Utrecht University

Master Thesis

Improving the Quality of Synthetic Data Generation with Application in Algorithmic Fairness

Artificial Intelligence

Author:
Aleksandra Wypych
(6624154)

Supervisor:
Dr. A.A.A. (Hakim) Qahtan
Second Supervisor:
Prof. Dr. Yannis Velegrakis

Utrecht University, Utrecht, The Netherlands

Utrecht University

July 21, 2022

Abstract

The development of machine learning algorithms has greatly influenced decision-making at various levels. However, these algorithms tend to incorporate biases. Racial profiling in legal and financial systems are the best-known examples of inequality stemming from algorithm decisions. Previous research has shown that one of the reasons for racial bias is imbalanced data. This research will focus on generating synthetic data using Generative Adversarial Networks (GANs) to reduce bias. Inspired by GANs, this paper proposes the Intag framework. This framework contains a modified version of Pate-GAN for synthetic data generation. The main modification from the original Pate-GAN is that the hard privacy constraint is dropped. Other changes, such as changing the architecture of the network, such that a number of hidden layers depends on the dimension of input data. Moreover, the framework will incorporate undersampling techniques to ensure that the synthetic data samples are of the highest quality. The framework's performance is evaluated on the basis of machine learning utility by checking the quality of the synthetic data generated by different methods. It is shown that the modified Pate-GAN achieves the best results. Furthermore, the framework improves the values of statistical parity and disparate impact, the two measures of fairness used in this study. We conclude that our proposed modification to Pate-GAN, and the framework in general, can be used for synthetic data generation. Moreover, it could be used as an aid for data generation to improve fairness in the case of an imbalanced dataset.

Keywords: Synthetic Data, Data Oversampling, Generative Adversarial Networks (GANs), Algorithmic Fairness

Acknowledgements

I would like to thank everyone that helped me and provided me with support in these last couple of months. Especially my supervisor, Hakim Qahtan, for his patience and guidance during this time. I am extremely lucky to have a supervisor that cares about my work and has always found time for me. I would also like to thank my second supervisor Yannis Velegrakis for putting time into reading this thesis.

Moreover, eternal gratitude to my partner Kees Haasnoot, for bearing with me with these last few months, for his endless patience and for his constructive criticism of my work. I would also like to thank my friends and family for always believing in me and supporting me these last months.

Contents

Chapter 1

Introduction

Currently, we collect a huge amount of data that could be used to support decision makers. Such data usage can affect society directly or indirectly. For instance, in big chain grocery stores, where data is collected, the process and underlying structure of most products bought together are used to decide which products should be placed next to each other and which products should go on promotion [1]. Another example is from the logistics of the food sector, where Amazon aims to predict what fresh products should be sold at the store to improve customer satisfaction [8]. However, in many classification problems where a decision needs to be taken between a limited number of options, the collected data might yield misleading outcomes because records from one class appear more frequently than others. This problem is known as the imbalanced data problem. Due to this problem, the classifier will learn more about the patterns of one class than about the others.

Several techniques can ensure the equality of representation of different classes in the dataset, solving the imbalanced data problem. One of the most prominent approaches to solving the problem is based on generating more examples from the under-represented class. There are multiple techniques that can be used to generate synthetic examples that can inherit the characteristics of actual examples in the dataset. The first technique is based on observing and determining the statistical properties of the data and then replicating those properties in a set of generated records [19]. Another technique uses agent-based models. In these, the created model can reproduce random samples that mimic the patterns in the original data [36]. The last technique mentioned here is based on training a deep learning model using the existing data and generating more examples by predicting what the value for each attribute should be [37]. The quality of the generated data would significantly affect the outcomes of machine learning (ML) models that are trained using this data. Hence, it is important to ensure that the generated data records are close to and reflect the patterns in the original data records. In this research, we propose a framework for generating more realistic synthetic data that can be used to improve the performance of ML models.

Besides improving the prediction accuracy of the ML model, synthetic data generation can also be used to mitigate data bias. This would lead to a significant improvement in the ML model's fairness. An example of a widely-used product that has been shown to be biased is COMPAS.[1] This tool helps assess if a recently-convicted person has a high chance of becoming a recidivist. The COMPAS tool is biased against black people, especially black males [24, 51]. With synthetic data generation, synthetic records can be generated to increase the number of black people who receive the favorable class label, so the ML model will learn from the new data and will show less bias against black people.

[1] https://www.equivant.com/practitioners-guide-to-compas-core/

Synthetic data generation can also have applications to privacy, where the decisions that are made might be influenced by sensitive information about individuals. For example, using the medical history information of individuals might create employment discrimination against individuals with a history of cancer treatments [45]. Dwork proposed the concept of Differential Privacy [28]. She argues that systems used or shared publicly should not contain patterns about individuals. In other words, it should not be possible to track an individual within any dataset. Generating synthetic data would help to train ML models that do not consider an individual's background when making decisions. The creation of generated records from the minority class (such as people with extensive medical history) would make the process of identifying individuals harder.

1.1 Motivation

Data is commonly used to train decision-making algorithms. An example of such an algorithm is the aforementioned COMPAS, used in the US judicial system. COMPAS is an aid in choice resolution for judges. It aims to identify whether or not a criminal defendant is likely to become a recidivist. However, the underlying dataset is imbalanced. This imbalance then influences the decision-making process, causing people from the misrepresented groups to be treated unfairly. One way to solve such misrepresentation would be to even out the imbalance in the underlying dataset by producing enough data samples. However, such produced data should still be of high quality, meaning that it must be close enough to the original data records and reflect the patterns of the original data (more realistic records). Producing synthetic data samples of high quality has proven to be a problem, even though many methods have been developed.

In this thesis project, we consider two classes of synthetic data generation techniques. The first technique used in this thesis is SMOTE, which chooses a random record and finds its k random neighbors and generates synthetic examples by linear interpolation of the selected example and one of its k nearest neighbors.

The second approach is to uses deep learning techniques such as Generative Adversarial Networks (GANs) [37]. This approach is based on creating two neural networks that compete in a zero-sum game. The first neural network, called the generator, learns how to generate realistic examples, and the second network, called the discriminator, learns how to classify the generated example as fake or real. This approach is currently one of the most distinguished approaches to synthetic data generation [36]. However, empirical studies show that the accuracy of the generator in generating realistic synthetic data does not exceed 50%. Different network architectures have been proposed to improve the quality of the generated data, such as CTGAN [113]. This is a variation of GANs, training the model on real data. However, when it generates the samples, it is possible to force it to generate samples from chosen classes. This is done by changing the representation of the data being passed onto the network. Then the additional encoded vector is concatenated such that the condition is enforced on the data generation.

Another example is Pate-GAN [50]. In Pate-GAN, the architecture of the network is changed to have three main layers of neural networks: i) A generator network that learns to generate synthetic examples; ii) a set of teachers, which classifies the generated examples as fake or real. Their decisions are then aggregated and provided to the student network (discriminator); iii) a student (discriminator), which takes a final decision if an example is fake or real.

No previous methods have succeeded in generating high-quality synthetic datasets. The implication is that the generated data cannot always be used in the field or as a substitute for the original data. Moreover, previous studies show that imbalanced data is one of the main reasons for discrimination in the decisions of ML models. However, if the synthetic data generation methods are imperfect, it is very hard for machine learning algorithms to discover hidden patterns within the data. The class imbalance could be reduced using GANs to produce more realistic data, leading to fairer ML predictions.

The goal of this thesis is to create a framework that will generate high-quality synthetic data. This has many applications in cases in which datasets are imperfect, especially when the aim is to reduce imbalance.

1.2 Research Questions

The main goal of this thesis is to develop a tool that will be proficient at synthetic data generation. Thus the research will focus on the following:

Research question: How can Pate-GAN improve synthetic data quality by 5% over known methods such as CTGAN or SMOTE?
To answer the main question, the following sub-questions will be answered:

1. What would be the best metrics of the quality of synthetic data?

2. How can Dwork's approach be modified to generate better examples to improve classification accuracy?

3. How can algorithmic fairness be affected by generating synthetic data examples?

1.3 Thesis Structure

This thesis will contain seven chapters. The remaining chapters will be structured as follows: starting with related work, then the theoretical background, following with the introduction of the framework, experiments, results and conclusion.

The related work chapter will provide relevant research on the main problem and current state-of-the-art methods for this problem. Next, it will dive into the theoretical background, which will introduce the problem statement and the methods that will be used in this thesis, namely SMOTE, CTGAN, and Pate-GAN. Following that, the Intag framework will be introduced. This chapter will provide a detailed explanation of the framework's various components. The next two chapters then introduce experiments and their results.

Finally, in the conclusion and future work chapter, the thesis will be summarized. It will also elaborate on the main findings, limitations of the framework, and directions on how to improve the Intag framework further.

Chapter 2

Related Work

This chapter will discuss related work connected to this thesis. It will start by discussing bias and its different types, followed by how bias impacts imbalanced datasets and machine learning algorithms. Then it will discuss solutions to this problem, such as oversampling and undersampling. And lastly, this chapter will discuss fairness, its connection to the imbalanced data problem, what the current state of the field is, as well as what fairness means and how it is evaluated.

2.1 Bias

The spread of machine learning is inevitable. Unfortunately, this tool is not perfect. The main danger identified by the field is that the decisions achieved with the help of algorithms will carry on bias [65]. There are many types of biases, and different kinds surface during distinct stages of algorithm creation.

Mehrabi et al. [65] distinguish three main types of biases in machine learning:

- Data to Algorithm
- Algorithm to User
- User to Data

Data to algorithm is a concept where the bias is already contained in the data passed to the algorithm. Examples of this include how the data is measured and reported in the dataset [98]. Another example is when data omits variables with crucial predictive power [23]. In that case, when the data samples are passed on it is not representative because of a lack of diversity [98].

Algorithm to user biases occur when the algorithm introduces bias, but it is not present in the data itself [86]. This can occur when the findings of algorithms are misinterpreted or the results are presented in a flawed way. This could be the case, for instance, when not all obtained information is given [86]. Another way algorithms shape prejudice is when ranking is included. This occurs, for instance, when displaying the results from search engines [47].

User to Data bias is created when users add to the bias, reflecting user choices. An example of this is historical bias, where the prejudice is already present in the real world, and is transferred into the data used in ML algorithms [98]. Generated datasets tend to be non-representative: for instance, when gathering data about one type of user, it is not possible to draw general conclusions about every type of user [41].

This research will focus on data to algorithm bias, where the data passed to the algorithm is imbalanced. To solve this, it will concentrate on synthetic data generation. By extension, it aims to correct bias in the dataset, not in the algorithms.

2.2 Imbalanced Data

Real-world data is not perfect. Many issues influence the quality of data. There is a direct link between the quality of data and the performance of the predictions based on that data [75]. There might be several reasons why the data is not satisfactory.

The first reason might be the inaccuracy of data. Depending on how data is used, inaccuracy might play a critical role. For instance, in healthcare, the accuracy of the data plays a crucial role. If the datasets are not accurate, this might lead to wrong conclusions [12]. An example is a study on self-reported energy intake to estimate actual energy intake. Because of the external errors and the design of the study, in which humans prone to error had to self-report data, the study gathered invalid data, resulting in poor estimates [27].

This closely connects to another reason why data can be imperfect. Sometimes, data is not shared entirely. There might be different reasons for this, from privacy to intellectual property. Although the validity of the reasons might differ, missing data will have an impact on predictions. Some variable associations that might be valid in sensitive research areas might not be found.

Another problem is data inconsistency. There are many types of inconsistencies, for which this research will not go into detail. However, this has an impact on research and the outcomes that cannot be omitted [118].

One of the main issues in classification carried by data is the imbalance of classes within data. This occurs when one class has much more representation in the dataset. This class is called the majority class. On the other hand, there is a minority class, which has less representation within the dataset. This has an impact on classification algorithms because methods tend to achieve high accuracy for majority classes and very low accuracy for minority classes [112]. Therefore, imbalanced datasets often have high overall accuracy [18]. This general behaviour is shown in Figure 2.1.

There are two main categories of imbalanced datasets. Firstly, the inter-class category refers to a case where one of the classes has many more positive examples than the other. The ratio between the majority and minority classes represents the degree of imbalance [71]. The other type of class imbalance occurs when there is inequality within the class itself. This occurs when the class consists of a few sub-concepts within which there is asymmetry [48]. Because the impact of imbalanced data is tremendous and can be found in many domains, it is essential to eliminate the imbalance. There are four main ways of solving the imbalance of datasets: collecting more data, oversampling, undersampling, and synthetic data generation.

2.2.1 Solutions for Imbalanced Data

There are different solutions for imbalanced datasets. Each of these solutions works on a different level of application.

1. Data level Solutions
   (a) Oversampling
   (b) Undersampling
2. Algorithmic Level Solutions
3. Other methods
   (a) Ensemble learning
   (b) Cost-effective algorithms

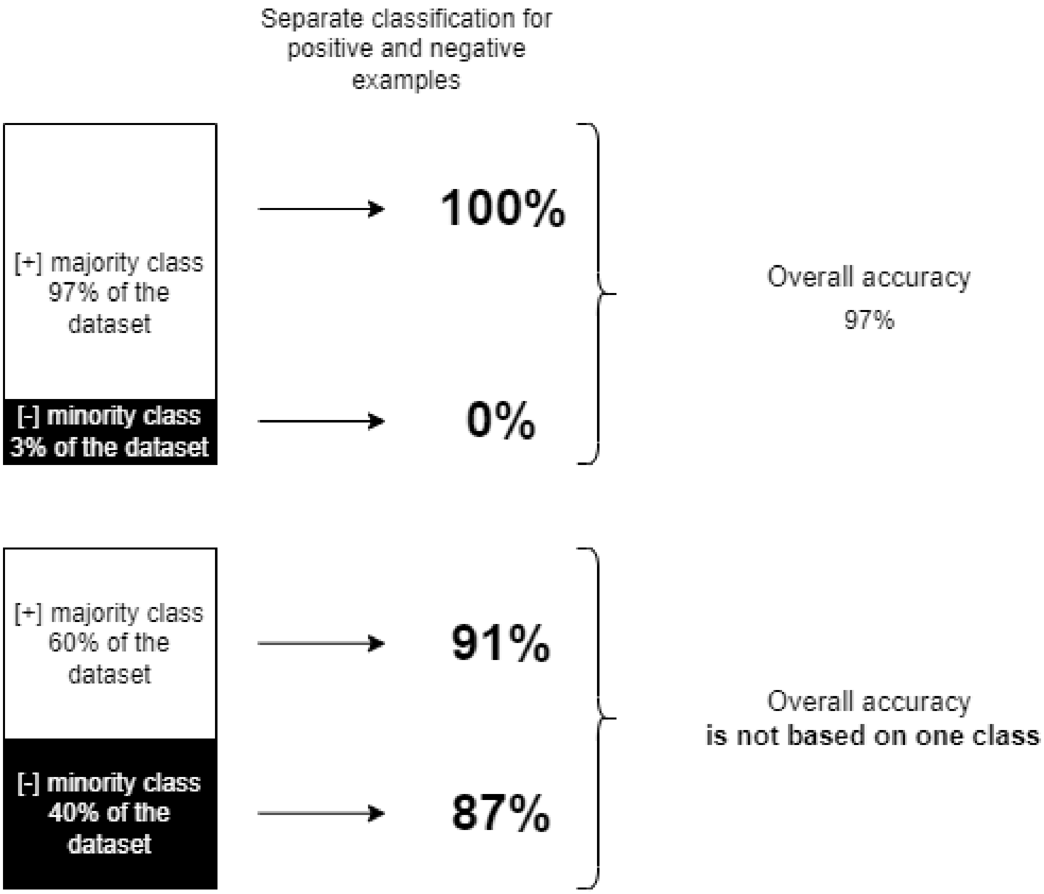
Separate classification for positive and negative examples
[+] majority class 97% of the dataset
[-] minority class 3% of the dataset
100%
0%
Overall accuracy 97%
[+] majority class 60% of the dataset
[-] minority class 40% of the dataset
91%
87%
Overall accuracy is not based on one class

FIGURE 2.1: An example of how an imbalanced dataset affects classification. Suppose there is a significant difference between the classes' distribution. In that case, classification algorithms will mostly consider the majority class, and minority classes will not have a consequential impact on overall accuracy [18]

Each of these categories is divided into more subcategories. They will be discussed in the upcoming sections, which follow Figure 2.2.

2.2.1.1 Data-level solutions

Data-level solutions try to solve the imbalance at the level of the data. They focus mainly on the reparation of the database itself. There are three ways in which data inequalities can be solved. Additional data collection is the first and most obvious way to decrease dataset imbalance. As the name suggests, the idea is to collect more data. There are many ways of doing that. Depending on the type of data one wants to gather, different techniques can be used. The best methods for qualitative or non-numerical data are interviews, focus groups, record keeping, observations, and case studies [46]. Generally, quantitative data is easier to gather and add to existing databases. Unfortunately, all techniques require a lot of time and skill, which may influence the quality of data.

