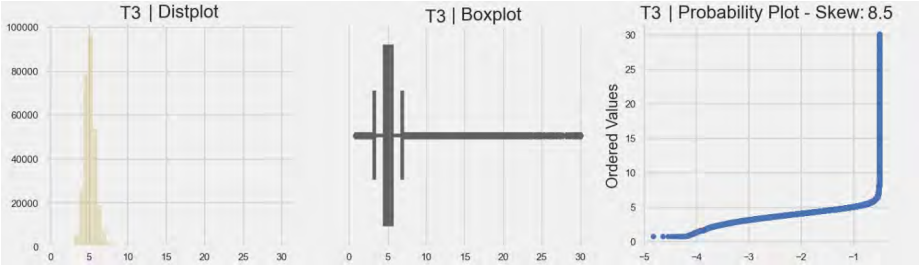


Figure 5.1: Distribution plot, boxplot and probability plot-skew of T3

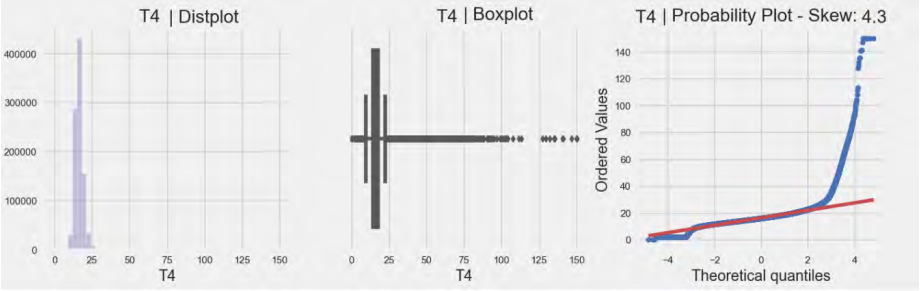


Figure 5.2: Distribution plot, boxplot and probability plot-skew of T4

The EDA clearly showed that we have an imbalance dataset. To simplify the problem, first, it was chosen to make two subsets of the data. First subset consists of "Euthyroid" and ("Hypo" or "Sub-Hypo") in which all the Sub-hypo were replaced by Hypo. And the second subset involves "Euthyroid" and ("Hyper" or "Sub-Hyper"). The SubHyper were substituted with Hyper. At second step, the GAN model was trained with all the 8 label classes.

[3]https://github.com/AutoViML/AutoViz

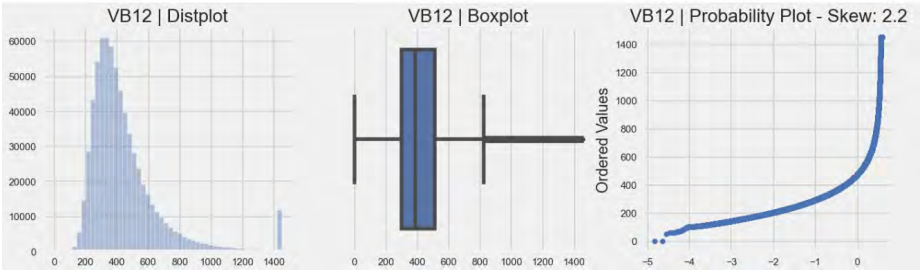


Figure 5.3: Distribution plot, boxplot and probability plot-skew of VB12

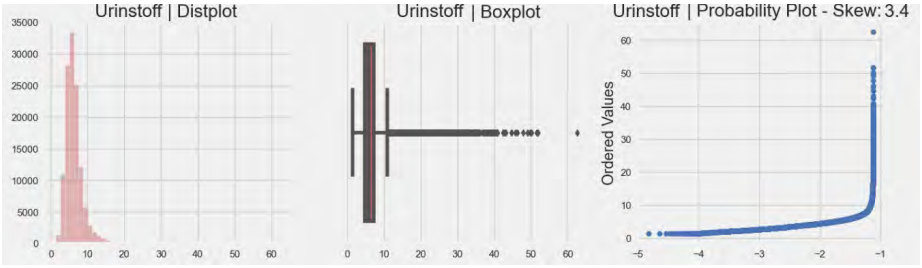


Figure 5.4: Distribution plot, boxplot and probability plot-skew of Urinstoff

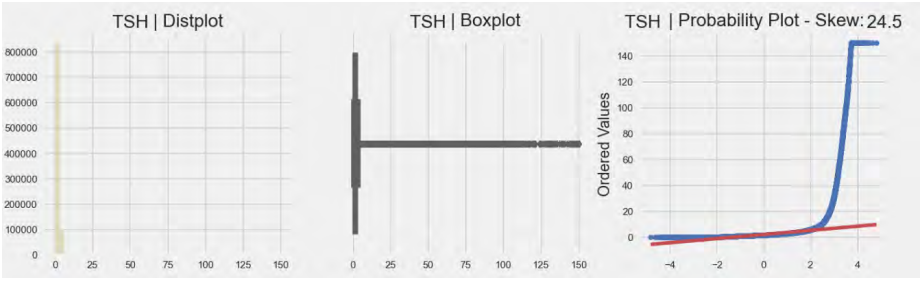


Figure 5.5: Distribution plot, boxplot and probability plot-skew of TSH

5.3 Generated data quality check

In this section all the evaluation metrics on both CTGNA and TGAN are described in details.

5.3.1 Similarity

The results of the table evaluator give us a good insight of how close the real data and the generated data are, both visually and statistically. The plots shows

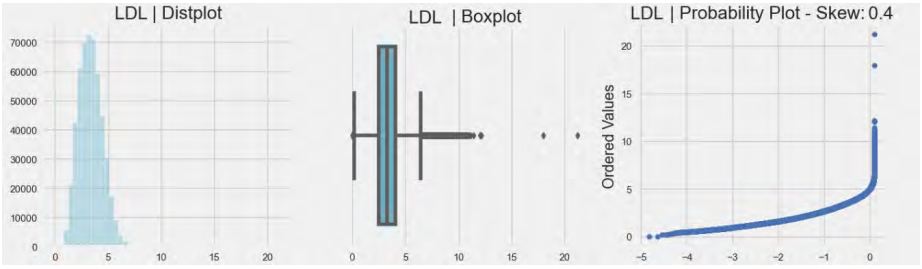


Figure 5.6: Distribution plot, boxplot and probability plot-skew of LDL

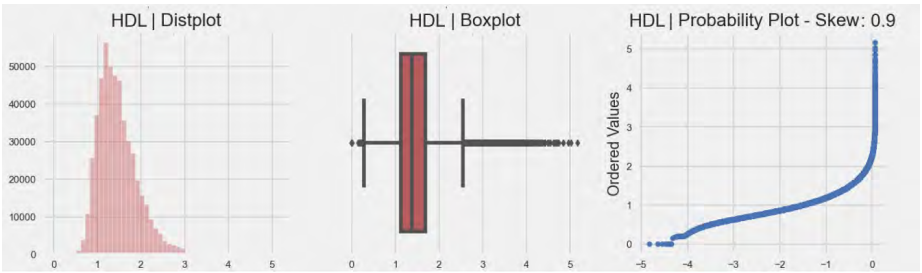


Figure 5.7: Distribution plot, boxplot and probability plot-skew of HDL

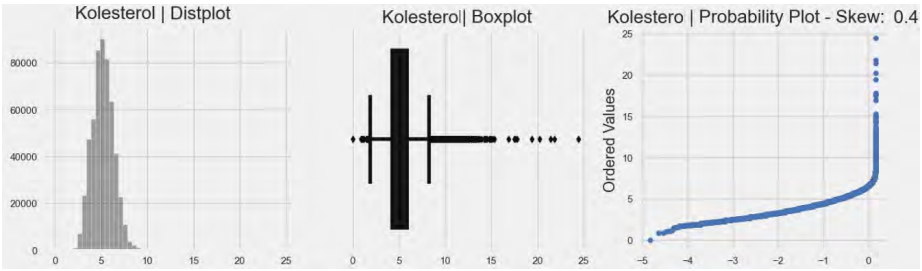


Figure 5.8: Distribution plot, boxplot and probability plot-skew of Kolesterol

almost overlaps in the distribution of the real and the generated data features in both CTGAN and TGAN, so, the distributions are quite the same.