There are many ways to collect quantitative or numerical data. The first of those is by gathering the results of experiments. Another way is to create simulated data. Moreover, it is also possible to collect data from companies, banks, or mobile phones. Although there are many ways to do it, there are disadvantages. First, adding more

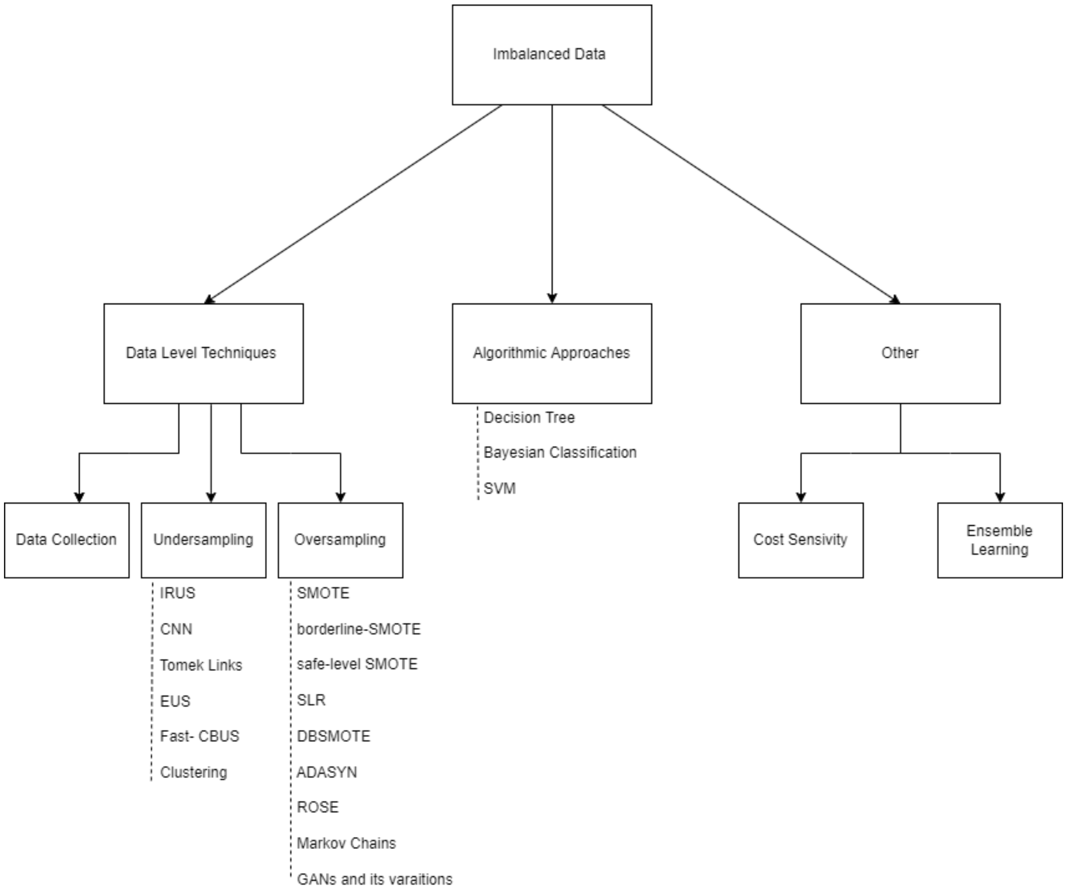

FIGURE 2.2: Different levels of approach for data imbalance

data to an existing dataset is unusual. Data may have changed; the outlets might have changed as well. Moreover, datasets should be consistent, and collecting additional data at a later stage might harm this consistency [34].

2.2.1.2 Oversampling and Synthetic Data Generation

Oversampling is a technique to modify the datasets to prevent data from being imbalanced. It is based on adding extra data examples from existing ones or generating synthetic examples.

Random Oversampling (ROS): this is a method that randomly replicates examples from the minority class [32]. Random samples are chosen, copied, and added to the dataset. This is known as sampling with replacement. Because the replication process is random, there is a possibility that the replicated samples will not cause any information gain. Moreover, the classifier might be more prone to create rules that are only accurate for the replicated samples. Hence, the algorithm may overfit.

Synthetic Data Generation: the second technique generates more examples by interpolating available examples or adding some noise to existing ones. Chwala et al. created the famous algorithm for oversampling called Synthetic Minority Oversampling TEchnique (SMOTE) [19]. The idea is to introduce artificial examples to the dataset. However, this method does not produce random samples. The idea is as follows: a random point from the minority class is chosen, along with its k neighbors. Between these points, at a random distance, a point is created. Finally, n of the k

instances are chosen to be added to the dataset. The main shortcoming of SMOTE is that this tool is impractical for high-dimensional data. It can also introduce more noise to the data. This method will be discussed in more detail in section 3.1.

Because SMOTE had a significant impact on the field, many different methods based on SMOTE were developed.

The first example of a variation of SMOTE is called borderline-SMOTE [39]. This predictor aims to learn the boundaries of the data. In borderline-SMOTE, however, the classifier chooses minority points to replicate near the boundary instead of learning the boundaries. Moreover, it also creates samples from the majority class. By doing this, the algorithm strengthens the boundary between the classes. Unfortunately, this method is susceptible to outliers.

Another extension to SMOTE was invented by Bunkhumpornpat et al., called safe-level-SMOTE. The main change of this extension involves adding a new variable to the dataset [15]. When artificial points are created along the lines between chosen points, the safe-level is calculated for each. The safe-level is defined for each point as a number of k nearest neighbors in the minority class. Next, the ratio (SLR) between the safe-level for each point is determined. The algorithm will decide if the artificial data point should be created based on the SLR. Because of this mechanism, the algorithm will not create noisy data. The main problem with this method is that it creates densely-concentrated samples of the minority class. Another technique based on SMOTE was again proposed by Bunkhumpornpat et al. [14]. Instead of creating a more safe-levels approach, they build upon a borderline-SMOTE tool called DBSMOTE. This algorithm clusters the minority and then generates random samples between the centroids of the clusters. The main disadvantage of borderline-SMOTE is that it does not operate well with outliers. Because DBSMOTE works on a cluster of the data, not single points, it overcomes this weakness. Unfortunately, as with all of the SMOTE-based methods, it tends to create noisy data.

SMOTE was one of the first techniques that proved to have a positive effect on the imbalanced data problem, revolutionizing the field. It was extended many times and other methods based on SMOTE were introduced. But this was not the only approach. There are other methods of synthetic data generation.

An example of such an algorithm is ADASYN - Adaptive Synthetic Sampling Approach [44]. First, this algorithm calculates the degree of imbalance, represented as a ratio between the majority and the minority class. This is used to determine how many samples should be generated to correct the class imbalance. For each of the minority points, the ratio between the minority and majority neighboring points is determined. For each of the minority points that dominate the major class examples, classification is harder to obtain near this point. Therefore those are harder to learn. For each neighbor, the number of samples that need to be added to the data is determined. The neighborhoods that are harder to learn will obtain more points to rebalance the dataset. Artificial points are created between the minority points within each of the neighborhoods.

This simple tool is very powerful. First, it generates the points in parts of the data where the classifier will have problems with learning. It also automatically decides on the number of points that needs to be generated. However, because it generates the data in the proximity of high amounts of majority data, it might happen that the generated data will be similar to the majority data. Hence, it will generate a lot of many false positive examples.

Menardi et al. introduced another approach for synthetic data generation called Random OverSampling Examples (ROSE) [66]. This method uses a smoothed bootstrap approach, which adds noise to each data point selected for replication [103].

It handles continuous and categorical data by generating synthetic examples from a conditional density estimate of the two classes.

Another approach uses Bayesian networks, where synthetic data is generated with inherited dependencies between the variables [36]. Li et al. propose a method for tabular data generation that is based on a distribution of the data and dependencies between variables with the Gaussian copula [60]. This way, the synthetic data mimics the properties of the actual data.

The approach by Park et al. uses a Markov chain with Monte Carlo based on obtaining approximate values from a specific distribution [78]. This methodology also uses the original dataset's distribution and other statistical properties to sample the synthetic data.

The most prominent machine learning method used for synthetic data generation is Generative Adversarial Networks (GANs) [36]. GANs were introduced by Goodfellow [37]. This architecture relies on two competing networks, a generator G and a discriminator D. Both networks compete in a zero-sum game between each other. As such, if one gains, the other loses.

More details on how GANs work can be found in section 3.2. Choi et al. proposed using GANs to generate data on Electronic Heart Records [21]. This was done by using GANs with an auto-encoder of the data. As an advantage, they showed that the model could reproduce realistic results. The main downside was that because the data was so specific and used auto-encoding, it is only possible to apply this method to Electronic Heart Records.

The widely-known variation on GANs was developed by Arjovsky et al.[4]. This architecture is called Wasserstein GANs. This extension of original GANs tries to approximate the underlying data distribution by using a novel way of generator model training. The discriminator is replaced by a critic that scores the sample to be authentic or fake. They show that Wasserstein GAN can potentially solve some issues with GANs. This approach is extended with synthetic data generation with privacy constraints [83]. In this thesis, the model's training is split into two phases. The first phase generates the synthetic data, and the second phase extends the generation by adding privacy and fairness constraints. The main problem with Wasserstein GANs is that it is harder to balance the generator and discriminator. Moreover, as with all GANs, it is possible that it will not converge [56].

Another variation was proposed by Xie et al., who added privacy constraints to the discriminator training [110]. Moreover, they enforce Wasserstein's loss by applying Lipschitz's constraint on the discriminator. Because privacy constraints are added, it is harder to optimize the hyperparameters in training [38].

Because of that, Cheng et al. proposed that the privacy constraints should be enforced with the generator instead of the discriminator [20]. This will ensure that the hyperparameter optimization will not be as complex as in the case of the research of Xie et al. [110].

Furthermore, Xu et al. proposed another solution to improve GANs. They added three novel additions to the model [113]. They allowed the generator to learn the underlying distributions of the minority class better and, by extension, produce more realistic samples. Moreover, they also introduced special normalization techniques for the data that make learning complex numerical distributions easier. Unfortunately, this model is not perfect and it cannot handle missing values. This methodology is examined more closely in section 3.2.1.

Pate-GAN is another variation of the GANs network [76]. The main difference is that Pate-GAN used the Pate framework for data generation. The Pate framework is Private Aggregation of Teacher Ensembles, creating a teacher-student role while learning. The idea begins that teacher models are used to train sensitive data, which does not have to be public. Therefore it allows a student to derive knowledge from the teacher model. The main disadvantage is that the student discriminator is only trained on the fake data, which might impact the accuracy. Pate-GANs will be used in this research, and more information on the methodology is in section 3.2.2.

2.2.1.3 Undersampling

Undersampling of the data is preserving all the minority classes and reducing the number of major instances [71]. Random Undersampling is the first and most straightforward method, which randomly removes entries from the dataset [69]. Because of its randomness, this approach tends to lead to a vast loss of information and data patterns that could have been deduced [43]. In order to improve the random sampling method, Tahir et al. constructed a method called Inversed Random Under Sampling (IRUS) [101]. This method makes it possible to randomly choose the deleted sample, but this method should not balance the majority and minority classes [100]. The primary dataset is split into r ways in such a way that the majority class becomes a minority. For each of the splits, a base classifier is trained. The combination of the base classifiers' outcomes allows for the construction of a boundary between the two classes.

In order to overcome the shortcomings of eliminating random entries, researchers found a way to eliminate neighboring entries instead. An algorithm called Condensed Nearest Neighbor (CNN) was proposed by Hart et al. [42]. CNN is an algorithm that takes into account the underlying data and its properties. The algorithm constructs a subset of the original data so that there is no information loss during the removal of samples. It creates a minimum consistent set. This is achieved by adding all the samples to the subset if they cannot be classified by the samples already in the set itself. Therefore, the algorithm will add all minority class samples while successfully removing the majority class samples. Therefore, CNN can remove instances from the dataset that do not add to the borderline cases, possibly adding noise to the dataset.

As a result, Ivan Tomek introduced a modification to the CNN [104]. This modification is called Tomek links. Instead of dealing with the nearest neighbors, it looks at a pair of instances from the majority and minority classes that are very close together. The algorithms then remove the majority of instances of such a pair. This ensures that the boundary between instances will be preserved and emphasized in most cases. Removal of the majority of instances can shift the decision boundary in the wrong direction.

Devi et al. tried to eliminate the shortcoming of Tomek-links by removing the noisy data, outliers, and the Tomek-links such that the removal of an instance only has a minor impact on the overall prediction power [26].

Another method for undersampling is called EUS, Evolutionary Prototype Selection. This method was developed by Garcia et al. [35]. Here, a genetic algorithm approach is used to determine which samples should be preserved in the dataset.

Another approach to undersampling is to extensively use clustering techniques to correctly decide upon entry removal. Yen and Lee propose a solution where undersampling is based on clustering [114]. The main idea is to create several data clusters with distinctive characteristics. For each cluster, the number of individuals from the minority class and the majority class is then determined. If the cluster consists of majority class examples, it will behave as such. Therefore, it will be possible to obtain the sample that clusters the data and is the most representative. As shown by

Rahman et al., this approach can be helpful in medical research [82], where the clustering is beneficial to overall prediction accuracy. Another extension of the clustering method is Fast-CBUS, proposed by Ofek et al. [74]. This method divides the data into K clusters. Each cluster contains data with an equal number of majority and minority classes. Because of that, researchers want to create an algorithm that does not discriminate and decreases the classification time. A classification algorithm is trained for each of the clusters. If the data instance cannot be assigned to any cluster, it is considered a majority class. Otherwise, it is classified accordingly to the cluster classification. Because of that unique idea, this approach is not prone to information loss. On the other hand, splitting the initial data into a defined number of clusters might be challenging.

Many other methods for data undersampling were invented, especially as a combination of clustering with already-known algorithms such as CUSBoost (clustering + AdaBoost)[85] or Fuzzy outlier clustering (Fuzzy C-Means + clustering [107],[108].

Because undersampling is based on discarding data, it is prone to information loss. This method of solving inequality is preferred with large datasets, so the lost information data is irrelevant. Moreover, this method is preferred when the data has a slight imbalance, so that the information loss is relatively small [26].

2.2.1.4 Algorithmic approach

An algorithmic approach to imbalanced data assumes that the imbalance will be solved within the classification. The idea is to choose an appropriate inductive bias.

Decision trees are known classifiers that will also be used later in this study. A few techniques will make it possible to use a decision tree on imbalanced data. The first is to make sure that while creating the leafs for decision making is to adjust the leaf's probabilistic estimate [81]. This can be done, for instance, by using a method of pattern recognition such as Predictive Association Rules [115], where the greedy algorithm is adapted to create association rules from the training data. Moreover, another approach is to adjust pruning in the decision trees. Zadrozny et al. argued that with imbalanced data, it is best to leave unpruned trees [117]. Instead of transforming obtained scores to class membership probability, the general argument for that was supported by Bradford et al. [11].

Another approach uses Bayesian classification, which infers data structure and properties from the underlying graph structure. Such a graph consists of nodes that represent the variables and edges that represent conditional dependencies between the nodes [80]. The idea is to find the network that displays and matches the internal data structure. Learning the most common patterns will ensure that the model does not overfit the data. As the main problem is that the patterns inferred from the minority class might not have a significant impact on the overall model; hence there is a possibility of being misclassified [57]. Different variations of the Bayesian network have been introduced. Klein et al. develop a network where data weights have been introduced that favour the minority class [54]. Moreover, different variations of the Bayesian network can be found in natural language processing [72].

One of the most-used algorithms with imbalanced data is Support Vector Machines (SVM). The general idea of SVM is to separate the data in higher-dimensional space, creating a rigid boundary between the samples. It has been proved that SVM is not affected by class imbalance more than other algorithms [49]. A main limitation of SVM is that it can contribute towards the bias of the minority class because it provides a hard boundary between the data. It also does not perform well with skewed datasets [3].

Liu et al. enhance SVM through novel use of the sampling method to be used in classification [62]. The authors used Ant Colony Optimization, where the algorithm eliminates noisy data and chooses the data that displays common behavior in the dataset. As a primary downside, the authors mention that this method is very expensive, both computationally and in terms of storage. SVM proved to be an excellent tool for imbalanced data prediction, but researchers started to combine SVM with different algorithms to obtain better results because of its shortcoming. An example is to combine it with the k-nearest neighbor (k-NN) algorithm. Majid et al. showed that combining two classifiers on balanced data works and has a high prediction accuracy [64]. Authors oversample the data using a method called MTD. It calculates statistical properties and uses it to create artificial samples [59] Beyan et al. propose a method for hierarchical decomposition of the data [9]. This method makes it possible to cluster the data based on the pairwise distance between variables. When the data is clustered, an algorithm looks for outliers within the clusters and segregates them into minority and majority clusters. This is applied to each level of feature selection. Therefore this method can correctly identify the correct classes. Unfortunately, the complexity of this method is very high. However, this is not the only approach of clustering that can be found for imbalanced data; many clustering methods have been used in undersampling.

2.2.1.5 Other Methods

Besides the methods working on data and algorithmic level, there are two other trends with imbalanced data. The first of them is cost-sensitive learning. In it, the model considers the mistakes of classification while training. An example of that is where a cost-sensitive objective is added to the predictor model [17]. Moreover, it is shown that cost-sensitive methods often tend to perform very well concerning precision and recall. Mienye and Sun showed that cost-sensitive learning with medical data outperforms oversampling and undersampling methods [68].

The second is ensembles learning, where multiple learning algorithms are combined to obtain the best results. This step often means combining the oversampling or undersampling methods with appropriate classification algorithms. An example of that is a combination of the cost-sensitive approach introduced by Fan et al. [30]. AdaCost is a merge of the AdaBoost algorithm, where the main change is that for each weight update, the penalty for misclassification is more significant than in the original AdaBoost.

2.3 Evaluation of synthetic data quality

For the evaluation of synthetic data, there are three different methods, which are:

1. Machine Learning Utility
2. Statistical Similarity
3. Privacy Preservability

2.3.1 Machine Learning Utility

Machine Learning Utility (MLU) is a method where various classifiers and the accuracy are compared [113]. This is done by training the classifier after and before the synthetic data is added. The classifiers should be optimized. MLU is achieved with

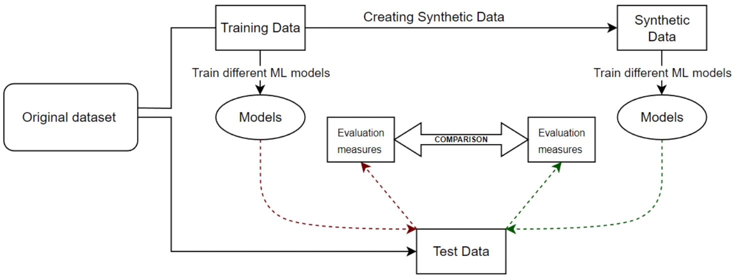

FIGURE 2.3: Steps of Machine Learning Utility measures for synthetic data generation as an evaluation step

the following steps in Figure 2.3. This is used to determine if the synthetic data can be used as a good proxy of the original data.

For synthetic data evaluation, this research will use different techniques. First, this thesis will use F1 score as it is said to be one of the best metrics to evaluate imbalanced data [116]. Moreover, it closely connects to the fairness measures,as it uses values from the confusion matrix, that is also used in various fairness definitions, more of that is described in 2.4.

If the data is imbalanced, the accuracy of the classifier as a whole will not reflect the accuracy for the minority class, as shown in Figure 2.1.

In order to make sure that the minority class is taken into account, other metrics will be used. An example of such a metric is Receiver Operating Characteristic (ROC) [99]. This metric is a graphical curve representing the trade-off between the True Positive Rate and the False Positive Rate. In the ideal scenario, the ROC curve would have a value of 1 for the True Positive Rate and 0 for the False Positive Rate. Because ROC is a curve plotted against the X and Y axis, where X is a Rate of False Positive and Y is a Rate of True Positive, the bigger the Area Under the Curve (AUC), the better the classification is. This represents the trade-off and shows that the True and False Positive rates change together.

In order to compare different classifiers, the AUC can be used. The AUC can be interpreted as the probability that the model ranks a random variable that is positive higher than a random instance that belongs to the negative class [32]. The value for such a score is always between 0 to 1. Moreover, AUC is not biased toward the model's performance on the majority or minority class, which makes this measure more appropriate when dealing with imbalanced data [116].

It is possible to obtain precision and recall from the confusion matrix, as shown in equation 2.3.1.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

Because of how it is defined, precision concentrates on measuring the probability of accurate classification of positive instances. Precision is not affected by imbalanced data because it relies on the number of true positive and false positive samples [13]. Moreover, it is possible to obtain recall, defined as:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

In the classification recall indicates the fraction of the data that has been classified successfully.

Furthermore, by combining the two, we can obtain the PR-curve, where the Y-axis is Precision and X is Recall. It has been said that the PR-curve is more informative for imbalanced datasets than the ROC-curve [88]. Because the PR-curve is designed to detect rare events, it will show how well the classifier can perform on the imbalanced data. It will be able to detect the performance on the minority class [13].

Another metric that is often used with imbalanced data is F-score, as below:

$$F - score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

The equation shows that it is the harmonic mean of precision and recall. This will evaluate the model and balance between precision and recall. It is less likely to be biased toward the majority or minority class [13].

2.3.2 Statistical Similarity

Another metric is Statistical Similarity. This metric determines if the data generated can be used as a proxy for the original data. It checks if the original data and the synthetic data have similar statistical properties [36].

One of the first metrics used for statistical similarity is Kullback-Leibler (KL) divergence, which is computed on the pair of real and synthetic marginal probability mass functions (PMF) [36]. It compares two variables and shows how much information is lost when the two distributions are analyzed. If KL divergences are 0, the distributions are the same. The increase in the KL value implies that the difference between the distributions is more significant. This concept comes from information theory. It quantifies how much information was lost.

Jensen-Shannon (JS) divergence is an extension of KL [61]. This metric also measures the PMF between the variables. This method provides a smoothed and normalized version of KL divergence. The values always vary between 0 to 1, where 1 means that the variables are could not be more different. Therefore, the results of JD divergence are easier to interpret.

Another metric that evaluates the synthetic data is Wasserstein distance [84]. In a similar manner as KD and JS divergence, it measure the difference between the distributions.

Pairwise Correlation Difference (PCD) is a measure that shows how much correlation there is between the variables [36]. It takes the absolute value of the difference in correlation of two variables. PCD measures the difference in the Frobennius norm of the Pearson correlation matrices computed from real and synthetic datasets.

2.3.3 Privacy

The last measure concerns privacy and how well the privacy between the original and synthetic datasets is preserved. Unfortunately, this metric is not as widely used. Researchers tend to either focus on the privacy aspect, such as Park et al. [77], or completely omit the topic like Xu et al. [113].

One of the metrics that measures this is Distance to Closest Record (DCR) [77]. This measure the Euclidean distance between an instance of the original dataset and the closest instance of synthetic data. The higher value of DCR, the better privacy is contained, and there is less risk of privacy preservation.

Another metric that can determine privacy preservation is called Nearest Neighbor Distance Ratio (NNDR) [63]. This method takes the two closest neighbors of a synthetic instance and measures the ratio of Euclidean distance between them. The value of this ratio is always between 0 and 1. The closer the ratio is to 1, the better privacy is preserved.

2.4 Fairness

Very often, outcomes of algorithms are discussed in terms of fairness. Unfortunately, there is no one definition of fairness widely accepted by the community. Such definitions are usually heavily influenced by the culture one is surrounded. Hence, it is challenging to come to one definition [92]. One commonly used definition is that fairness can be perceived as a lack of any prejudice with similar and simultaneous treatment of similar individuals or different groups [28].

Because fairness is a complex topic, different work uses different approaches and paradigms. First of all, we can distinguish Fairness Through Unawareness versus Fairness Through Awareness; the second one is the "What You See Is What You Get" (WYSIWYG) vs "We Are All Equal" (WAE).

Fairness Through Unawareness is a simple idea, where we do not include all the variables in the prediction process [33]. The unawareness is achieved by omitting the variable from the protected classes; hence those will not be used in the final product. This ensures that any method will not rely on the sensitive attributes. If one would compare two similar individuals, different protected attributes would not influence the prediction outcome [105]. There are downsides to this approach. First of all, other variables used in the algorithm's training, so-called -*quasi-identifiers* can be used to identify the sensitive attributes. An example of this is where simple demographic data can be used to identify a person [47]. Another disadvantage of this approach is that the excluded variables can often be helpful in research. An example of this is race. Researchers claim that the multidimensionality of the concept of race can be helpful in sociological or medical research [40].

On the other hand, Fairness Through Awareness does not exclude protected variables. It calls for understating that data and the sensitive attributes. The idea of fairness through awareness is simple - mapping of the outcomes should map similar people. For example, Dwork et al. use Lipschitz continuity mapping, where the distance between two individuals is mapped over the outcomes in the space of probability distribution. Lipschitz continuity is used as a hard constraint, where the distance between probability distributions cannot be greater than the output distance. This approach prevents explicit discrimination, reverse redlining, self-fulfilling prophecies, and reverse tokenism [29].

On the other hand, there is WYSIWYG and WAE. WYSIWYG states that the real world and obtained datasets are the same within a small threshold error. Therefore, the collected data represents real-life statistical properties and features. The implication is that all the differences present in the datasets are also present in the real world. WAE states that there are no major differences between the distinctive groups. If bias appears in the dataset, this suggests an inherent problem with the dataset and data collection. Therefore, it does not represent the real world.

Fairness measures can be applied at different stages in the development of predictive algorithms. First, it can be applied in the pre-processing stage, where its primary focus is the dataset fed to the algorithm. It can also be applied to the post-proceeding stage, where they are used on the outcomes of the predictive tool. The simplest measures of fairness are called the *statistical measures of fairness*. This type of metric uses a confusion matrix, shown below:

		Prediction outcome	
		positive	**negative**
Actual value	**positive**	True Positive *TP*	False Negative FN
	negative	False Positive FP	True Negative TN

From this confusion matrix, one can construct many measures used in the literature, for instance positive predictive value (PPV). This calculates all properly-classified corrected cases over all positively-classified cases. Very often, PPV is regarded as precision and shows the probability of an individual belonging to a positive class [105]. There are many more measures that rely on the confusion matrix, such as false discovery rate (FDR), negative predictive value (NPV), true positive rate (TPR), and false positive rate (FPR). Very often, more advanced fairness measures are based on these definitions.

There are two ways we can categorize the statistical definitions:

1. Based on the predicted outcome.

2. Based on the predicted and actual outcome.

The most popular measure that is based on the predicted outcome is *statistical parity* [29]. This is also called group fairness. The groups are an unprivileged group (Upr), which are the class that is discriminated against, and a privileged group (Pr), which is a primary group with positive classification outcomes [87]. These metrics assess the probability of being assigned to the positive group from both privileged and unprivileged groups [119], [95],[29],[33]. This measure on its own does not detect any discrimination. For instance, it does not say how or why the privileged groups have been classified as positive.

Statistical parity definition can be extended in such a way that it is permitted to use a set of attributes to calculate the value. This is called *conditional statistical parity* [25]. This checks if attributes have any power over the algorithm's outcome.

On the other hand, we have metrics that are based not only on the prediction but also on their actual outcome. The first discussed measure of this type is predictive parity, also known as the outcome test. This definition uses PPV and states that both groups - privileged and unprivileged - have the same value of PPV [105]. Therefore, both groups should have an equal probability of truly being classified positively.

Another metric of fairness is called *disparate impact*. It compares the proportion of people who get a positive result between two groups: the non-privileged group and the privileged group. It is calculated as the proportion of positive results received by the unprivileged group divided by the proportion of positive results received by the privileged group [105].

Unfortunately, it is not possible to satisfy all the statistical measures at once [55], [91]. Because of that limitation, researchers use more elaborate techniques to introduce fairness into algorithms. One of their focuses with this is Individual Fairness.

Individual Fairness measures mainly focus on the outcomes for each individual. Similar individuals should be treated similarly [28]. In order to satisfy this constraint, the notion of *fairness through unawareness* was proposed. Another fairness measure for an individual was proposed by Joseph et al., where they used fairness in the context of a multi-armed bandit problem. The proposed metric assumes that an individual should always be classified positive if the conditions are met, regardless of the sensitive attributes. Authors ensure this outcome by a regret bound imposed on an algorithm. Therefore, it cannot favor any individual over another. Unfortunately, this approach assumes that whatever strong relations are implemented in algorithmic computations will also be held in real-life settings.

Another individual fairness measure is introduced by Lahoti et al., by the name of iFair [58]. This approach focuses on the fair representation of data. The authors treat fairness as a property of the dataset. Therefore, it is possible to achieve some level of fairness by pre-processing. The data is transformed so that the cost function minimizes fairness and information data loss. In one of the examples of transformation models, there is a trade-off between the degree of transformation and the effect that it has on the predictor's performance [31].

A final branch of fairness focuses on causal reasoning. Causal reasoning is a process of identification of underlying relationships between variables. By applying this causal reasoning, one can find underlying connections in the data [79]. Such relations can be derived from a proxy attribute, from which it is possible to assume one of the sensitive attributes. One of the fairness measures that deals with causality is counterfactual fairness. This method tests if there are any proxy attributes. If the decision depends on them, counterfactual fairness is not satisfied. This definition can be extended to no proxy discrimination, a framework proposed by Kilbertus et al.[53]. They developed a procedure to remove proxy discrimination if the causal graph constructed on the data has no paths from the protected variables to proxy variables.

The main shortcoming of the causal fairness framework is that it is tough to construct such a graph, as well as to identify the correct sensitive, protected, and proxy variables.

2.4.1 Fairness Frameworks

One of the reasons bias and lack of fairness are created is the lack of a balanced dataset. Therefore, by creating more artificial data, one will solve the lack of representation of minority classes.

Many tools try to achieve that. One of the most prominent ones was created by IBM, and it is called AIF360 [7]. It was created to unify the approaches to fairness and provide a standard network for industrial use. The tool aims to detect any biases and unfairness that datasets might include. It includes extensive metrics that can be used to test for biases as well as adding an explanation of what it implies.

Furthermore, it provides solutions and different bias mitigation algorithms that can be applied to data to increase fairness. It can be applied to pre-processing, in-processing and post-processing. The main limitation is that it is currently only limited to a web interface with limited datasets and limited classification tools.

Another tool was developed for Microsoft by Bird et al. and is called FairLearn [10]. This toolkit was created to assess and help mitigate the bias within AI techniques. It provides interactive visualization and mitigation algorithms. Unfortunately, this tool has a very limited number of algorithms applied to the dataset.

FairVis was developed by Cabrera et al. [16]. This tool is a different example of a framework developed to help with bias and unfairness. It also tries to unify the fairness definition. It proposes the visualization of data with its biases. It focuses on fairness and subgroup fairness. It allows its users to visualize the differences and drill into details of differences between the distinct subgroups. Such differences can be discovered thanks to a novel approach that clusters the subgroups and applies fairness measures within the clusters. The main disadvantage is that this tool does not include bias mitigation techniques.

The framework proposed by Bantilan called Themis-ml is another approach for Fairness and its applications [5]. This framework was developed to unify different approaches to fairness and bring attention to the misconception that algorithms are categorically objective. This tool contains a few fairness metrics, as well as different solutions for bias. A simple classification ML pipeline consists of five steps: data ingestion, data pre-processing, model training, model evaluation, and prediction generation on new examples. Moreover, it is possible to use prepared datasets to explore the tool. Uniquely, the author also provides a discussion and solutions for the trade-off between the fairness and the accuracy of classifiers. Unfortunately, this tool is still limited with respect to the definitions of fairness, its purposes, and the solutions that can be obtained because of it.

The last tool discussed will be DiscriLens, developed by Wang et al. [106]. This is an interactive visualization tool that makes it possible to display the data differently. In order to show detailed data facts, it uses causal modelling with classification mining rules to identify potential variables that cause discrimination. Moreover, it explains discrimination in the dataset. Unfortunately, this tool does not provide any solution for unfairness but only focuses on detecting it.

There are more tools available currently, such as TensorFlow, Fairness Indicator [111], Aequitas [89], What-If [109], FairSight [2]. However, those tools are not as widely known, and they present only a fraction of usefulness because they do not offer any solutions for unfair data, just like DiscriLens provides the explanation and bias detection.

Chapter 3

Theoretical background

Problem statement

Synthetic data generation is a very hard task. Many different attempts have been made to create data of sufficient quality. A lot of datasets do not include enough samples to create models that will lead to good results. Datasets may also lack balance between samples. Finally, datasets might include information that is sensitive and should not be used directly in classification problems. Because of the aforementioned problems, it is not always possible to use the original dataset. One of the solutions to these problems might be synthetic data. Unfortunately, this is a very complex problem, for which no perfect solution has been found. During this research, the following problems will be tackled:

- Increasing synthetic data quality.
- Quantification of data quality.
- Identification of the problems with synthetic data techniques.
- Development of a framework that will be able to avoid the limitations of other synthetic data generation models.
- Empirical analysis of the proposed framework with different methods while assessing the quality of the produced synthetic data.
- Framework evaluation based on the application to fairness.

This research will focus on achieving these goals. In order to do so, it will create a framework with different methods. This thesis will use oversampling as well as undersampling methods. For synthetic data generation, this research will use SMOTE, CTGAN, Pate-GAN, and a modified version of Pate-GAN that was developed specifically for this thesis. For undersampling, this thesis used Condensed Nearest Neighbor. This will be applied to the synthetic data generated by the other methods. This chapter will describe these methods.

Sampling methods

3.1 SMOTE

In order to generate synthetic data, many approaches can be used. Chawala et al. [19] created a tool that used oversampling of the data to create artificially-generated information, called Synthetic Minority Over-sampling Technique (SMOTE). The algorithm chooses a point from the minority class. From that point, n nearest neighbors are chosen. Between the first point and the n neighboring points, straight lines are created. Along these lines, synthetic data points are generated. This process is visualized in Figure 3.1. In this figure, the first minority point is coloured in black and created synthetic data is shown in red. Depending upon the amount of oversampling required, neighbors from the k nearest neighbors are randomly chosen. If 200% oversampling is needed to generate sufficient data, 2 out of 5 randomly generated are further selected to be added to the data. Moreover, these 2 samples out of 5 are chosen randomly.

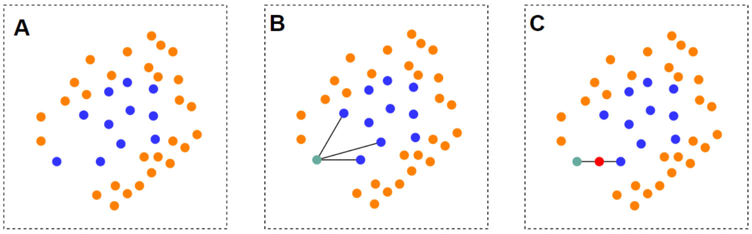

FIGURE 3.1: An example of how SMOTE operates. The first random point with its n nearest neighbors is chosen. Then, the algorithm creates artificial data points along the lines between the points, as shown by the red dot [93].