F1 score is the weighted average of precision and recall which contains both false positive and false negative predictions. One of the benefit of F1 score is that it is able to be performed on the imbalance data as well[4], which suits our dataset. In cases where the datasets are severely skewed, a model which performs poorly and only predicts the majority class will appear accurate based

[4]https://stephenallwright.com/good-f1-score/

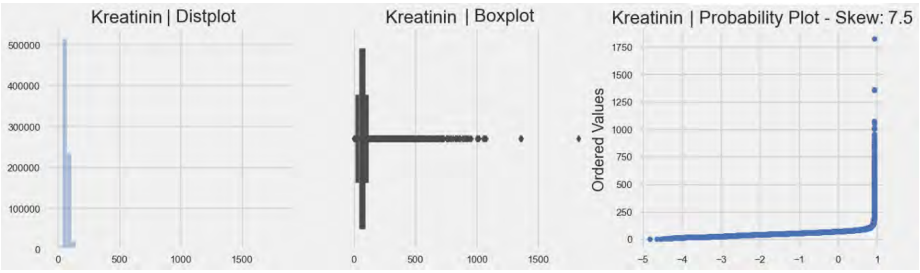


Figure 5.9: Distribution plot, boxplot and probability plot-skew of Kreatinin

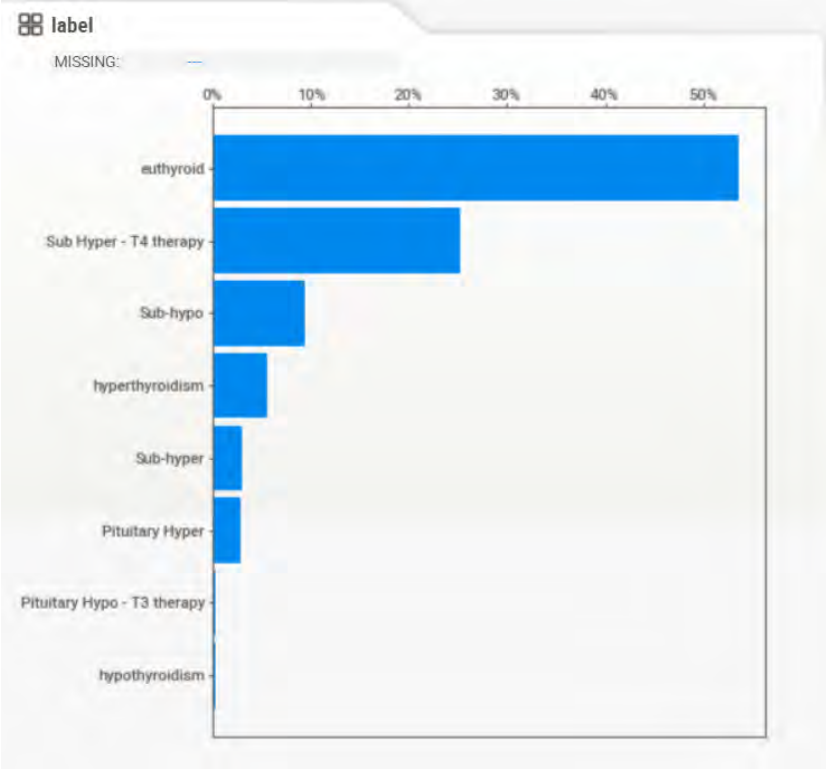


Figure 5.10: Distribution of the label feature

on other metrics, such as accuracy. The model won't have excellent precision or recall on the positive class, hence the full level of under-performance will be revealed in the F1 score (equation 5.1).

The F1 score has a range of 0 to 1, with 0 denoting the poorest possible result and 1 denoting a flawless result, meaning that the model accurately predicted

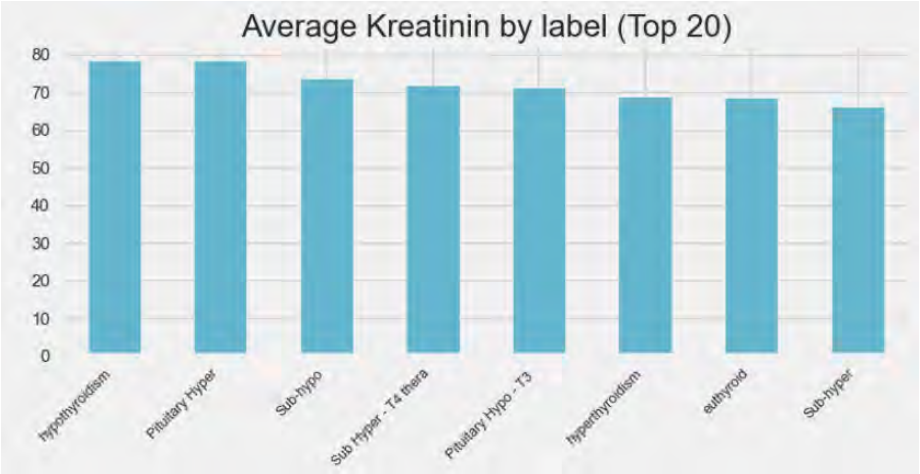


Figure 5.11: Bar plot for Kreatinin by label

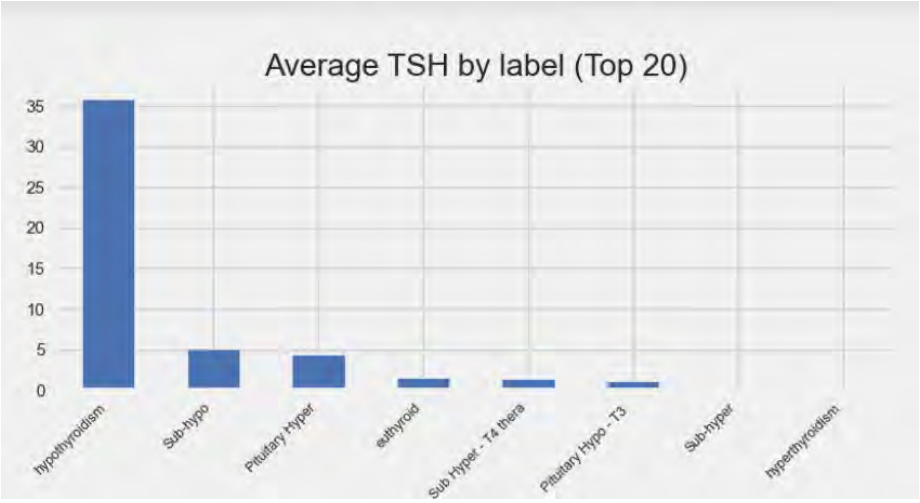


Figure 5.12: Bar plot for TSH by label

each observation.

$$\mathcal{F}1Score = 2 * (Recall * Precision)/(Recall + Precision) \qquad (5.1)$$

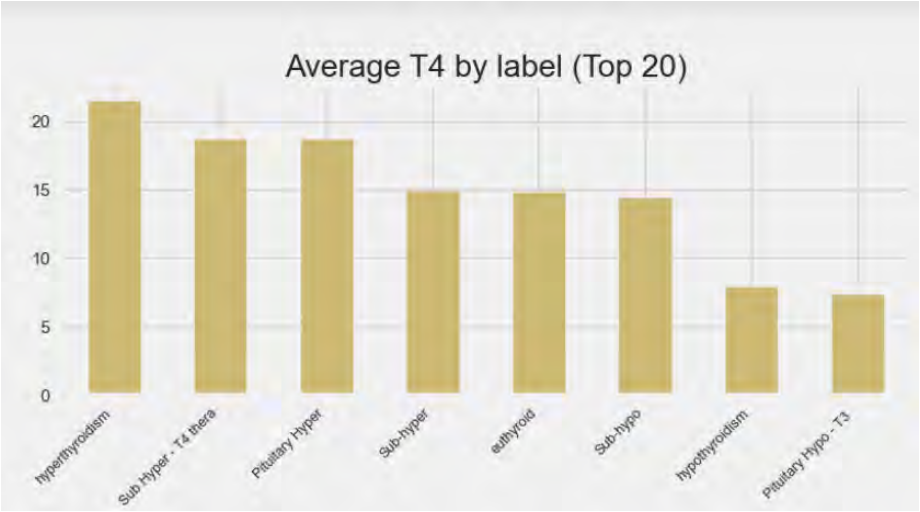


Figure 5.13: Bar plot for T4 by label

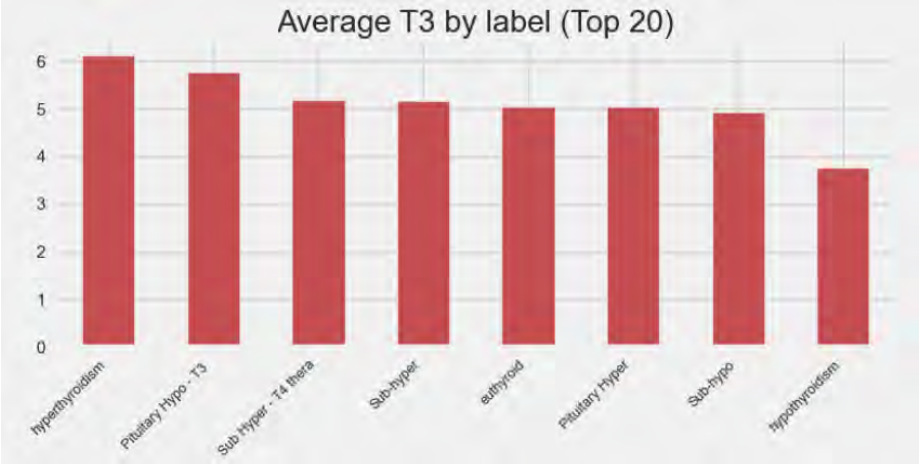


Figure 5.14: Bar plot for T3 by label

Table evaluator results for CTGAN

Whole dataset: The F1 score shows a similar behaviour and high score accuracy both for real and generated sample with different machine learning models. More precisely, if we train a model on the generated data, it will be able to predict great on the real dataset as well, which means it has a close distribution to the real data. The similarity score is 0.94 which represents very high similarity

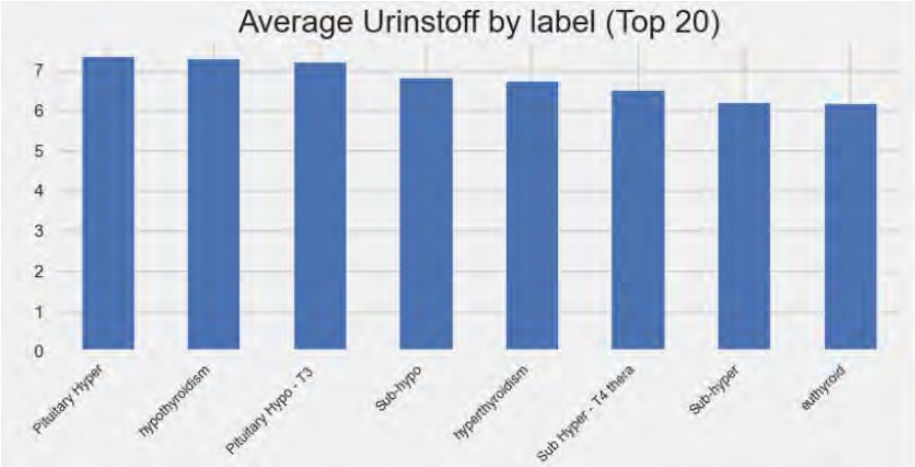


Figure 5.15: Bar plot for Urinstoff by label

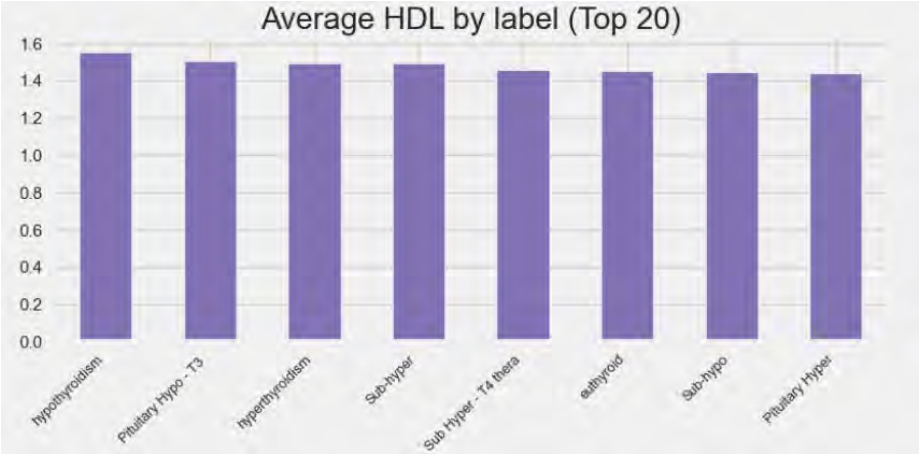


Figure 5.16: Bar plot for HDL by label

between real and generated samples. The overall column correlation distance root mean square error RMSE between real and generated sample is 0.04 and the mean absolute error MAE is 0.03 which approve the high resemblance of the real and the generated data (Table 5.2, 5.4).

sub_hypo: Similarly, the F1 score for the hyper subset were high and represented a high accuracy in predicting the generated sample after training on the real data and vice versa. The similarity score is 0.96, RMSE between real and generated sample is 0.04 and MAE is 0.02, which indicates a well generated