SMOTE operates by creating data points based on the distance between the variables, or group of variables. Depending on the variable type, it can be done easily (as, for example, a difference between ages). The algorithm measures the distance between the variables as the Euclidean distance between the two variables x,y, as follows:

$$d(x, y) = \sqrt{\sum_{i}^{z} = (x_i = y_i)^2} \tag{3.1}$$

Where z is the number of features for each point, and i is the current index variable. The distance is obtained by obtaining the difference, then squaring it and summing for the z variables. Because the sample is created along the line of the Euclidean distance, this causes the selection to be aligned with the line segment between the chosen points. The new data point will be created for each chosen neighbor. Such data points will lie on the line between the points with random proximity. Therefore, for two points x and y, between which a new point will be created, this applies:

$$NewPoint(x, y) = x + r \cdot |(x - y)|, where : r \leq 0 \leq 1 \tag{3.2}$$

The value of r is chosen with a random probability. Therefore, it is responsible for the proximity of the newly-generated data to the original point. Because of how SMOTE operates, it is a very powerful synthetic data generator. The most significant advantage of this method is that the algorithm does not duplicate the entries. It creates new data points similar to the one in the original dataset. On the other hand, such a method is prone to oversampling the uninformative or noisy data [97]. Moreover, it is hard to determine the perfect n for n nearest neighbors, as well as which data points should be chosen in the first place.

3.2 Generative Neural Networks

Artificial Neural Networks (NN) are computing systems made to imitate neural networks in human brains. They are a collection of units, called nodes, connected by edges. Each edge carries a signal which is a real number. Outputs of each neuron are computed by the sum of inputs, but each edge has a weight that helps the algorithm to learn [73]. Different architectures of NN are good at different tasks. A novel architecture of NN was introduced by Goodfellow et al., called Generative Adversarial Networks (GANs) [37]. GANs consist of two networks. One is called the Generator, G, and the other is called the Discriminator D. The goal of the Generator is to generate samples of data. In contrast, the goal of the Discriminator is to distinguish between fake and real samples.

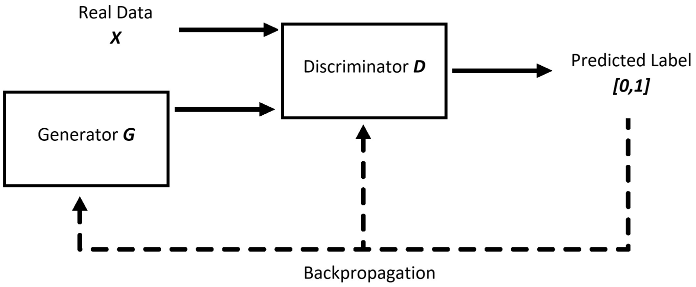

FIGURE 3.2: Simple scheme of GANs. Real data, as well as data from the Generator, is fed to the Discriminator. It then decides if the data is real or not, and based on the feedback network G learns.

Generator

The Generator aims to learn the probability P of all the variables in the sample space Ω. This is achieved by using feedback from the Discriminator network. The objective is to create samples from the given data and learn the function of mapping variables to the sample space. This can be represented by G: A → Ω, in that A, is a vector representation of the data. Moreover, because the Generator learns from the feedback of the Discriminator, it is possible to determine non-linear mapping to the sample space. Therefore, the objective is to maximize the value generated by the samples with the highest possibility of being classified as real.

One of the best-known pitfalls of the Generator is a mode collapse. The Generator should produce a variety of data that are passed to the Discriminator network. However, the Generator may get stuck in a local minimum and therefore produce the same output each time [90]. Because of that, the Discriminator will not be able to give any feedback to the Generator, and there is nothing more to learn.

Discriminator

The role of a Discriminator is to recognize whether the samples passed to the network are samples from the original dataset or a sample generated by the Generator. For each of the received samples, the Discriminator assigns a probability p of it being fake or real. The objective is to create create a mapping from the received data X, such that D: X→ $[0,1]$. The main objective of the Discriminator is to correctly recognize between samples generated artificially and ones from the original dataset.

One of the pitfalls of the Discriminator might be a diminished gradient; when the Discriminator always correctly predicts fake samples, then there is nothing to learn for the Generator.

Zero-Sum Game

Because of the two main objectives of the networks in GANs, the networks are competing with each other. The Generator has to maximize the value of the feedback from the Discriminator. The Discriminator has to maximize the value of guessing correctly if the image is real. This process resembles the known game theory concept of a zero-sum game. Each side, G or D, can only gain if the other loses.

3.2.1 CTGAN

One of the most promising results in synthetic data generation was obtained with conditional GANs (CTGAN) [113]. In order to overcome some drawbacks of normal GANs, CTGANs start with different modelling of data representation. The data is transformed with mode-specific normalization. Moreover, it contains three key novel elements: the conditional vector, the generator loss, and the training by sampling method.

The conditional vector generates samples from the conditional probability of the data for the chosen categorical variable. All categorical variables are treated as a one-hot encoded vector. For such, the mask is applied, where the given condition is enforced. A conditional vector concatenates masks and results in a one-zero vector. Generator loss ensures that the Generator is forced to produce variables from one category. Given the condition for the Generator, it should output something similar, such that the masks for the input and output data match. The cross-entropy is calculated between the two masks. The added loss creates a constraint that forces the Generator to generate samples according to the conditional vector. This can be achieved thanks to training-by-sampling, where the data is sampled according to the log-frequency of each category. Based on this, the conditional vector is chosen. These three improvement criteria ensure that the data is sampled evenly, according to the distribution of the original dataset.

One of the shortcomings of CTGANs is that small training datasets can have a negative impact on the performance of the tool. Moreover, this methodology does not perform well if missing values are present in the datasets. Therefore, this limitation impacts real-life data application since existing data tends to be imperfect.

3.2.2 Pate-GAN

Pate-GAN is a variation of the GANs network. The main difference is that Pate-GAN uses the Pate framework for data generation. The Pate framework is Private Aggregation of Teacher Ensembles, creating a teacher-student role while learning and training the network. The idea begins that teacher models are used to train sensitive data, which does not have to be public. Therefore it allows transferring the knowledge from teacher models to student models without directly accessing sensitive infractions [76].

This idea is extended by Jordon et al., in which Generator G is the standard GAN framework [50]. All the changes are introduced in Discriminator D, where the Pate-scheme replaces it. As a consequence of this, the data is split into k smaller datasets, and k teacher-discriminators are trained. Moreover, each is trained in the same way as usual with GAN, changing the smaller part of the dataset. Furthermore, an ϵ value is added, which adds the noise to the model during training to create differential privacy guarantees [6].

The main difference comes when this framework introduces a student-discriminator. This part is trained on the output data from the teacher-discriminator [50]. An example of how the networks look can be found in Figure 3.3.

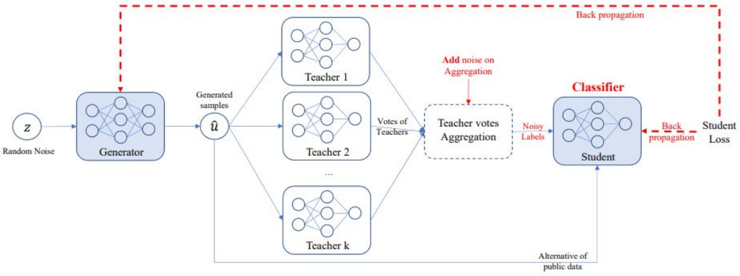

FIGURE 3.3: Block diagram of the training procedure for the student-discriminator and the generator. The student-discriminator is trained using noisy teacher-labelled generated samples [50]
Source: [50]

The main limitation of this framework is that the student-discriminator is only trained on the data passed from the teacher-discriminator. This data is ensured to be deferentially private because of the additional condition that was added by the authors as well as the original Pate mechanism. As a consequence, all the samples passed are with added noise. This can be problematic since we want a student-discriminator to become proficient in recognizing the samples as real or fake. Because the teacher-generator relies on feedback from the student-generator, if the feedback is reliable nor insightful, it might lead to the deterioration of the created sample quality.

Data Undersampling

The method used to undersample the data in this Framework is Condensed Nearest Neighbor (CNN). This method seeks a collection of the data points that results in no loss in the performance of the data, and this is referred to as a minimal consistent set [42]. It identifies the borderline cases from the given dataset. This is feasible because CNN evaluates each of the data samples and adds them to the final set if the samples cannot be correctly classified by the current content of the final set. Such that E is the original dataset, a subset E_i is created that contains all the positive examples from the dataset and one randomly selected negative example. Then the classification of k nearest neighbor is performed, and all the misclassified examples are moved into E_i. This is repeated until there are no more data points to be added to E_i. The procedure is illustrated in 3.4.

The main drawback of this method is that it is possible to retain the cases that introduce noise to the data and do not contribute to the boundary.

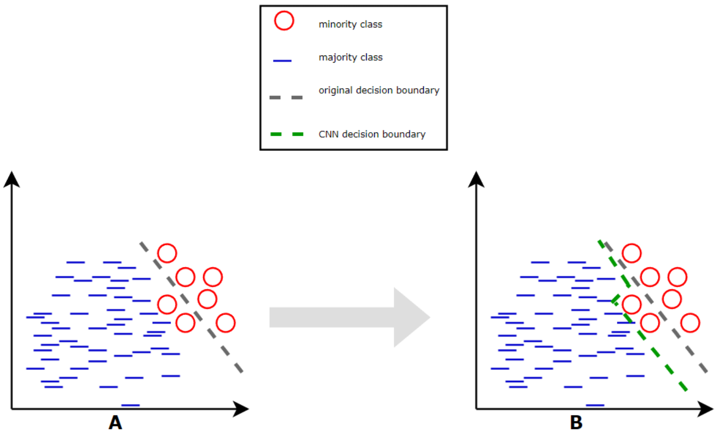

FIGURE 3.4: CNN algorithm [50]

Chapter 4

The Intag Framework

In this thesis, we propose the Intag[1] framework to increase the quality of produced data. The framework is available on GitHub [2]. It consists of the following set of components: Data pre-processing, generating, and post-processing. All of these will be discussed in the following sections.

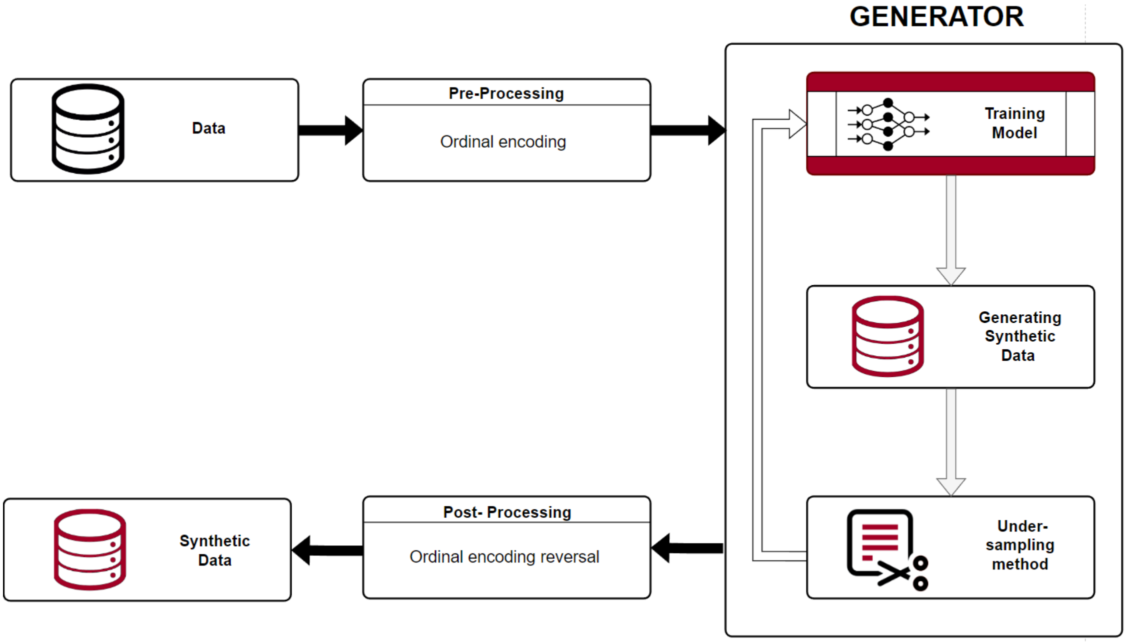

FIGURE 4.1: Framework visualisation

The framework consists of 3 parts. The first part takes the dataset as an input and then pre-processes it. This processed data then enters the main part of the framework. The main component of the framework consists of the following phases. It starts with the Training Model. Then the synthetic data is generated. In the next step, the data is undersampled. After this, the dataset is prepared for Post-Processing, after which a classification model can be trained and evaluated.

4.1 Data Pre-processing

In order to ensure the quality and consistency of the datasets, each of the datasets is cleaned of missing values. Moreover, the representation of the values is processed through ordinal encoding. The reason for that is twofold: firstly, it has been shown

[1] from Arabic - *to generate*

[2] $https://github.com/qahtanaa/SynDataGen_Ola$

that ordinal encoding performs similarly to one-hot encoding but shortens training time [22]; secondly, the main methodology of the framework is dependent on data dimensions. Therefore, using one-hot encoding would not be beneficial to the framework's design.

A user is also able to specify the variables that are considered sensitive, but this does not change anything in the Pre-Processing step.

4.2 Generator

The main component of this part of the framework starts with the model training.

Within the framework, it is possible to indicate which method of synthetic data generation will be used 4.2. Each of the three methods is described in section 3. The main method developed for this system is a modified version of Pate-GAN. A user is able to specify a sensitive attribute, which in this thesis will be a minority class. This will force CTGAN and SMOTE to produce more samples with that attribute. The number of samples will be calculated in such a way that the number of produced data samples generated will equal the imbalance between majority and minority classes. If the sensitive attributes are not specified, all the methods will double the data that has been input into the network.

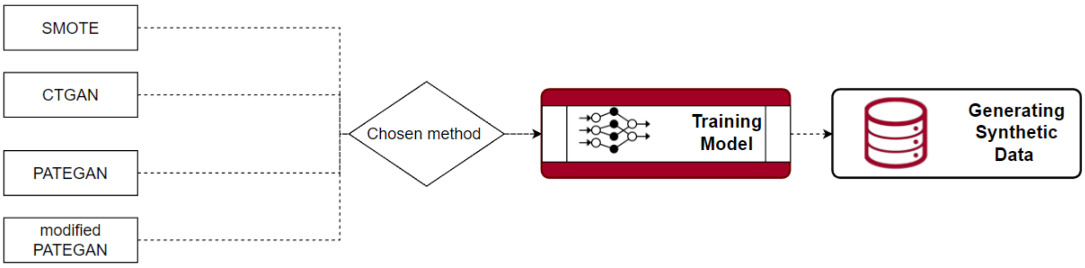

FIGURE 4.2: Method choice for the dataset training

Modified Pate-GAN

Numerous changes were applied in order to improve this method. This section will describe the changes that were made to the original Pate-GAN described above. A series of modifications were made in order to improve the quality of the generated synthetic data. The most significant changes were made to the network, such that the Pate mechanism was preserved but the hard constraint on conditioning the generator output was removed. This implies that the model still splits the data into k different parts. The multiple teachers' models were trained on the disjointed partitions of the data. This preserves the privacy constraint but does not follow the model of differential privacy proposed by Dwork [28] [76].

In line with this, the students were trained on the outputs from the teachers' generators, such that the outputs were differentially private with regard to the original dataset. As argued in [76], multiple teacher mechanisms were enough to achieve the privacy constraint.

On the other hand, only the artificial data was passed to the student generator, with minimal noise aggregation within.

Because the limitation of the original Pate-GAN was that the data was trained on the samples that were restricted by the differential privacy constraint, the samples were not similar to the original samples. Because of this the differential privacy constraint was dropped. This implies that the student discriminator is able to give more meaningful feedback to the generator while still receiving only artificial samples.

Furthermore, there were multiple changes to the architecture of the network. The model adapts the number of hidden layers depending on the input data. Choosing the right number of hidden layers is tricky. It critically depends on the number of training examples and the complexity of the classification the model is trying to learn [52]. If the number of hidden layers is too big, the network might overfit; on the other hand, if it is too small, it might not learn the data representation. Therefore, this method uses a simple heuristic, based on which it decides how many hidden layers the network will have. If the level of complexity within the data is small, the data dimension is small. The network will then use one hidden layer. As the data complexity increases with the number of attributes, the network will grow. However, in order to avoid an overly-complex network, the network never uses more than three hidden layers. This was changed from the original model, where the number of hidden layers was held constant at one.

In order to ensure the stability of the method with the proposed changes, the learning rate of GANs was changed. Mescheder et al. showed that GANs converge more easily while the learning rate is small [67]. Therefore, the value of the learning rate has been changed to $\lambda = 10^{-5}$. Another small change was made on how the network passes binary variables. In the original model, the entire dataset was scaled to values between 0 and 1. In the proposed version, only categorical variables are scaled.

Last but not least, changes were made to model evaluation. Within each iteration, the model produces a set of artificial samples. Each of those sets is evaluated using two measures, namely AUROC and AUPRC. In order to represent a real-life setting, the improved model is not only evaluated on the previous setting but also on the accuracy of its different predictions. The synthetic data is then returned.

More changes to the network were made, but the results were not promising. These can be found in Appendix B.

Undersampling

The next step is to undersample the data. This is done with Condensed Nearest Neighbor. This method was chosen to ensure data samples of high quality are returned. Because of the way CNN works, it only returns the samples that create a minimal consistent set. This implies that the samples chosen to be returned as final synthetic data will be the ones that contribute to the boundary between the different variables. In the undersampling step, the nearest neighbor is chosen to be $k = 1$. There were two main reasons for choosing this number. First of all, the smaller k, the lower the complexity of the algorithm. Because the framework is already complex, there was no need for adding extra complexity that would increase execution time for large datasets. Evidencing this, for the datasets used in this research, the accuracy after simple regression classification is best using $k = 1$, or $k = 3$, as shown in 4.3. Therefore it was not necessary to increase the complexity by increasing k.

4.3 Post-Processing

The last step in this framework is to post-process the data that was created. Because the synthetic data that was generated is also ordinally encoded, this step will reverse this. Therefore, the data that is numerical is changed to be assigned to the proper

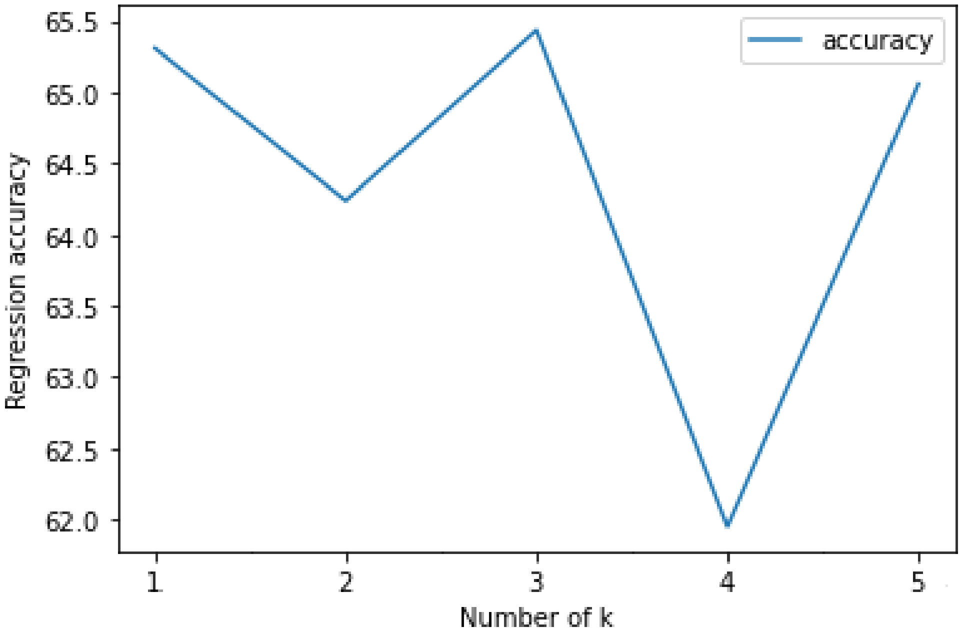

Figure 4.3: CNN with different k

categories. As such, the encoding is reversed. The synthetic data created uses the exact structures and categorical variables as the original data fed into the network. This synthetic data is then used for the experiments described in the next section.

Chapter 5

Experiments

This section will describe different experimental setups used in this study. All of the experiments will be conducted on the datasets that are presented in 5.1. This section will first introduce the datasets. Then it will introduce the experiments. The main goal of the experimental setup is to discover if the developed method and framework can increase the quality of the generated synthetic data. Moreover, the aim is to analyse if the framework is suitable to improve fairness outcomes through data generation and solve the imbalanced dataset problem.

All of the synthetic data will be evaluated using Machine Learning Utility with three main methods: Decision Tree, Regression and Multi-layer Perceptron. More details about this can be found in Appendix A.

5.1 Dataset

In order to make this research comparable to others in the field; it will use databases that are widely-known and used in similar problems. The experiments will focus on evaluating the generated synthetic data, as well as the direct application of the developed framework to fairness. Because different experiments will be conducted, different datasets will be used. The reason behind this is that if the imbalanced dataset were to be used to check the quality of the generated synthetic data, the data would carry a bias. The evaluation of the data would then be biased too. Therefore, this section will be split into two. It will first describe the datasets that are used to strictly evaluate the quality of the data and then the datasets that will be used in fairness evaluation. All of the datasets will be used in the evaluation of the framework.

5.1.1 Pima Indians Diabetes Database

The National Institute of Diabetes and Digestive and Kidney Diseases is the source of this dataset. Based on specific diagnostic metrics present in the dataset, the dataset's goal is to diagnostically forecast whether a patient has diabetes or not. These instances were chosen from a bigger database under several restrictions. Mainly, all patients at this facility are Pima Indian women who are at least 21 years old [96].

The dataset consists of one target variable, Outcome, and some medical predictor variables. The patient's BMI, insulin level, age, number of previous pregnancies, and other factors are predictor variables. This is a relatively small dataset, which is always a challenge for data replication techniques because there is less information that the dataset carries within. This is an example of a dataset that is balanced, and it will be used to evaluate the synthetic data generated by the framework.

PIMA	
Instances	769
Attributes	8

TABLE 5.1: Pima Indians Diabetes Database

NHANES	
Instances	5515
Attributes	16

TABLE 5.2: NHANES Diabetes Dataset

5.1.2 NHANES Data

The National Health and Nutrition Examination Survey (NHANES) is a set of studies designed to assess the health and nutritional status of adults and children in the United States. This dataset can also be used to predict diabetes. The data consists of 16 variables chosen by feature selection to help discover if a person has diabetes [94]. This dataset is used for binary classification prediction.

This is an example of a dataset that is balanced, and it will be used to evaluate the generated synthetic data.

5.1.3 Adult

The Adult dataset is built upon data from the US Census from 1994, and it is used to predict if a given person has a yearly income that passes the threshold of 50,000. The dataset consists of data on over 48,000 individuals described by 14 variables. It is skewed towards individuals that do not pass the threshold of 50,000, which makes it imbalanced.

Moreover, within the 14 variables presented in the dataset, we can find information such as sex, race or age. All of these variables are considered to be protected variables. Table 5.3 shows the measurements for the fairness variables.

5.1.4 German

Another benchmark dataset used in this research is the German dataset. This data consists of only about 1,000 entries, with twenty descriptive variables. The German dataset classifies if a given individual can repay a taken loan or not; therefore, it is a risk-averse dataset.

The German dataset is an example of an imbalanced dataset, where only 30% of the bank clients are categorized as good customers who would repay the loan. This data is similar to previous datasets, but it has significantly fewer entries.

Within the twenty variables, at least two are considered protected variables, such as sex, age, and whether or not the person is a foreign worker. Table 5.4 represents the summary statistics of the data, as well as including the first fairness statistics.

Adult Dataset

Instances	48842
Attributes	14
Male vs. Female	30527 vs. 14695
Disparate impact	0.29
Statistical parity	-0.33
White vs. Non-White	38903 vs. 6319
Disparate impact	0.55
Statistical parity	-0.18

Table 5.3: Adult Dataset

German Dataset

Instances	1000
Attributes	20
Male vs. Female	690 vs. 310
Disparate impact	0.97
Statistical parity	-0.02
Old vs. Young	vs.
Disparate impact	0.48
Statistical parity	-0.3

Table 5.4: German Dataset

5.1.5 COMPAS

The Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) dataset is taken from the very popular software, used in the American Judicial System. This tool operates as a decision-making aid for judges in the United States. It analyses a person that committed a crime and offers a score of how likely this person would be to commit a crime again. It has been argued that COMPAS is biased in favour of white people [51].

The database contains 28 variables, of which two are considered to be protected attributes: sex and race. This dataset is processed as it was by ProPublica [24].

Compas Dataset	
Instances	6167
Attributes	28
Male vs. Female	4994 vs. 1173
Disparate impact	0.59
Statistical parity	-0.36
White vs. Non-White	2100 vs.4067
Disparate impact	0.75
Statistical parity	-0.18

Table 5.5: COMPAS Dataset

5.2 Experimental setup

5.2.1 Intag vs other methods

The first experiment will be conducted to compare the framework that used modified Pate-GAN with other methods of synthetic data replication. Methods that this research will compare are described in 3. The choice for that is as follows: SMOTE is one of the best-known methods of oversampling, while GANs are becoming increasingly widely-recognized in data oversampling.

The idea behind this experiment is simple: create synthetic data with the Intag framework and then create alternative synthetic data using the methods mentioned above. Then compare the quality of the data generated with Intag and other methods, which will be measured by Machine Learning Utility.

This experiment will be performed on the datasets Pima Indians Diabetes Database (Test I) and NHANES Diabetes Dataset (Test II), as those two datasets are made up of balanced data. Moreover, those datasets are examples of real-life data used in medicine.

5.2.2 Intag Framework learning setup

We introduce different training testing settings to empirically validate the quality of the generated dataset by the Intag framework. This evaluation will check all the methods that are available in the framework, namely modified Pate-GAN, Pate-GAN, SMOTE, and CTGAN. This experiment will show how well the framework performs on different datasets (balanced vs imbalanced). It is also designed to check how well the framework can reproduce the statistical properties of the data. This will be done by measuring the Machine Learning Utility with the synthetic datasets.

Three types of experimental setup will be conducted. Each of these is described below.

- Learning from balanced data: the framework is tested on all five datasets with different methods of synthetic data generation. The goal of this experiment is to compare modified Pate-GAN to other synthetic data generation methods.

- Learning with 50% synthetic data: in order to check if the framework can replicate the statistical properties of the data, a different setting will be used. First, the framework will run on the original data. Doing this, it will output the first synthetic dataset. Next to the framework, Intag will run again. This time, as an input, it will take a dataset that consists of 50% of the original dataset, as well as 50% of synthetic data generated. This is called *Hybrid Synthetic Data*. If the quality of the generated data is preserved, the Machine Learning Utility measures will not change significantly.

- Learning using pure synthetic data: another way to check if the framework can replicate the data and its properties is to feed it *Fully Synthetic Data* and evaluate it. Therefore this third setting will focus on first creating a dataset of synthetic data. After this, the synthetic data will be fed into the framework again and then evaluated. If the quality of the generated data is preserved, the Machine Learning Utility measures will not change significantly.

5.2.3 Application to Fairness

The last experiment in this thesis will test if the framework can be used as an aid in fairness application. The framework will focus on data replication in the three datasets that are imbalanced, namely German, Adult, and COMPAS. All of these datasets carry a bias within because some of the classes are misrepresented, such that there is an imbalance between samples. As discussed, a solution to this is to create high-quality synthetic data that can even out the imbalance. Two main measures of fairness will be used in this experiment: *Statistical parity difference* and *Disparate impact*.

In this experiment, the framework with its different data replication techniques will be used to create synthetic data, and then the aforementioned measures will be compared against the original datasets.

Chapter 6

Results

This chapter will discuss and highlight the findings of the experiments of this thesis. Moreover, it will discuss the strengths and weaknesses of the framework. Only the most interesting experiments were presented in this thesis. More can be found in Appendix. This chapter will be structured as follows: First part will present the results of an experiment that compares the Intag framework against other methods. The second section shows the results on how well the framework can replicate the data and, by extension, how machine learning algorithms can learn from it. The last section and the last experiment present the results of the Intag framework and its application in fairness.

6.1 Intag vs other methods

The first experiment aims at evaluating the Intag framework with other methods of data replication. Two datasets were used to evaluate this, namely PIMA and NHANES. The results can be found in Tables 6.1 and 6.2.

As shown in Table 6.1, which evaluates the framework against other methods of synthetic data generation for the PIMA dataset, the framework with modified Pate performs the best in terms of accuracy of the synthetic data. The average improvement upon other methods is an eight percentage points increase. Moreover, it also performs best in terms of the F1 score, and there is a slight if the negligible difference between the ROC curve score. The F1 score, as well as ROC curve averages, indicate good classifier choices for the data.

Table 6.2 shows the results for the NHANES dataset. As with the results of the previous experiment, there is an increase in accuracy regarding the framework. Intag records an 11.33 percentage point increase in accuracy over other methods. Contrary to the previous part of the experiment, the difference between ROC curve values is not negligible. Specifically, the modified Pate framework performs much better than the CTGAN method here.

Both datasets provide consistent results for this experiment. There is an increase in accuracy over the alternative methods. The framework also shows the best results for the F1 score.

Table 6.1: Intag vs other methods of data replication on PIMA dataset

	Pate modified				Pate-GAN				CTGAN				SMOTE			
	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC	Acc	A.0.	F1	AUR
Decision Tree	0.85	0.47	0.84	0.80	0.81	0.48	0.81	0.81	0.76	0.56	0.76	0.76	0.79	0.76	0.80	0.8
Regression	0.83	0.49	0.82	0.78	0.74	0.48	0.74	0.75	0.65	0.63	0.64	0.63	0.75	0.76	0.75	0.75
MLPClassifier	0.80	0.48	0.79	0.75	0.80	0.51	0.80	0.80	0.77	0.52	0.76	0.75	0.68	0.62	0.67	0.68
Average	**0.83**	0.47	**0.82**	0.78	0.78	0.49	0.79	**0.79**	0.73	0.57	0.72	0.71	0.74	**0.71**	0.74	0.75

Acc stands for accuracy of the model
A.0. stands for the accuracy of the model on the original dataset, while trained on the synthetic data.

Table 6.2: Intag vs other methods of data replication on NHANES dataset

	Pate modified				Pate-GAN				CTGAN				SMOTE			
	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC	Acc	A.0.	F1	AUR
Decision Tree	0.82	0.53	0.82	0.80	0.80	0.45	0.80	0.81	0.65	0.51	0.57	0.56	0.75	0.70	0.75	0.75
Regression	0.67	0.50	0.69	0.67	0.81	0.48	0.81	0.81	0.62	0.58	0.6	0.56	0.76	0.65	0.76	0.76
MLPClassifier	0.79	0.50	0.79	0.78	0.78	0.41	0.71	0.78	0.63	0.55	0.61	0.59	0.77	0.68	0.76	0.76
Average	**0.84**	0.51	**0.77**	0.74	0.79	0.45	0.77	**0.80**	0.63	0.54	0.59	0.57	0.76	**0.74**	0.76	0.76

Acc stands for accuracy of the model
A.0. stands for the accuracy of the model on the original dataset, while trained on the synthetic data.

6.2 Learning with synthetic data

6.2.1 Learning from balanced data

This experiment aims to show and evaluate how well the framework performs with different data replications methods. First Setting A: Original data is fed into the framework, and then the synthetic data is evaluated. For this experiment, we tested balanced as well as imbalanced datasets; this was done to see if the framework could deal with flawed datasets. Five datasets were tested as part of this experiment.

Table 6.3 shows the results run on the German dataset. It shows that the modified Pate-GAN framework performs best in terms of accuracy, as well as F1 measure and ROC. This stands for all the datasets tested, as shown in 6.5, 6.4, 6.6, and 6.7. The framework has a better performance on the datasets that are not imbalanced, which are 6.6 and 6.7.

Moreover, all the results are consistent with the expectations of this research. Modified Pate-GAN outperforms other methods. Furthermore, the framework performs better when dealing with balanced datasets. As the undercutting method within the framework ensures that fairness will be preserved at the cost of accuracy, this is in line with expectations.

Table 6.3: Setting A: using original **German** dataset to test Intag framework with different methods of data replication.

	Pate modified				Pate-GAN				CTGAN			
	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC
Decision Tree	0.74	0.46	0.73	0.69	0.72	0.53	0.71	0.61	0.72	0.41	0.69	0.69
Regression	0.79	0.51	0.79	0.75	0.74	0.46	0.74	0.71	0.63	0.48	0.52	0.51
MLPClassifier	0.58	0.51	0.4	0.5	0.55	0.48	0.4	0.48	0.50	0.53	0.4	0.51
Average	**0.70**	0.49	**0.64**	**0.64**	0.67	0.49	00.61	0.6	0.61	0.47	0.54	0.57

Acc stands for accuracy of the model
A.0. stands for the accuracy of the model on the original dataset, while trained on the synthetic data.

Table 6.4: Setting A: using original **Adult** dataset to test Intag framework with different methods of data replication.

	Pate modified				Pate-GAN				CTGAN			
	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC
Decision Tree	0.80	0.40	0.80	0.77	0.73	0.51	0.72	0.67	0.88	0.59	0.88	0.88
Regression	0.69	0.50	0.65	0.6	0.62	0.54	0.61	0.6	0.51	0.41	0.4	0.48
MLPClassifier	0.61	0.48	0.51	0.52	0.51	0.49	0.44	0.44	0.49	0.59	0.38	0.49
Average	**0.7**	0.46	**0.65**	**0.63**	0.62	0.51	0.59	0.57	0.63	**0.53**	0.55	0.61

Acc stands for accuracy of the model
A.0. stands for the accuracy of the model on the original dataset, while trained on the synthetic data.

Table 6.5: Setting A: using original **Compas** dataset to test Intag framework with different methods of data replication.

	Pate modified				Pate-GAN				CTGAN			
	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC
Decision Tree	0.77	0.43	0.76	0.71	0.65	0.48	0.63	0.68	0.91	0.86	0.66	0.65
Regression	0.77	0.53	0.77	0.72	0.65	0.48	0.64	0.59	0.84	0.92	0.53	0.54
MLPClassifier	0.74	0.54	0.73	0.70	0.69	0.56	0.68	0.67	0.92	0.8	0.31	0.5
Average	**0.76**	0.50	**0.75**	**0.71**	0.66	0.50	0.65	0.65	0.58	0.50	0.50	0.56

Acc stands for accuracy of the model
A.0. stands for the accuracy of the model on the original dataset, while trained on the synthetic data.

Table 6.6: Setting A: using original **PIMA** dataset to test Intag framework with different methods of data replication.