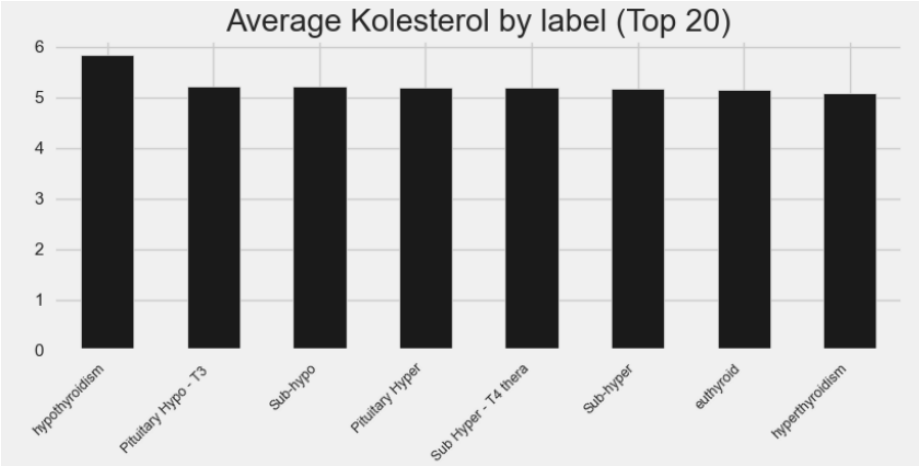


Figure 5.17: Bar plot for Kolesterol by label

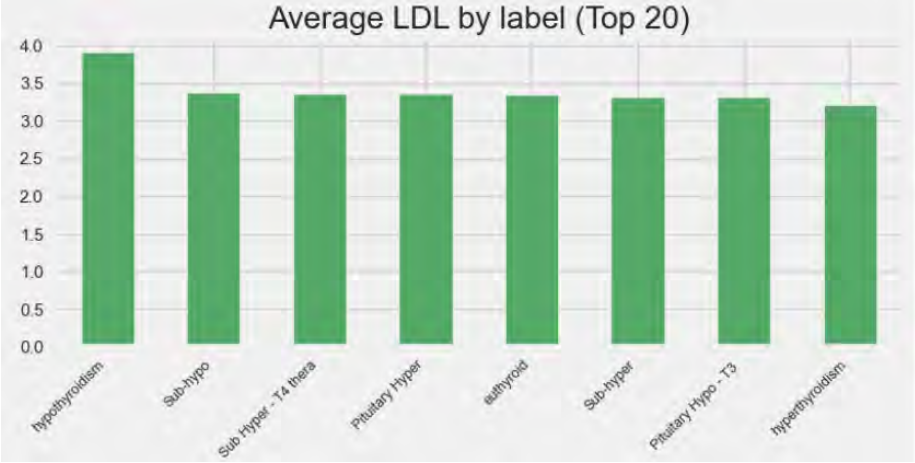


Figure 5.18: Bar plot for LDL by label

synthetic sample similar to the real dataset distribution (Table 5.2, 5.4).

sub_hyper: This subset has a very similar results to the subset of hypothyroidism. The results showed a similar and high accuracy performance for real and generated sample.

The overall review of the table evaluator F1 score and the statistical scores indicates a slightly better performance of hypothyroidism and hyperthyroidism

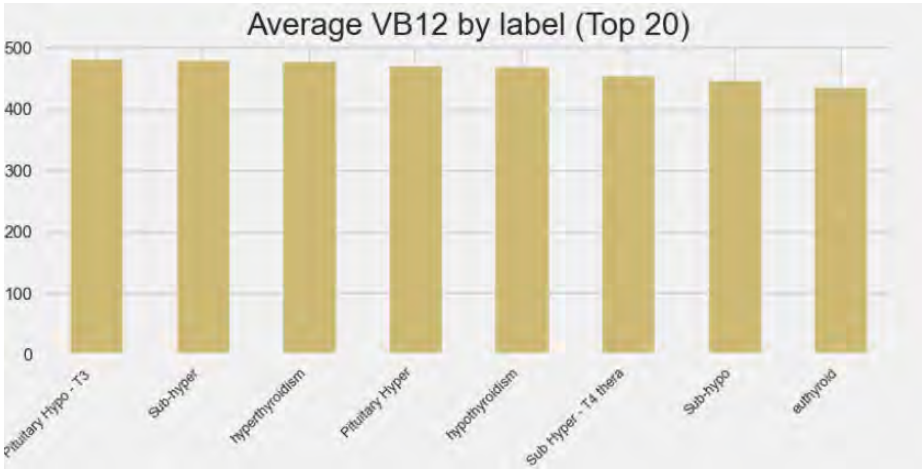


Figure 5.19: Bar plot for VB12 by label

subsets compared to the whole dataset (Table 5.2, 5.4).

Table evaluator results for TGAN

Whole dataset: The results represented a modest performance in terms of similarity. The F1 score related to the machine learning models' results showed that after training the model on the generated sample, it cannot predict well on the real dataset, which implies not a very close distribution to the real sample. For instance, after training a decision tree classifier on the generated sample, the accuracy on the test set of the generated sample was 0.51 which is like a random guess, and the accuracy on the real sample was 0.14 which indicates a very bad result and the similarity score is 0.40 (Table 5.3, 5.4).

Sub_hypo: According to F1 score for this subset, after training different machine learning model on real sample, the overall evaluation shows that it has a great performance on the test set of the real sample, but it did not performed well on the generated sample. For example, after training a decision tree classifier model on the training set of the real sample, the accuracy on the test set of the real sample was 1, while it was 0.24 on the generated sample which denotes the poor performance of the model. Similarly, after training the decision tree classifier on the generated sample, the accuracy on the test set of the generated sample was 0.88 which represents a good performance, but it was 0.34 accurate on the real dataset.

The similarity score between the real and generated sample is 0.43 in TGAN which is not a perfect result in case of similarity. The overall column correlation distance root mean square error RMSE between real and generated sample is 0.13 and the mean absolute error MAE is 0.07 which denotes some differences

in the these two datasets (Table 5.3, 5.4).

Sub_hyper: This dataset shows the best performance among all other sets in TGAN. After training a random forest model on the generated sample, the accuracy of the prediction on the real data was 0.81, and the similarity between the real and the generated sample is 0.69 (Table 5.3, 5.4).

5.3.2 Synthetic Data Vault (SDV)

It was decided to experiment more evaluation metrics for the synthetic data. Synthetic Data Vault (SDV) [36] is a set of libraries in which the users can generate new data from single-table, multi-table and timeseries datasets. Moreover, it has various methods to evaluate the quality of the synthetic data. SDV framework is benefited from deep learning based techniques and multiple probabilistic graphical modeling.

SDV statistical metrics:

These metrics do the one by one feature comparison between the real and the synthetic data. two of these metrics are used in this evaluation, KSTest and CSTest.

- sdv.metrics.tabular.KSTest: KSTest compare the distribution of the continuous columns by using two-sample Kolmogorov–Smirnov test and empirical CDF. Empirical CDF (ECDF) is the probability distribution of the sample generated by the data, and two-sample Kolmogorov–Smirnov check if the two samples have the same distribution. This metric measures the maximum distance between the expected CDF and the observed CDF value for each column in which it is 1 minus the KS test D statistic and the output value is the average score of these values. More precisely, it measures the likelihood that the two samples were taken from the same distribution in which the higher the value the similar the distributions are.

- sdv.metrics.tabular.CSTest: This metric compare the distribution of two discrete columns using Chi-Squared test. The output is the probability of those columns having the same distribution. Similar to KSTest, the closer it get to 0, the more likely it is to have a similar distribution.

These two statistical metrics were used both in CTGAN and TGAN. In each case, the statistical test will be run on all the compatible columns. So, categorical or boolean columns for CSTest and numerical columns for KSTest, and the results will be reported with the average score.