	Pate modified				Pate-GAN				CTGAN			
	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC
Decision Tree	0.85	0.47	0.84	0.80	0.77	0.58	0.75	0.72	0.79	0.48	0.77	0.76
Regression	0.83	0.49	0.82	0.78	0.77	0.60	0.76	0.73	0.63	0.49	0.62	0.6
MLPClassifier	0.80	0.48	0.79	0.75	0.78	0.61	0.74	0.75	0.75	0.47	0.73	0.70
Average	**0.83**	0.47	**0.82**	**0.78**	0.77	**0.59**	0.75	0.73	0.73	0.48	0.71	0.69

Acc stands for accuracy of the model
A.0. stands for the accuracy of the model on the original dataset, while trained on the synthetic data.

Table 6.7: Setting A: using original **NHANES** dataset to test Intag framework with different methods of data replication.

	Pate modified				Pate-GAN				CTGAN			
	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC
Decision Tree	0.82	0.53	0.81	0.80	0.71	0.52	0.70	0.65	0.70	0.41	0.67	0.64
Regression	0.67	0.50	0.67	0.66	0.70	0.45	0.69	0.64	0.70	0.50	0.69	0.69
MLPClassifier	0.79	0.50	0.79	0.77	0.64	0.75	0.68	0.60	0.63	0.21	0.62	0.59
Average	**0.76**	0.51	**0.76**	**0.74**	0.68	**0.55**	0.69	0.63	0.68	0.37	0.66	0.64

Acc stands for accuracy of the model
A.0. stands for the accuracy of the model on the original dataset, while trained on the synthetic data.

6.2.2 Learning with 50% synthetic data

The aim of Setting B in this experiment is to use the framework with *Hybrid Synthetic Data*, which is a mix of synthetic and original data. First, the framework generates synthetic data (according to the method chosen), and then it will be mixed with the original data. Finally, the results will be evaluated in the same way as in previous settings. All of the experiments were conducted on the five aforementioned datasets, and all of the results can be found below in Tables 6.8, 6.9, 6.10, 6.12 and 6.13.

The results are comparable to the findings of Experiment 2 Setting A. Modified Pate-GAN outperforms other methods.

TABLE 6.8: Setting B: using *Hybrid Synthetic Data* **German** dataset to test Intag framework with different methods of data replication.

	Pate modified				Pate-GAN				CTGAN			
	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC
Decision Tree	0.98	0.55	0.98	0.97	0.77	0.50	0.76	0.74	0.79	0.43	0.78	0.76
Regression	0.79	0.40	0.79	0.78	0.71	0.53	0.70	0.68	0.70	0.47	0.69	0.66
MLPClassifier	0.64	0.3	0.5	0.5	0.58	0.7	0.73	0.5	0.51	0.46	0.4	0.48
Average	**0.80**	0.41	**0.76**	**0.75**	0.68	**0.57**	0.73	0.64	0.67	0.45	0.62	0.63

Acc stands for accuracy of the model
A.0. stands for the accuracy of the model on the original dataset, while trained on the synthetic data.

TABLE 6.9: Setting B: using *Hybrid Synthetic Data* **Adult** dataset to test Intag framework with different methods of data replication.

	Pate modified				Pate-GAN				CTGAN			
	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC
Decision Tree	0.77	0.36	0.76	0.71	0.68	0.47	0.66	0.65	0.68	0.47	0.66	0.65
Regression	0.68	0.56	0.62	0.57	0.62	0.67	0.61	0.6	0.62	0.4	0.53	0.54
MLPClassifier	0.57	0.44	0.45	0.51	0.58	0.51	0.42	0.38	0.46	0.65	0.31	0.5
Average	**0.67**	0.45	**0.61**	**0.60**	0.60	**0.55**	0.56	0.54	0.58	0.50	0.50	0.56

Acc stands for accuracy of the model
A.0. stands for the accuracy of the model on the original dataset, while trained on the synthetic data.

Table 6.10: Setting B: using *Hybrid Synthetic Data* **Compas** dataset to test Intag framework with different methods of data replication.

	Pate modified				Pate-GAN				CTGAN			
	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC
Decision Tree	0.79	0.5	0.79	0.73	0.70	0.52	0.69	0.68	0.74	0.49	0.71	0.68
Regression	0.80	0.49	0.80	0.75	0.75	0.51	0.75	0.59	0.64	0.48	0.57	0.56
MLPClassifier	0.78	0.53	0.77	0.72	0.75	0.56	0.75	0.67	0.66	0.48	0.64	0.61
Average	**0.79**	**0.51**	**0.79**	**0.73**	0.66	0.50	0.73	0.65	0.68	0.48	0.64	0.62

Acc stands for accuracy of the model
A.0. stands for the accuracy of the model on the original dataset, while trained on the synthetic data.

Table 6.11: Test I

Table 6.12: Setting B: using *Hybrid Synthetic Data* **PIMA** dataset to test Intag framework with different methods of data replication.

	Pate modified				Pate-GAN				CTGAN			
	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC
Decision Tree	0.88	0.50	0.88	0.86	0.80	0.53	0.80	0.78	0.79	0.46	0.78	0.78
Regression	0.85	0.52	0.84	0.79	0.79	0.56	0.79	0.76	0.65	0.51	0.65	0.64
MLPClassifier	0.78	0.51	0.75	0.66	0.72	0.52	0.69	0.67	0.69	0.45	0.68	0.67
Average	**0.84**	0.51	**0.82**	**0.77**	0.77	**0.54**	0.76	0.73	0.71	0.47	0.70	0.70

Acc stands for accuracy of the model
A.0. stands for the accuracy of the model on the original dataset, while trained on the synthetic data.

Table 6.13: Setting B: using *Hybrid Synthetic Data* **NHANES** dataset to test Intag framework with different methods of data replication.

	Pate modified				Pate-GAN				CTGAN			
	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC
Decision Tree	0.88	0.44	0.88	0.86	0.84	0.57	0.83	0.79	0.79	0.41	0.76	0.76
Regression	0.77	0.46	0.77	0.74	0.81	0.54	0.80	0.77	0.70	0.52	0.68	0.67
MLPClassifier	0.94	0.39	0.96	0.94	0.71	0.21	0.70	0.70	0.72	0.49	0.72	0.70
Average	**0.86**	0.43	**0.87**	**0.85**	0.78	0.44	0.77	0.75	0.74	**0.47**	0.72	0.71

Acc stands for accuracy of the model
A.0. stands for the accuracy of the model on the original dataset, while trained on the synthetic data.

6.2.3 Learning using pure synthetic data

The last experiment of this part has the same aim, to find how well the framework can replicate data. The difference with previous setups is that here, it will evaluate how well the framework can deal with data that is *Fully Synthetic*. This means that the data will first be replicated with the framework; then, this fully synthetic data will be replicated, and then all of it will be evaluated against the original data.

The experiment was conducted on the same five datasets. All results can be found below in tables 6.14, 6.15, 6.16, 6.17 and 6.18.

The framework using modified Pate-GAN still performs the best compared to the other methods. Again, the framework performs better on datasets that are balanced.

The most interesting development of this experiment is that the accuracy of the original dataset increases, contrary to the other set of experiments. This holds for all the datasets, but it is primarily with balanced data, where the accuracy can reach 60%. This is an interesting development. It shows that the *pure synthetic data* fed to the network can be replicated very well after it goes into the framework with undercutting.

Table 6.14: Setting C: using *Fully Synthetic Data* **German** dataset to test Intag framework with different methods of data replication.

	Pate modified				Pate-GAN				CTGAN			
	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC
Decision Tree	0.82	0.56	0.81	0.78	0.74	0.65	0.73	0.68	0.77	0.5	0.75	0.76
Regression	0.68	0.61	0.63	0.58	0.69	0.47	0.69	0.62	0.55	0.35	0.53	0.54
MLPClassifier	0.60	0.44	0.50	0.51	0.56	0.47	0.47	0.50	0.34	0.36	0.28	0.32
Average	**0.70**	**0.54**	**0.65**	**0.62**	0.66	0.53	0.63	0.6	0.55	0.40	0.52	0.54

Acc stands for accuracy of the model
A.0. stands for the accuracy of the model on the original dataset, while trained on the synthetic data.

Table 6.15: Setting C: using *Fully Synthetic Data* **Adult** dataset to test Intag framework with different methods of data replication.

	Pate modified				Pate-GAN				CTGAN			
	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC
Decision Tree	0.77	0.36	0.76	0.71	0.68	0.47	0.68	0.65	0.68	0.47	0.66	0.65
Regression	0.68	0.56	0.62	0.57	0.68	0.67	0.64	0.60	0.62	0.40	0.53	0.54
MLPClassifier	0.57	0.44	0.45	0.51	0.59	0.51	0.42	0.38	0.46	0.65	0.31	0.50
Average	**0.67**	0.45	**0.61**	**0.62**	0.65	**0.55**	0.58	0.54	0.59	0.50	0.50	0.56

Acc stands for accuracy of the model
A.0. stands for the accuracy of the model on the original dataset, while trained on the synthetic data.

TABLE 6.16: Setting C: using *Fully Synthetic Data* **Compas** dataset to test Intag framework with different methods of data replication.

	Pate modified				Pate-GAN				CTGAN			
	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC
Decision Tree	0.75	0.53	0.71	0.66	0.70	0.49	0.70	0.67	0.70	0.49	0.66	0.66
Regression	0.74	0.51	0.71	0.66	0.70	0.50	0.67	0.62	0.64	0.48	0.62	0.6
MLPClassifier	0.65	0.55	0.56	0.56	0.60	0.59	0.59	0.57	0.58	0.48	0.57	0.54
Average	**0.71**	**0.53**	**0.66**	0.62	0.66	**0.53**	**0.65**	0.65	0.64	0.48	0.60	0.62

Acc stands for accuracy of the model
A.0. stands for the accuracy of the model on the original dataset, while trained on the synthetic data.

TABLE 6.17: Setting C: using *Fully Synthetic Data* **PIMS** dataset to test Intag framework with different methods of data replication.

	Pate modified				Pate-GAN				CTGAN			
	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC
Decision Tree	0.95	0.59	0.94	0.90	0.79	0.50	0.79	0.78	0.83	0.55	0.82	0.79
Regression	0.83	0.61	0.82	0.79	0.76	0.47	0.74	0.73	0.81	0.54	0.76	0.72
MLPClassifier	0.92	0.61	0.92	0.90	0.58	0.45	0.54	0.54	0.64	0.50	0.59	0.56
Average	**0.90**	**0.60**	**0.89**	**0.86**	0.71	0.47	0.69	0.68	0.76	0.53	0.72	0.69

Acc stands for accuracy of the model
A.0. stands for the accuracy of the model on the original dataset, while trained on the synthetic data.

Table 6.18: Setting C: using *Fully Synthetic Data* **NHANES** dataset to test Intag framework with different methods of data replication.

	Pate modified				Pate-GAN				CTGAN			
	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC	Acc	A.0.	F1	ROC
Decision Tree	0.83	0.45	0.82	0.78	0.79	0.53	0.79	0.74	0.79	0.41	0.76	0.76
Regression	0.74	0.52	0.73	0.68	0.79	0.56	0.78	0.73	0.60	0.52	0.58	0.57
MLPClassifier	0.78	0.49	0.77	0.76	0.68	0.21	0.64	0.62	0.62	0.49	0.52	0.50
Average	**0.78**	**0.49**	**0.77**	**0.74**	0.75	0.43	0.73	0.69	0.67	0.47	0.62	0.61

Acc stands for accuracy of the model
A.0. stands for the accuracy of the model on the original dataset, while trained on the synthetic data.

6.3 Application in Fairness

6.3.1 German

This experiment aims at evaluating if the framework has an application in fairness[1].

Three datasets were evaluated in this experiment, namely German, Adult and Compas. Each dataset has been replicated by the network with different synthetic data replication methods. Moreover, each dataset has two sets of variables considered protected attributes. The framework is forced to replicate this, especially by SMOTE and CTGAN. If the network was not forced to replicate specific values, SMOTE performance was poor, similar to the experiments before.

Two measures are used to indicate fairness in this research - statistical parity difference and disparate impact.

Firstly, we will examine the framework and its performance on the German dataset. The first protected attribute from the dataset is age, where the unprivileged group is young people below 25 years of age. The results and the comparison with the original can be found in table 6.19. The statistical parity for the original dataset is -0.3, which implies bias in the data. The ideal value for this measure is 0. The dataset is considered fair if the value is between -0.1 and 0.1. All of the data replication methods in the framework are within the boundaries of fairness according to statistical parity.

Another value that was measured to indicate the fairness after the synthetic data is disparate impact. Table 6.19 shows that the value of that measure in the original dataset is 0.48, which indicates bias. The ideal value of this metric is 1.0. A value < 1 implies a higher benefit for the privileged group, and a value >1 implies a higher benefit for the unprivileged group. Within this metric, a dataset is considered fair if the value is between 0.8 and 1.25. Based upon the results, the framework creates fair data according to the disparate impact measure, except when using SMOTE.

Table 6.19: Fairness - German - Age, privileged: Old, unprivileged: Young

	Original	PATE	PATEGAN	SMOTE	CTGAN
Fairness measure					
Statistical parity difference [1]	-0.3	0.11	-0.0014	0.08	-0.015
Disparate impact [2]	0.48	1.17	1.0	1.32	0.969

[1] Fairness for this metric is between -0.1 and 0.1; with ideal value: 0.
[2] Fairness for this metric is between 0.8 and 1.25; with ideal value : 1.0.

Regarding the other set of protected variables, female vs male, the original dataset indicates bias with statistical parity difference. In the table 6.20, we present the results of the dataset that was replicated by the framework.

Statistical parity difference decreased for all methods and is closer to its ideal value of 0. It does show that the network added some bias using modified Pate and SMOTE regarding disparate impact. Both of the values exceed the fairness values. Because the value is bigger than 1, it shows a higher benefit for the unprivileged

[1] The experiment for fairness uses the AIF360 framework, which makes it possible to calculate the scores used in this experiment.

TABLE 6.20: Fairness - German - Gender -Sex, privileged: Male, unprivileged: Female

	Original	PATE	PATEGAN	SMOTE	CTGAN
Fairness measure					
Statistical parity difference [1]	-0.2	0.105	-0.0015	0.16	0.012
Disparate impact [2]	0.97	1.34	0.98	1.53	1.04

[1] Fairness for this metric is between -0.1 and 0.1; with ideal value: 0.
[2] Fairness for this metric is between 0.8 and 1.25; with ideal value : 1.0.

group, in that case - females. Modified Pate-GAN and CTGAN remain close to the ideal value of 1, however.

6.3.2 Adult

The Adult dataset has two sets of classes that are considered protected. In the Adult dataset, we can see the results of the framework for the race class in table 6.21.

Besides SMOTE, all the methods score higher on both of the fairness measures. Modified Pate and Pate-GAN are within the boundary of fairness, where disparate impact is on the border, skewing towards a slight bias towards the unprivileged group. As for CTGAN, the values also improve but not significantly enough to deem this data fair.

TABLE 6.21: Fairness - Adult - Race, privileged: White, unprivileged: Non-white

	Original	PATE	PATEGAN	SMOTE	CTGAN
Fairness measure					
Statistical parity difference [1]	-0.18	0.051	0.031	0.4	-0.159
Disparate impact [2]	0.55	1.18	1.19	0.42	0.72

[1] Fairness for this metric is between -0.1 and 0.1; with ideal value: 0.
[2] Fairness for this metric is between 0.8 and 1.25; with ideal value : 1.0.

The second variable that is considered protected in this dataset is gender. Table 6.22 show the results from the framework. Besides SMOTE, all of the methods show an improvement in fairness. Moreover, modified Pate shows almost perfect scores for fairness. However, CTGAN crosses the boundary of fair data when considering disparate impact by a small margin of 0.03.

6.3.3 COMPAS

The last dataset that will be evaluated for fairness is COMPAS. This is one of the most controversial datasets of the last few years. It is widely used in the US court

Table 6.22: Fairness - Adult - Sex, privileged: Male, unprivileged: Female

	Original	PATE	PATEGAN	SMOTE	CTGAN
Fairness measure					
Statistical parity difference [1]	-0.33	-0.016	-0.085	0.22	0.11
Disparate impact [2]	0.29	1.025	0.904	1.62	1.28

[1] Fairness for this metric is between -0.1 and 0.1; with ideal value: 0.

[2] Fairness for this metric is between 0.8 and 1.25; with ideal value : 1.0.

system. As with the other datasets, there were two classes which were considered protected and for which bias was indicated.

The first class is race, where the privileged group is Caucasian, and the unprivileged group is non-Caucasian. In the original dataset, it is shown that the privileged group has a higher benefit.

In table 6.23, we can see the results of the framework. The framework managed to improve fairness measures with all of the data replication methods. Nonetheless, as with the previous examples value of the disparate impact of CTGAN is on the border of indication of bias. However, it changed from an indication of benefit for the privileged group to the unprivileged group. This might be caused by the fact that the Intag replicated the values that were explicitly given, such as unprivileged classes.

Table 6.23: Fairness - COMPAS - Race, privileged: Caucasian, unprivileged: Not Caucasian

	Original	PATE	PATEGAN	SMOTE	CTGAN
Fairness measure					
Statistical parity difference	-0.18	-0.078	-0.093	-0.0006	0.084
Disparate impact	0.75	0.905	0.99	0.99	1.16

[1] Fairness for this metric is between -0.1 and 0.1; with ideal value: 0.

[2] Fairness for this metric is between 0.8 and 1.25; with ideal value : 1.0.