In CTGAN, the KSTest for continuous columns of all sets (whole dataset, hypo and hyper subsets) was 0.95 which represents a very similar distribution

	Real			Fake		
	ALL DATASET	SUB_HYPO	SUB_HYPER	ALL DATASET	SUB_HYPO	SUB_HYPER
real_data_LogisticRegression_F1	0.79	0.98	0.99	0.78	0.96	0.96
real_data_RandomForestClassifier_F1	1	1	1	0.81	0.96	0.96
real_data_DecisionTreeClassifier_F1	1	1	1	0.67	0.88	0.88
real_data_MLPClassifier_F1	0.97	0.99	0.99	0.85	0.97	0.97
fake_data_LogisticRegression_F1	0.70	0.91	0.91	0.71	0.91	0.91
fake_data_RandomForestClassifier_F1	0.76	0.91	0.91	0.77	0.91	0.91
fake_data_DecisionTreeClassifier_F1	0.76	0.91	0.91	0.69	0.87	0.87
fake_data_MLPClassifier_F1	0.76	0.91	0.91	0.79	0.91	0.92

Table 5.2: Table evaluator Classifier F1scores on CTGAN

	Real			Fake		
	ALL DATASET	SUB_HYPO	SUB_HYPER	ALL DATASET	SUB_HYPO	SUB_HYPER
real_data_LogisticRegression_F1	0.78	0.98	0.99	0.25	0.84	0.35
real_data_RandomForestClassifier_F1	0.99	1	1	0.44	0.79	0.86
real_data_DecisionTreeClassifier_F1	1	1	1	0.34	0.24	0.67
real_data_MLPClassifier_F1	0.98	0.99	0.99	0.52	0.80	0.82
fake_data_LogisticRegression_F1	0.29	0.34	0.34	0.54	0.88	0.88
fake_data_RandomForestClassifier_F1	0.14	0.28	0.80	0.58	0.93	0.93
fake_data_DecisionTreeClassifier_F1	0.14	0.28	0.81	0.51	0.92	0.92
fake_data_MLPClassifier_F1	0.14	0.28	0.77	0.59	0.91	0.92

Table 5.3: Table evaluator Classifier F1scores on TGAN

of the real and the generated samples. In TGAN this values is slightly less and the highest value is drawn in the whole dataset with 0.78 (Table 5.5).

The CSTest values for categorical column were quite the same for both CTAGN and TGAN, and in both cases the highest value was for the whole dataset with value 0.99. Merely There was a small difference in the hyper subset in which the value was 0.93 for CTGAN and 0.88 for TGAN (Table 5.5).

The overall review shows that the real and the generated samples in CTGAN have more similarities in distributions compared to TGAN.

SDV Detection Metrics:

This metric uses a machine learning model to assess how difficult it is to distinguish the synthetic data from the real data. Therefore, the real data and synthetic data will be shuffled along with the flags showing the real and synthetic data. Afterwards, the machine learning model will be cross validated by attempting to predict the flags. The output value of this metric will be 1 minus the average score of ROC AUC of all the cross validation splits. ROC AUD score represents how good the classification model is at predicting the classes.The higher the AUC score is, the better model is at prediction. Accordingly, in detection metrics, the higher the values is, the harder it is to distinguish the real and the synthetic data.

- sdv.metrics.tabular.LogisticDetection: this metric implements a LogisticRegression classifier from scikit-learn.

After implementing this metric on the CTGAN and TGAN, we noticed a massive difference between the results of these two methods. The value of the LogisticDetection metric in CTGAN was 0.78 for the whole dataset, 0.77 for hypo subset, and 0.83 for hyper subset. So, the hyper subset performed better compared to the other two sets while this value in TGAN was 0.07 for the whole dataset, 0.03 for hypo subset, and 0.06 for the hyper subset. This implies that it is much more difficult to distinguish the generated and the real data in CTGAN which indicate a very similar distribution for these two datasets (Table 5.6).

Efficacy Metrics:

By training a machine learning model on the synthetic data and then analyzing the score it receives when evaluated on the real data, these metrics will determine whether it is possible to solve a machine learning problem without using real data. This metric required a target column to run a classification model. In Fürst dataset, the target is the label column which indicates the whether the patient status is normal or not. The quality of our synthetic data and the difficulty of the Machine Learning problem we are attempting to answer, both affect the value produced by this metric.

	CTGAN			TGAN		
	ALL DATASET	SUB_HYPO	SUB_HYPER	ALL DATASET	SUB_HYPO	SUB_HYPER
Similarity Score	0.94	0.96	0.96	0.40	0.43	0.69
Column Correlation Distance RMSE	0.04	0.04	0.04	0.15	0.13	0.15
column correlation distance MAE	0.03	0.02	0.02	0.08	0.07	0.08

Table 5.4: Table evaluator statistical results on CTGAN and TGAN

	CTGAN			TGAN		
	ALL DATASET	SUB_HYPO	SUB_HYPER	ALL DATASET	SUB_HYPO	SUB_HYPER
KSTest	0.95	0.95	0.95	0.78	0.68	0.71
CSTest	0.99	0.89	0.93	0.99	0.91	0.88

Table 5.5: SDV statistical metric values on CTGAN and TGAN

- MulticlassDecisionTreeClassifier: This metric was implemented on the whole dataset since it has 8 labels in the target column.
- BinaryDecisionTreeClassifier: This metric was chosen for the two subsets, since they have two labels each.

After executing decision tree classifier metrics, the hypo and hyper subsets in CTGAN presented very good results with value 0.94 and 0.98 respectively which means it can successfully solve a machine learning problem without using real data. Although it was not a quite good result for the whole dataset which was 0.48. Similarly, the efficacy metric for hypo and hyper subsets in TGAN were 0.85 and 0.88 which also represented good results. Moreover, the whole dataset in TGAN showed a very bad performance with the value 0.13 which indicates that it was unable to train a machine learning model (Table 5.7).

SDV Privacy Metrics:

these metrics assess a synthetic dataset's privacy considering the notion of given the synthetic data, can an attacker identify sensitive dataset characteristics. To answer that, An adversarial attacker model will be fit on the synthetic data to predict the sensitive attributes. Afterward, the accuracy of the real data will be evaluated using this model.

Two other inputs are needed in this metric, the list of the private columns (sensitive_fields) and the list of columns that the prediction of the sensitive columns are based on(key_fields). Age and gender are considered as the key_field and, label as the sensitive_fields. Hence, we want to know if attackers have the age and gender of a patient, can they predict the status(label) of the patient.

These metrics are categorized into categorical metrics and numerical metrics. Based on the data type, one should be selected to implement the metric on the data and it can not handle data that contains both categorical and numerical data. Hence, the Numerical Linear Regression privacy metric and the Numerical Support-vector Regression privacy metric were chosen to evaluate the privacy of the datasets. Therefore, the label column in the Fürst dataset was label-encoded. Since these metrics do not allow missing data, we substituted 0 for each missing value before running them. Afterward, the numerical privacy metric was applied on the data. The output range of the metric is 0 and 1. A higher value represents higher score in privacy of the data.

The results of the CTGAN and TGAN for these metrics were relatively the same. in CTGAN the numericalLR and numericalSVR metrics value for the hypo subset were 0.06 and 0.05 respectively which indicate a none private dataset. Similarly the value for numericalLR was 0.04 and 0.05 for numericalSVR in the whole dataset. The hyper subset surpasses the other results of the CTGAN with the value of 0.27. Hence, this subset is more private compared to other sets. In TGAN also the hyper subset outperformed the other datasets in terms of privacy with the value 0.26 (Table 5.8).