For the second protected variable in this dataset - gender, the results can be found in table 6.24. As shown, the framework does improve fairness with all of the replication methods within the framework.

It is worth mentioning that all the findings are consistent with the previous experiments.

Table 6.24: Fairness - COMPAS - Sex, privileged: Female, unprivileged: Male

	Original	PATE	PATEGAN	SMOTE	CTGAN
Fairness measure					
Statistical parity difference	-0.36	-0.0015	-0.07	0.076	0.10
Disparate impact	0.59	1.14	0.88	1.11	1.18

[1] Fairness for this metric is between -0.1 and 0.1; with ideal value: 0.

[2] Fairness for this metric is between 0.8 and 1.25; with ideal value : 1.0.

Chapter 7

Conclusion

This chapter will present a summary of this thesis. It will highlight the strengths and weaknesses of the framework. Moreover, it will describe its limitations and outline other ways to improve Intag.

7.1 Summary

Data is the source for decision-making algorithms, but it is not always perfect. Imbalanced datasets are a known problem. These datasets carry inherent bias, such as that not all classes within the dataset are equally represented. Algorithms will replicate the bias, leading to unfair decisions. One of the solutions for that unfairness would be to change the data in such a way that the imbalance is reduced. Unfortunately, this has proven to be a very difficult task.

This thesis has focused on the problems connected to data generation methods, and the difficulties associated with creating data and measuring if synthetic data is of high quality. Moreover, it checks if the replicated data can be used as an aid in fairness problems with imbalanced datasets.

As a solution, it proposes the Intag framework, with a modified Pate-GAN developed for this thesis as the best performing method of data generation.

The framework consists of several steps. First, the data is fed into Intag. In the pre-processing step, it is then encoded. Next, the Generator, the main part of this framework, replicates the data.

The framework focuses on generating data from the specified data classes, which can be the minority classes. In the next step, the synthetic data is subjected to undersampling with Condensed Nearest Neighbor. Then the data is decoded in the post-processing step.

Multiple experiments are conducted to measure and evaluate the performance of the proposed framework. All the experiments conducted in this thesis showed that modified Pate-GAN improves upon other methods of synthetic data generation. The framework performs better on balanced datasets.

One of the most interesting findings from the conducted experiments is that if the data is trained on *pure synthetic data* from the framework, it will return a dataset that can achieve higher performance on the classifier trained on the original data. This is due to the fact that the samples that did not add a lot of information were undersampled by CNN.

Additionally, multiple experiments were conducted to check if the framework improves the fairness within datasets. Three datasets were chosen, and all the experiments consistently show that Intag improves the fairness within the datasets.

The main advantage of using the Intag Framework is that it is possible to choose different methods of data generation and that these different methods will improve the quality as well as fairness values of the underlying dataset. However, not all of the data generation methods performed very well, as empirically shown by the experiments.

7.2 Answers to the research questions

To answer the main research question, it is possible to improve the synthetic data quality upon CTGAN and SMOTE. The two primary data generation methods that can achieve this are modified Pate-GAN and Pate-GAN, although it has been shown in section 5 that the best method is the modified Pate-GAN. One of the other sub-questions, which would be the best measure of synthetic data quality, is answered in section 2.3. This thesis uses the most common ways to indicate the quality of synthetic data. It uses Machine Learning Utility, one of the best ways to measure data quality [114]. Otherwise, it is tough to show that the created dataset has been improved.

As shown the Intag framework is able to improve the accuracy compared to other known methods. The modified Pate-GAN created for that framework performs the best of the various methods of data generation. The increase is stable at a 5 percentage point increase. Moreover, the framework works well with three out of four methods proposed within.

The most interesting finding is that the framework trained on pure synthetic data is able to significantly increase the performance of the classifier trained on the original data, with an increase on average between 5 datasets of 4 percentage points.

As for the last sub-question, the Intag framework did improve the fairness in imbalanced datasets. As shown, Intag significantly improves fairness with three out of four methods that are usable in the framework. Even though the framework forces SMOTE to synthesise the values from the unprivileged class, it still does not improve fairness measures in some cases.

7.3 Limitations

There are a few limitations of the Intag framework. The first limitation of the framework is connected to synthetic data generation. The primary method, modified Pate-GAN, usually works well with numerical data. Yet it could improve on categorical data or textual data. If improved, it would be possible to use the framework on more datasets, such as those where it is not possible to encode all the variables, for instance, textual data such as Amazon Product Dataset [8].

One of the limitations of the framework is the way the data is undersampled using CNN. It is possible that CNN undersampling will leave out samples that are noisy and do not add to the boundary. Therefore, noisy samples could be included in the generated synthetic data. This influences the accuracy of the classifiers as synthetic data produced includes samples of lower quality.

7.4 Future work

There are multiple ways the Intag framework can be improved. The first and most important one is to further develop the modified Pate-GAN method. One potential avenue of further research is to investigate if the proposed architecture is optimal or whether a different architecture would improve accuracy. Drawing on the framework's limitations, one way to improve the framework would be to make it possible for all types of data, including textual data, to be replicated. This would, for example, make it possible to generate synthetic data for the Amazon Product Dataset.

Another improvement to the framework would be to change the undersampling of CNN. For instance, this could be done by creating an algorithm that takes k nearest neighbors and, based on the similarity score, decides if the sample is good. Nevertheless, this might be tricky for imbalanced datasets, so this should be implemented in a careful manner.

It would also be possible to improve upon the CNN method. An example could be to implement CNN with Tomek links or to choose another method that might keep high-quality samples.

A way to improve the framework is to create boundaries of the classes and, based on that, replicate the samples. Hard boundaries will be created between the samples, so that all the new instances will be clustered within.

Concluding, this thesis sought to develop a new method of generating synthetic data, which would increase the quality of the synthetic data over previously-developed methods. While much work remains to be done, the framework proposed within this thesis is a step forward. Among other applications, this framework may help mitigate bias within data, leading to a significant improvement in the fairness of machine learning models.

Appendix A

Predictive algorithms

All different setups will be used to train three different predictive algorithms that are used in this paper.

A.1 Regression

This research will use regression as a tool. Regression is a simple yet powerful predictor. Very often, it is used as a benchmark algorithm. The first algorithm will be the easiest regression, usually used as a benchmark classification algorithm. In regression, the aim is to obtain a value of a dependent variable based on the input of independent variables. The relation between this is depicted as a linear equation:

$$y = X\beta + \epsilon$$

Where y is a matrix of observed values, the dependent variable. X is a matrix consisting of input variables, the independent variables. β represents the coefficient of the estimated degree of change in the outcome concerning the dependent variable. ϵ is called an error term, which calculates for the noise in data that are not captured but the rest of the equation. The power of regression lies in its simplicity. This simple model captures relationships and their magnitude between variables. Moreover, it does not require nor makes any assumption about the underlying data. It can also be used to determine the feature importance in a model. On the other hand, because of its simplicity, the relationship that regression can capture are not complex. Therefore, it assumes the linear relation between dependent and independent variables.

A.2 Decision Tree

Decision Tree (DT) is another simple yet powerful algorithm. The idea begins DT is to derive an approximate target function that will be represented in a decision tree form. Learned rules can be represented as a set of {if-else} rules to help the human reader understand the decisions that predictor takes [70].

The primary mechanism of DT is to split attributes so that rules are derived. There are many ways to measure the quality of the split; this research will use Gini Index. Gini index of variables is a probability of this variable being incorrectly classified when selected randomly, where the value of it varies from 0 to 1 [70]. The following equation calculates it:

$$GiniIndex = 1 - \sum_{i=1}^{n}(P_i)^2$$

A splitting attribute will be the smallest value of the Gini Index. The main advantage of the decision trees lies in their human-readable form. Moreover, the data for the classifier does not need to be perfect; a decision tree can deal with missing values. On the other hand, this tool is very sensitive to changes in the data, as well as the complexity of the decision trees can increase with the amount of data.

A.3 Multi-layer Perceptron

Multi-layer Perceptron (MLP) is a type of NN. The architecture of this network is simple; it consists of at least three layers, one input layer, one output layer, and at least one or more hidden layers. The input is passed through the network by taking the dot product with the weights on edges between layers. Then this is passed to nodes of a hidden layer. A weighted sum of all inputs is calculated, and if the minimum threshold value is reached, the neuron in the hidden layer will be activated. This process is repeated for each hidden layer until the output layer is reached. This single pass of instance in the network does not derive any information. In order to learn, MLP passes the values once more but backwards, and this process is called Backpropagation.

Along with this process, weights in the network are adjusted. The process of forwarding and backward pass of the data is repeated until there is nothing to update between input-output or such that the model has converged.

The main advantage of this classifier lies in the ability to capture complex, non-linear relations between the variables. Moreover, it also works with the large input data, as well as with the smaller datasets. On the other hand, a multi-layer perceptron is a fully connected and very complex network that requires much time to train.

Appendix B

Unused changes to the network

Initialization Function

Another change that was tried in this paper, is changing the initialization function. As the authors with the original paper indicate, the authors used Xavier initialization function to initialize the parameters of the generator randomly. As argued by the authors of the original paper used it to allow the model to reach deep into within. The original Xavier initialization function presents as follows in B.1, where n, is the data dimension

$$W = \mathcal{U} \sim [-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}] \tag{B.1}$$

The change version, where the data dimension is squared looks as follows in B.2:

$$W = \mathcal{U} \sim [-\frac{1}{\frac{\sqrt{n^2}}{2}}, \frac{1}{\frac{\sqrt{n^2}}{2}}] \tag{B.2}$$

Because of that change the function is able to generate narrower bounds for the values of the weights. As shown in the graph

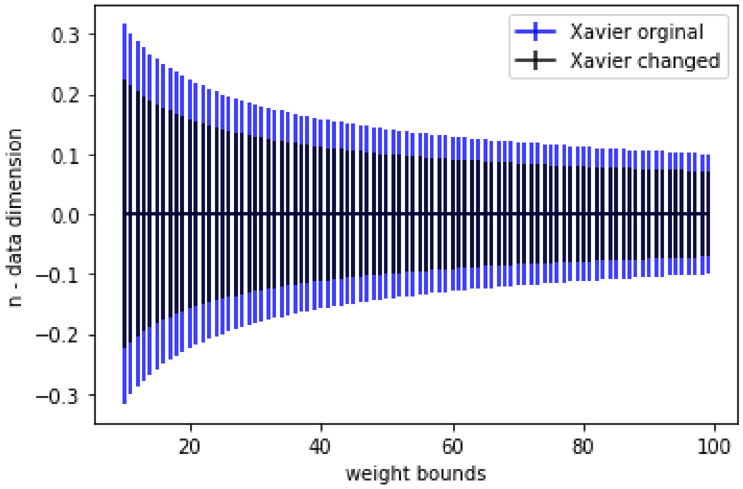

Figure B.1: : Xavier Initialization function bounds and variation of Xavier Initialization function bounds

And as well it will avoid a pitfall of becoming a value of 0 [102] to avoid poor performance.

Moreover, the comparison between 3 standard functions and modified version that is used in this papers, shows that the modified version still shows the smallest generated value, that is not a constant B.2. Unfortunately, all the experiments performed with this changed showed that the network is stable with small dimension datasets,

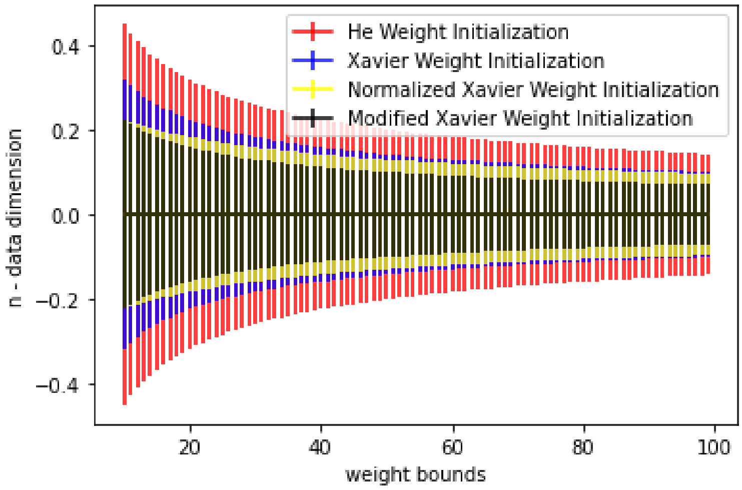

FIGURE B.2: :Initialization function bounds differences

whereas the datasets with higher dimension the accuracy dropped significantly below 30%.

Appendix C

Additional experiments

C.1 Pearson statistical correlation experiment

Another way to check the quality of the generated data is to compare it with the original one and run the Pearson correlation test, also known as Pairwise Correlation Difference. This experiment will be shown for all of the dataset and how this changed the association between the original dataset and the one generated by the Intag framework with modified Pate-GAN. Moreover, the bigger the association the better the synthetic data is.

German dataset

As shown in the Figures C.1,C.2,C.3 l for the synthetic data generated the association becomes stronger. Therefore it shows that the framework with modified Pate-GAN data generation it can replicate the properties of the original data.

Adult

As shown in Figures: C.4,C.5,C.6, the association between the figures shows that there was indeed a small increase in the association.

Compas

As shown in Figures: C.7,C.8,C.9, the association between the figures shows that there was indeed a small increase in the association, especially more some of variables.

Test I

As shown in Figures: C.10,C.11,C.12, the association between the figures shows that there was significant increase for some of the variables.

Test II

As shown in Figures: C.13,C.14,C.15, the association between the figures shows that there was indeed a small increase in the association.

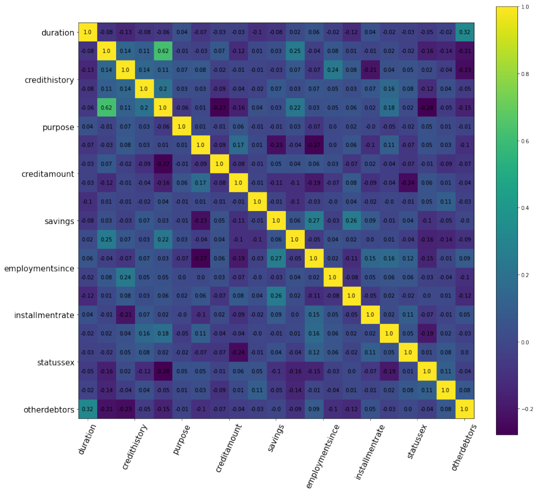

FIGURE C.1: German dataset correlation with the original data.

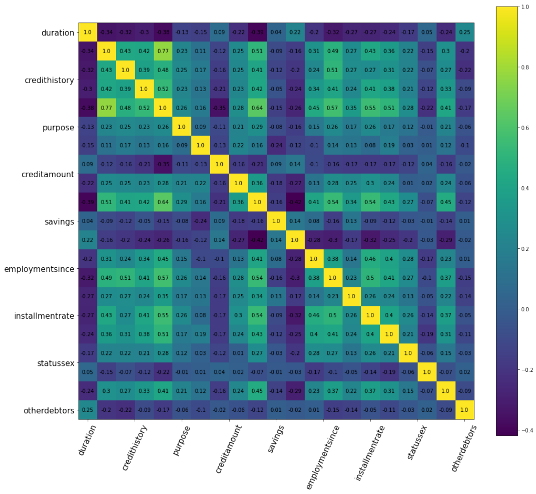

FIGURE C.2: Original dataset correlation with the synthetic data generated.

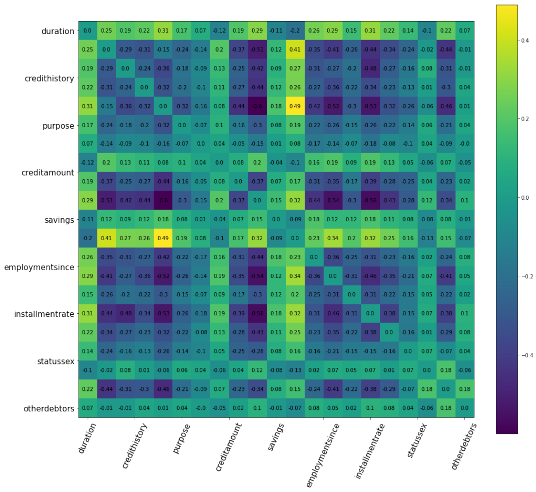

FIGURE C.3: Difference between the association in the original data and the original data with the synthetic data

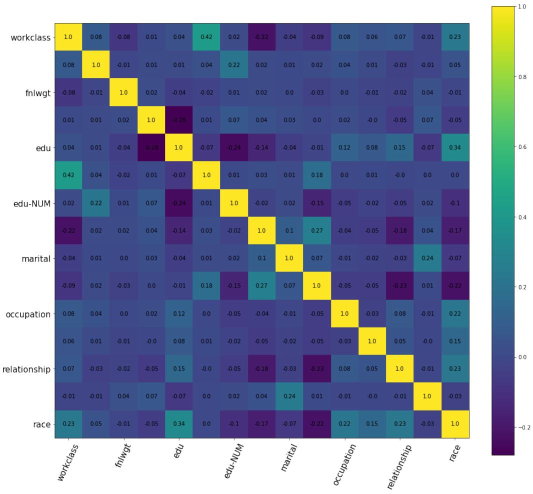

FIGURE C.4: Original ataset correlation with the original data.