	CTGAN			TGAN		
	ALL DATASET	SUB_HYPO	SUB_HYPER	ALL DATASET	SUB_HYPO	SUB_HYPER
LogisticDetection	0.78	0.77	0.83	0.07	0.03	0.06

Table 5.6: SDV Detection metric value on CTGAN and TGAN

	CTGAN			TGAN		
	ALL DATASET	SUB_HYPO	SUB_HYPER	ALL DATASET	SUB_HYPO	SUB_HYPER
DecisionTreeClassifier	0.48	0.94	0.98	0.13	0.85	0.88

Table 5.7: Machine Learning Efficacy Metrics value on CTGAN and TGAN

	CTGAN			TGAN		
	ALL DATASET	SUB_HYPO	SUB_HYPER	ALL DATASET	SUB_HYPO	SUB_HYPER
NumericalLR	0.04	0.06	0.27	0.04	0.05	0.26
NumericalSVR	0.05	0.05	0.27	0.06	0.05	0.26

Table 5.8: Machine Learning Efficacy Metrics value on CTGAN and TGAN

Chapter 6

Conclusions

6.1 Summary

In different medical contexts such as patient's medical records, accessing real and valuable data is severely limited due to the privacy requirements. One of the possible solutions is utilizing synthetic data which mimics the distribution of the real data and tries to generate samples with distribution as close as possible to the real data. There are several methods that can generate synthetic data. In this study it was aimed to investigate different GAN methods that are capable of generating tabular data. We explored different methods to find the most suitable method and to get a better understanding of the trade-offs between privacy and quality of the generated data. The methods were selected based on their ability to generate tabular data with similar distribution as the real data, handle missing values, manage an imbalanced dataset, as well as preserving privacy.

The selected GAN methods were CTGAN, TGAN, WGAN and ADS-GAN. These investigations were initiated with experiments on an open-access dataset, the Adult Census Income dataset provided by the Kaggle data repository. The reason for choosing this dataset is due to having ordinal, categorical, and numerical values of their attributes, same as our real dataset. In each GAN method, a model was fitted and from the learned model, new synthetic samples were generated. Afterwards, different evaluation metrics were implemented to explore the performance of each of the selected methods and check the quality of their generated data. The generated samples were examined in two aspects: 1) The statistical characteristics of the synthetic data match those of the real data, and 2) The leakage of private data from the model is insignificant.

The results on the analysis revealed that none of the methods outperformed the others in both considered aspects. Therefore the choices were limited to the overall performance score of the methods. Hence, CTGAN and TGAN were selected to be investigated on the real world Fürst dataset. The investigation

was carried on with the real private dataset. Models were fitted both in CTGAN and TGAN and the synthetic samples were generated by these models. Through rigorous evaluations it was determined that in respect of similarity, CTGAN performed superior to other GAN methods. In the matter of privacy, when we were evaluating the confidentiality of the data on the dummy dataset in section 4.2.4, the TGAN outperformed others with a high margin. While TGAN and CTAGN's outcomes in the SDV privacy evaluation metric were almost the same and CTGAN even performed slightly better.

Moreover, the generated datasets were not private enough based on the SDV privacy metrics. Nonetheless, the hyper subset performed far better for both CTGAN and TGAN compared to the other sets. When we created subgroups of the dataset, not many findings were significantly impacted in terms of similarity but the outcomes were affected concerning privacy. Compared to other sets, the hyper subset seems to be more private.

6.2 Contributions

In section 1.2, three research questions were raised to be considered in this thesis. The work is summarize by identifying the goals and explaining how this study answered the given problems.

- ***RQ 1:*** *What are the best GAN methods for generating synthetic tabular data?*

  This question was answered in chapter 3 where we studied different GAN methods and examined their structures and functionalities. We concluded that among all the explored methods, ADS-GAN, CTGAN, TGAN and WGAN appeared to be the most relevant GAN methods for generating synthetic tabular data that can follow the two characteristics of similarity and privacy.

- ***RQ 2:*** *How well do the proposed GAN methods perform in terms of data privacy and similarity?*

  These assessments were accomplished in chapter 4 where we conducted different evaluation metrics on the generated data by CTAGN, TGAN, WGAN, and DS-GAN methods. The evaluations were performed both for similarity and privacy inspections. According to the performed metrics in this chapter, none of the GAN methods were able to outperform in all the evaluation metrics Therefore, based on the overall score, CTGAN and TGAN were selected. Overall we can conclude that GANs are not privacy preserving by default and should be considered with care when used for highly sensitive data.

- ***RQ 3:*** *How effective are the chosen GAN methods for generating synthetic data from a real world dataset, namely the Fürst dataset?*

The two final selected GAN methods were used for the Fürs data. Since the data was imbalanced, all the experiments were conducted on three datasets: the whole dataset, and two subsets of the data, to counter the unbalancing of the data and investigate the results. After generating the synthetic data, additional assessments were performed for more in-depth evaluations. The SDV evaluation metrics were calculated to compare the real and the generated data. According to these results, CTGAN performed excellently concerning the similarity and the statistical properties of the synthetic data are equivalent to the real data. Concerning privacy, both CTGAN and TGAN failed to obtain a high score while it was shown in section 4.2.4 that TGAN can produce private synthetic samples. However, the hyper subset appeared to be more private compared to the two other sets (whole dataset and the hypo subset). This confirms our finding in RQ 2 that synthetic data generated with GANs is not privacy preserving by default.

We discussed the trade-offs of multiple methods and metrics, offered advice on factors to keep in mind when creating and utilizing synthetic medical data. Concluding we can say that different use cases and different metrics may result in different outcomes. Broadly, we have to make a compromise on what is the most important factor for generating our synthetic data. Generating too similar synthetic data may risk privacy concerns and we may have to increase the dissimilarities of the two datasets.

Chapter 7

Future work:

There are several directions that can be explored in the future work. In this project a thorough investigation of the existing methodologies and their evaluations was conducted.

One of the next experiments can be adding the evaluation metrics to the learning steps of the model. For instance implementing the generator and discriminator loss plots (discussed in chapter 4.2.2) in the learning steps of the generator and discriminator to monitor the losses trends and determine where to stop the training procedure. In addition, there could be methods to tune parameters to suit both similarity and privacy and not only create identical data.

Another interesting study can be generating time series synthetic data which is a collection of measurements that have been obtained repeatedly over time. Due to the requirement to preserve strict ordering and complex relationships between time and measurement results, time-series data has proven to be one area that has been particularly difficult for producing realistic synthetic data. Therefore investigating and evaluating methods in which they can produce high quality time series data can be conducted in the future.

Certainly, more methods and more evaluation metrics can be implemented to examine the quality of the generated data.

Bibliography

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

[2] Insaf Ashrapov. Tabular gans for uneven distribution, 2020.

[3] Taiwo Oladipupo Ayodele. Types of machine learning algorithms. *New advances in machine learning*, 3:19–48, 2010.

[4] Randolph C Barrows Jr and Paul D Clayton. Privacy, confidentiality, and electronic medical records. *Journal of the American medical informatics association*, 3(2):139–148, 1996.

[5] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. *Pearson Correlation Coefficient*, pages 1–4. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[6] Giuseppe Bonaccorso. *Machine learning algorithms*. Packt Publishing Ltd, 2017.

[7] Jason Brownlee. *Generative adversarial networks with python: deep learning generative models for image synthesis and image translation*. Machine Learning Mastery, 2019.

[8] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. In *Machine Learning for Healthcare Conference*, pages 286–305. PMLR, 2017. ISSN: 2640-3498.

[9] Geoffrey I. Webb (eds.) Claude Sammut. Encyclopedia of machine learning and data mining. 2017.

[10] Jessamyn Dahmen and Diane Cook. Synsys: A synthetic data generation system for healthcare applications. *Sensors*, 19(5), 2019.

[11] Fida K. Dankar and Mahmoud Ibrahim. Fake it till you make it: Guidelines for effective synthetic data generation. *Applied Sciences*, 11(5), 2021.

[12] Juan de Benedetti, Namir Oues, Zhenchen Wang, Puja Myles, and Allan Tucker. Practical lessons from generating synthetic healthcare data with bayesian networks. In *ECML PKDD 2020 Workshops*, Communications in Computer and Information Science, pages 38–47. Springer International Publishing, 2020.

[13] Richard O Duda, Peter E Hart, et al. *Pattern classification*. John Wiley & Sons, 2006.

[14] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, pages 1–12, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[15] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[16] Issam El Naqa and Martin J. Murphy. *What Is Machine Learning?*, pages 3–11. Springer International Publishing, Cham, 2015.

[17] Nina Gantert, Steven Soojin Kim, and Kavita Ramanan. Cramér's theorem is atypical. In *Advances in the Mathematical Sciences*, pages 253–270, Cham, 2016. Springer International Publishing.

[18] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks, 2017.

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, oct 2020.

[21] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[22] Jiawei Han, Micheline Kamber, and Jian Pei. 4 - data warehousing and online analytical processing. In Jiawei Han, Micheline Kamber, and Jian Pei, editors, *Data Mining (Third Edition)*, The Morgan Kaufmann Series in Data Management Systems, pages 125–185. Morgan Kaufmann, Boston, third edition edition, 2012.

[23] Geoffrey Hinton and Terrence J Sejnowski. *Unsupervised learning: foundations of neural computation*. MIT press, 1999.

[24] Patrik Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. *Advances in neural information processing systems*, 21, 2008.

[25] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models, 2019.

[26] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115. IEEE, 2007.

[27] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

[28] Yunhui Long, Suxin Lin, Zhuolin Yang, Carl A. Gunter, and Bo Li. Scalable differentially private generative student model via pate, 2019.

[29] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3–es, 2007.

[30] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[31] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models, 2020.

[32] Sergey I. Nikolenko. *Synthetic Data for Deep Learning*. Springer Optimization and Its Applications, 174. Springer, 2021.

[33] Beata Nowok. Utility of synthetic microdata generated using tree-based methods. *Administrative Data Research Centre, University of Edinburgh*, 2015.

[34] Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *Proc. VLDB Endow.*, 11(10):1071–1083, June 2018.

[35] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[36] N. Patki, R. Wedge, and K. Veeramachaneni. The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 399–410, Oct 2016.

[37] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 399–410, 2016.

[38] Haoyue Ping, Julia Stoyanovich, and Bill Howe. DataSynthesizer: Privacy-preserving synthetic datasets. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, pages 1–5. ACM, 2017.

[39] Jules Polonetsky, Omer Tene, and Kelsey Finch. Shades of gray: Seeing the full spectrum of practical data de-intentification. *Santa Clara L. Rev.*, 56:593, 2016.

[40] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2015.

[41] Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. *Advances in Neural Information Processing Systems*, 31, 2018.

[42] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. 1998.

[43] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[44] Henri Theil. On the estimation of relationships involving qualitative variables. *American Journal of Sociology*, 76(1):103–154, 1970.

[45] Amirsina Torfi and Edward A. Fox. CorGAN: Correlation-capturing convolutional generative adversarial networks for generating synthetic healthcare records. In *The Thirty-Third International Flairs Conference*, 2020.

[46] Cinzia Viroli and Geoffrey J McLachlan. Deep gaussian mixture models. *Statistics and Computing*, 29(1):43–51, 2019.

[47] Alexandra Wood, Micah Altman, Aaron Bembenek, Mark Bun, Marco Gaboardi, James Honaker, Kobbi Nissim, David R O'Brien, Thomas Steinke, and Salil Vadhan. Differential privacy: A primer for a non-technical audience. *Vand. J. Ent. & Tech. L.*, 21:209, 2018.

[48] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional GAN. 2019.

[49] Lei Xu and Kalyan Veeramachaneni. Synthesizing tabular data using generative adversarial networks. 2018.

[50] Andrew Yale, Saloni Dash, Ritik Dutta, Isabelle Guyon, Adrien Pavao, and Kristin P. Bennett. Generation and evaluation of privacy preserving synthetic health data. *Neurocomputing*, 416:244–255, 2020.

[51] Jinsung Yoon, Lydia N. Drumright, and Mihaela van der Schaar. Anonymization through data synthesis using generative adversarial networks (ads-gan). *IEEE Journal of Biomedical and Health Informatics*, 24(8):2378–2388, 2020.

[52] Jinsung Yoon, Lydia N Drumright, and Mihaela Van Der Schaar. Anonymization through data synthesis using generative adversarial networks (ads-gan). *IEEE journal of biomedical and health informatics*, 24(8):2378–2388, 2020.

[53] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2019.

Appendices

Appendix A

Additional graphs

In chapter 5, we decided to make subsets of the dataset to reduce the complexity of the dataset. Below is the distribution of the labels in CTGAN and TGAN:

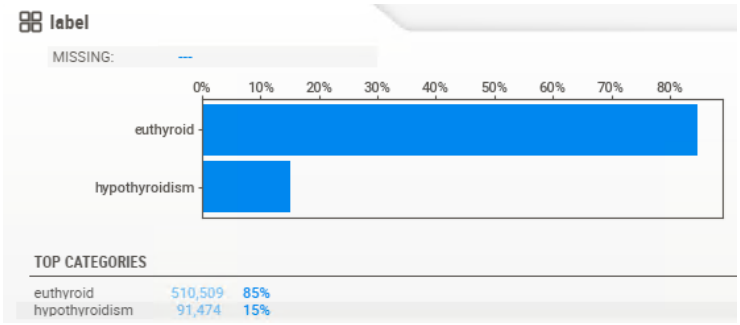


Figure A.1: Distribution of the label column in Hypothyroidism subset

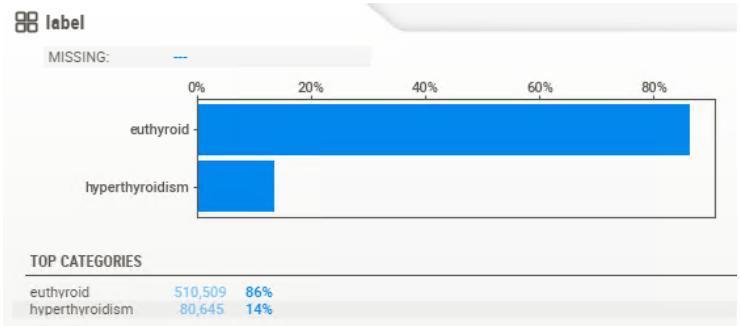
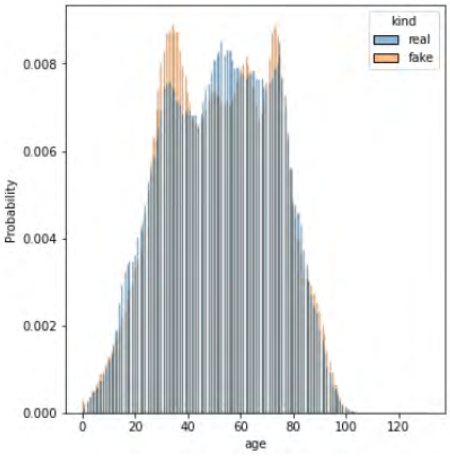


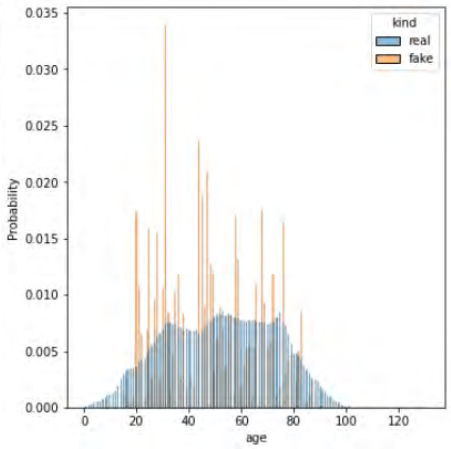
Figure A.2: Distribution of the label column in Hyperthyroidism subset

In section 5.3.1 We discussed about the table evaluator results on the Fürst

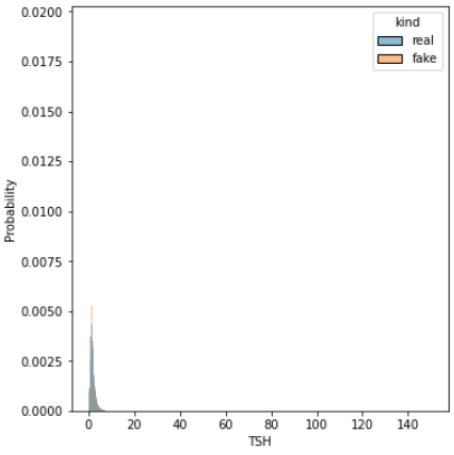
dataset. Below we show the distributions of some of the features in CTGAN and TGAN.



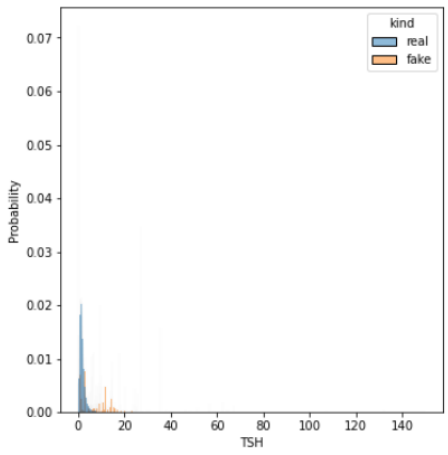
(a) Distribution of age in CTGAN



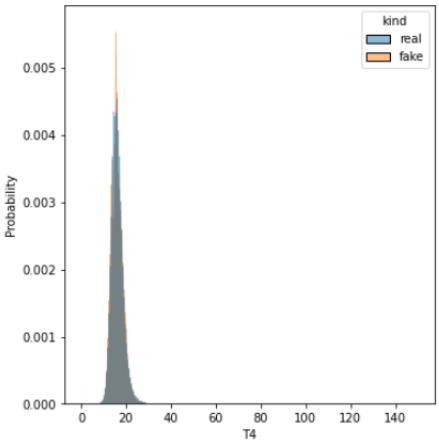
(b) Distribution of age in TGAN



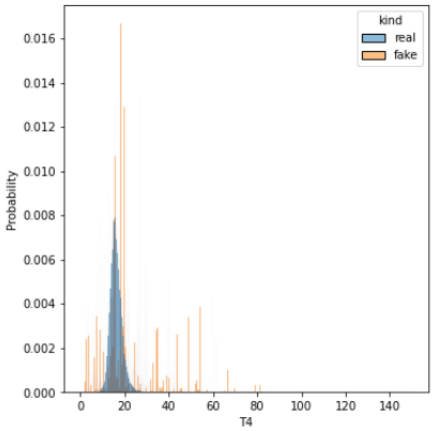
(c) Distribution of TSH in CTGAN



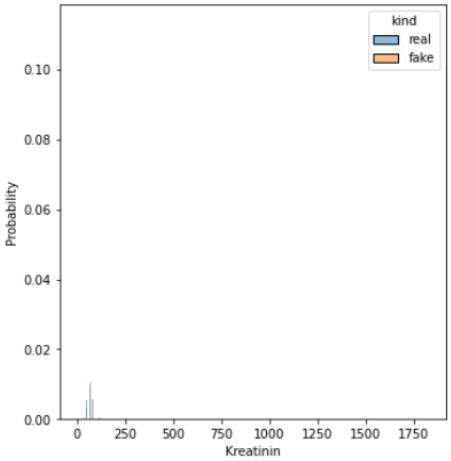
(d) Distribution of TSH in TGAN



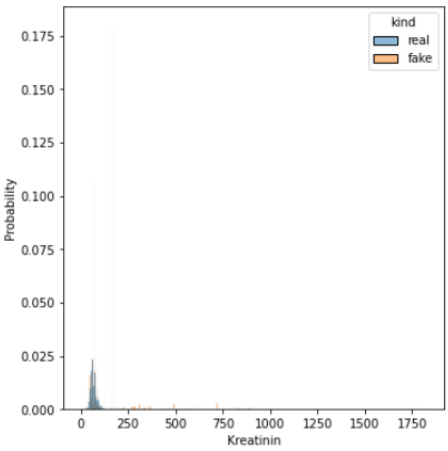
(e) Distribution of T4 in CTGAN



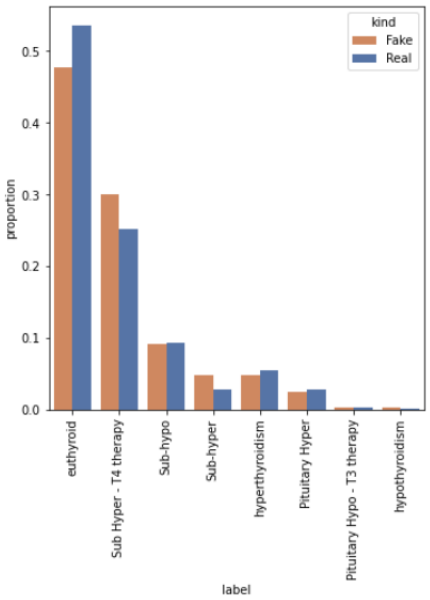
(f) Distribution of T4 in TGAN



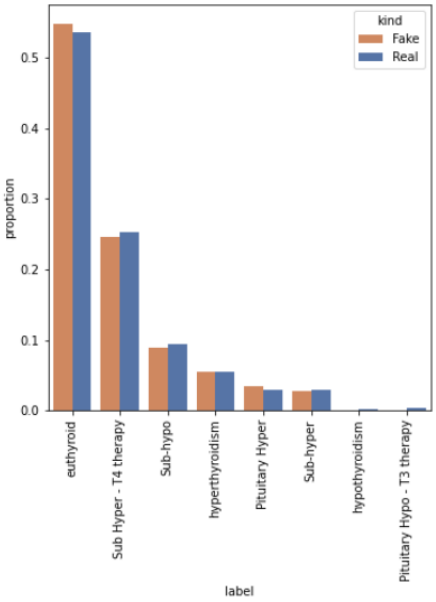
(g) Distribution of Kreatinin in CTGAN



(h) Distribution of Kreatinin in TGAN



(i) Distribution of Label in CTGAN



(j) Distribution of Label in TGAN