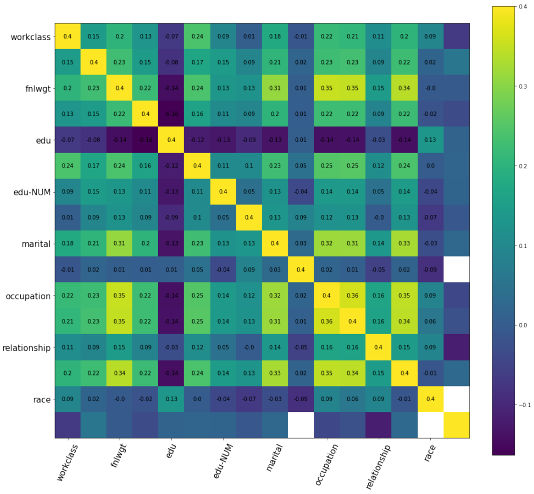

FIGURE C.5: Original dataset correlation with the synthetic data generated.

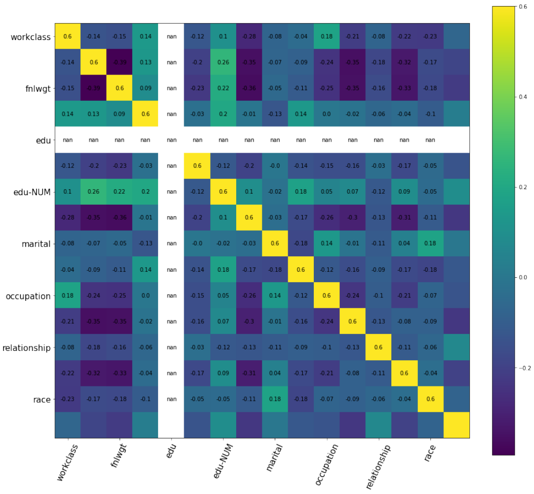

FIGURE C.6: Difference between the association in the original data and the original data with the synthetic data

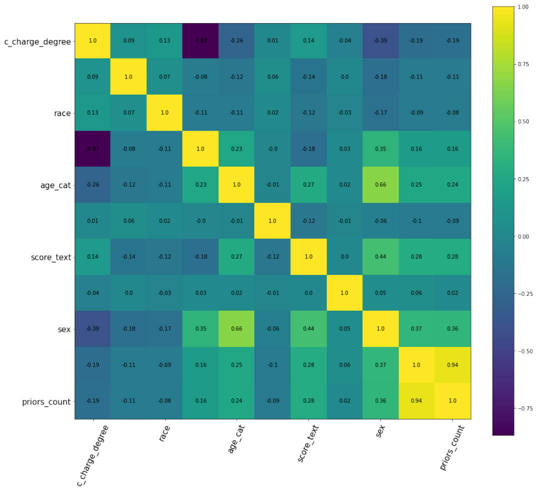

Figure C.7: Original ataset correlation with the original data.

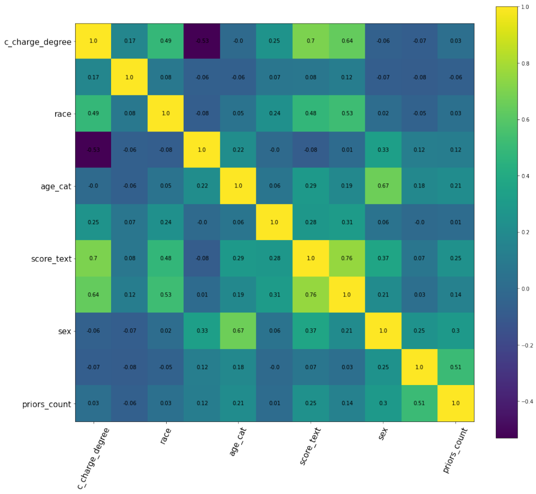

Figure C.8: Original dataset correlation with the synthetic data generated.

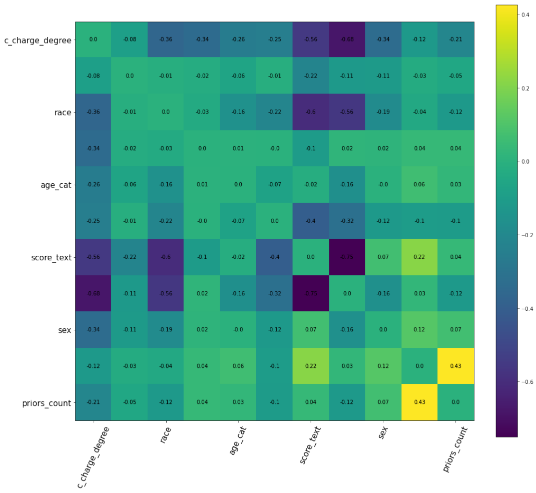

Figure C.9: Difference between the association in the original data and the original data with the synthetic data

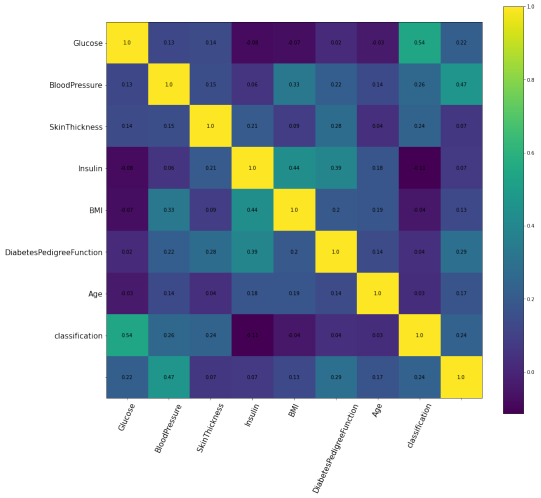

FIGURE C.10: Original ataset correlation with the original data.

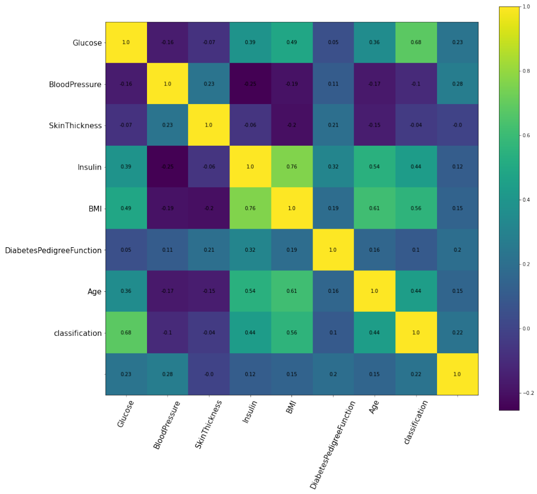

FIGURE C.11: Original dataset correlation with the synthetic data generated.

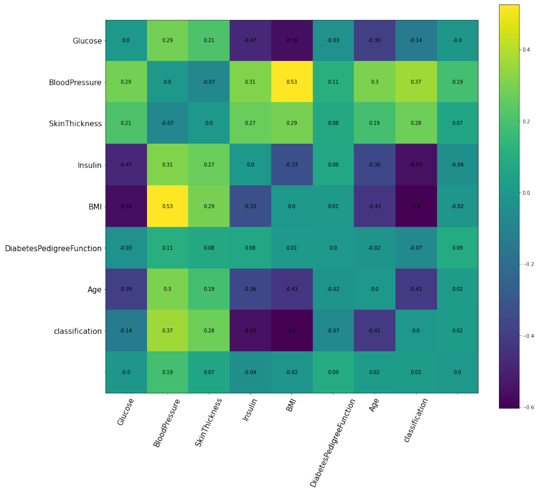

FIGURE C.12: Difference between the association in the original data and the original data with the synthetic data

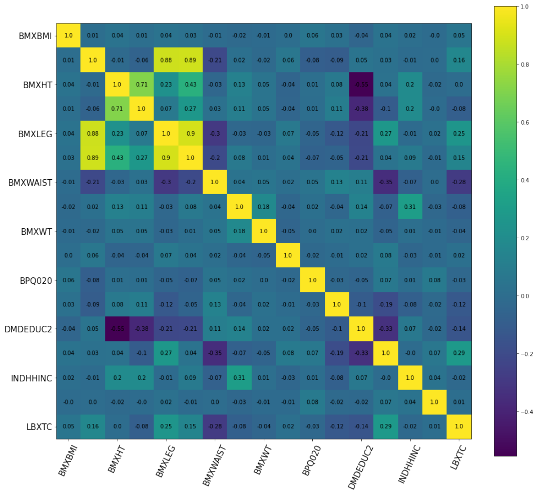

FIGURE C.13: Original ataset correlation with the original data.

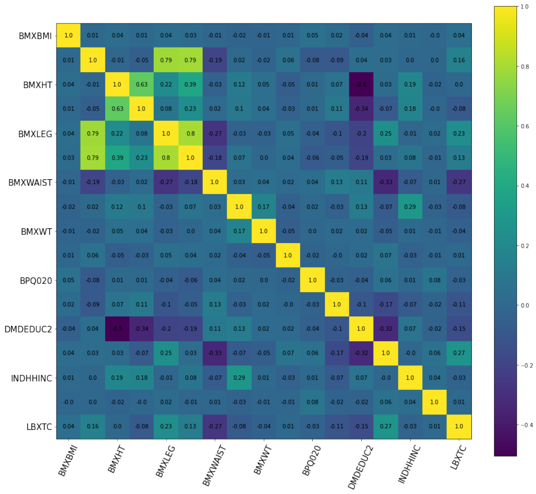

FIGURE C.14: Original dataset correlation with the synthetic data generated.

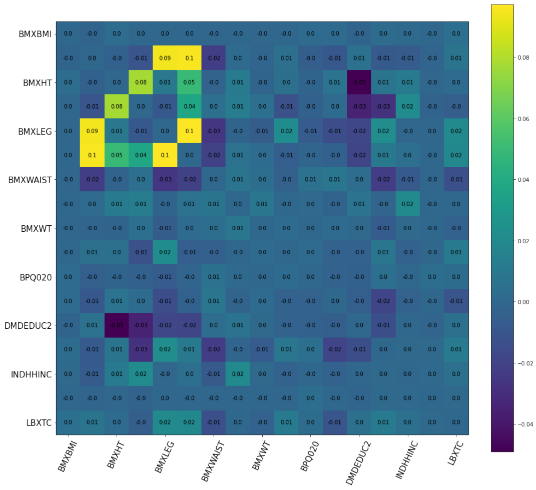

FIGURE C.15: Difference between the association in the original data and the original data with the synthetic data

Bibliography

[1] Rakesh Agrawal, Ramakrishnan Srikant, et al. "Fast algorithms for mining association rules". In: *Proc. 20th int. conf. very large data bases, VLDB*. Vol. 1215. 1994, pp. 487–499.

[2] Yongsu Ahn and Yu-Ru Lin. "Fairsight: Visual analytics for fairness in decision making". In: *IEEE transactions on visualization and computer graphics* 26.1 (2019), pp. 1086–1095.

[3] Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. "Applying support vector machines to imbalanced datasets". In: *European conference on machine learning*. Springer. 2004, pp. 39–50.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks". In: *International conference on machine learning*. PMLR. 2017, pp. 214–223.

[5] Niels Bantilan. "Themis-ml: A fairness-aware machine learning interface for end-to-end discrimination discovery and mitigation". In: *Journal of Technology in Human Services* 36.1 (2018), pp. 15–30.

[6] Brett K Beaulieu-Jones et al. "Privacy-preserving generative deep neural networks support clinical data sharing". In: *Circulation: Cardiovascular Quality and Outcomes* 12.7 (2019), e005122.

[7] Rachel KE Bellamy et al. "AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias". In: *arXiv preprint arXiv:1810.01943* (2018).

[8] Dan Berthiaume. "Amazon applies machine learning to fresh grocery experience". In: *Chain Store Age* (2021). URL: https://chainstoreage.com/amazon-applies-machine-learning-fresh-grocery-experience.

[9] Cigdem Beyan and Robert Fisher. "Classifying imbalanced data sets using similarity based hierarchical decomposition". In: *Pattern Recognition* 48.5 (2015), pp. 1653–1672.

[10] Sarah Bird et al. "Fairlearn: A toolkit for assessing and improving fairness in AI". In: *Microsoft, Tech. Rep. MSR-TR-2020-32* (2020).

[11] Jeffrey P Bradford et al. "Pruning decision trees with misclassification costs". In: *European Conference on Machine Learning*. Springer. 1998, pp. 131–136.

[12] Andrew W Brown, Kathryn A Kaiser, and David B Allison. "Issues with data and analyses: Errors, underlying themes, and potential solutions". In: *Proceedings of the National Academy of Sciences* 115.11 (2018), pp. 2563–2570.

[13] Michael Buckland and Fredric Gey. "The relationship between recall and precision". In: *Journal of the American society for information science* 45.1 (1994), pp. 12–19.

[14] Chumphol Bunkhumpornpat, Krung Sinapiromsaran, and Chidchanok Lursinsap. "DBSMOTE: density-based synthetic minority over-sampling technique". In: *Applied Intelligence* 36.3 (2012), pp. 664–684.

[15] Chumphol Bunkhumpornpat, Krung Sinapiromsaran, and Chidchanok Lursinsap. "Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem". In: *Pacific-Asia conference on knowledge discovery and data mining*. Springer. 2009, pp. 475–482.

[16] Ángel Alexander Cabrera et al. "FairVis: Visual analytics for discovering intersectional bias in machine learning". In: *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE. 2019, pp. 46–56.

[17] Cristiano L Castro and Antônio P Braga. "Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data". In: *IEEE transactions on neural networks and learning systems* 24.6 (2013), pp. 888–899.

[18] Nitesh V Chawla, Nathalie Japkowicz, and Aleksander Kotcz. "Special issue on learning from imbalanced data sets". In: *ACM SIGKDD explorations newsletter* 6.1 (2004), pp. 1–6.

[19] Nitesh V Chawla et al. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[20] Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. "Gs-wgan: A gradient-sanitized approach for learning differentially private generators". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 12673–12684.

[21] Edward Choi et al. "Generating multi-label discrete patient records using generative adversarial networks". In: *Machine learning for healthcare conference*. PMLR. 2017, pp. 286–305.

[22] Allen Chieng Hoon Choong and Nung Kion Lee. "Evaluation of convolutionary neural networks modeling of DNA sequences using ordinal versus one-hot encoding method". In: *2017 International Conference on Computer and Drone Applications (IConDA)*. 2017, pp. 60–65. DOI: 10.1109/ICONDA.2017.8270400.

[23] Kevin A Clarke. "The phantom menace: Omitted variable bias in econometric research". In: *Conflict management and peace science* 22.4 (2005), pp. 341–352.

[24] Sam Corbett-Davies et al. "A computer program used for bail and sentencing decisions was labeled biased against blacks. It's actually not that clear." In: *The Washington Post* (2016). URL: https://www.washingtonpost.com/news/monkey-cage/wp/2016/10/17/can-an-algorithm-be-racist-our-analysis-is-more-cautious-than-propublicas/.

[25] Sam Corbett-Davies et al. "Algorithmic decision making and the cost of fairness". In: *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*. 2017, pp. 797–806.

[26] Debashree Devi, Saroj K Biswas, and Biswajit Purkayastha. "A review on solution to class imbalance problem: Undersampling approaches". In: *2020 International Conference on Computational Performance Evaluation (ComPE)*. IEEE. 2020, pp. 626–631.

[27] Nikhil V Dhurandhar et al. "Energy balance measurement: when something is not better than nothing". In: *International journal of obesity* 39.7 (2015), pp. 1109–1113.

[28] Cynthia Dwork. "Differential privacy: A survey of results". In: *International conference on theory and applications of models of computation*. Springer. 2008, pp. 1–19.

[29] Cynthia Dwork et al. “Fairness through awareness”. In: *Proceedings of the 3rd innovations in theoretical computer science conference*. 2012, pp. 214–226.

[30] Wei Fan et al. “AdaCost: misclassification cost-sensitive boosting”. In: *Icml*. Vol. 99. Citeseer. 1999, pp. 97–105.

[31] Andrew Guthrie Ferguson. “Big data and predictive reasonable suspicion”. In: *U. Pa. L. Rev.* 163 (2014), p. 327.

[32] Alberto Fernández et al. *Learning from imbalanced data sets*. Vol. 10. Springer, 2018.

[33] Pratik Gajane and Mykola Pechenizkiy. “On formalizing fairness in prediction with machine learning”. In: *arXiv preprint arXiv:1710.03184* (2017).

[34] Honghao Gao et al. “An approach to data consistency checking for the dynamic replacement of service process”. In: *IEEE Access* 5 (2017), pp. 11700–11711.

[35] Salvador García and Francisco Herrera. “Evolutionary Undersampling for Classification with Imbalanced Datasets: Proposals and Taxonomy”. In: *Evol. Comput.* 17.3 (2009), 275–306. ISSN: 1063-6560.

[36] Andre Goncalves et al. “Generation and evaluation of synthetic patient data”. In: *BMC medical research methodology* 20.1 (2020), pp. 1–40.

[37] Ian Goodfellow et al. “Generative adversarial nets”. In: *Advances in neural information processing systems* 27 (2014).

[38] Ishaan Gulrajani et al. “Improved training of wasserstein gans”. In: *Advances in neural information processing systems* 30 (2017).

[39] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. “Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning”. In: *International conference on intelligent computing*. Springer. 2005, pp. 878–887.

[40] Alex Hanna et al. “Towards a critical race methodology in algorithmic fairness”. In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 2020, pp. 501–512.

[41] Eszter Hargittai. “Whose space? Differences among users and non-users of social network sites”. In: *Journal of computer-mediated communication* 13.1 (2007), pp. 276–297.

[42] Peter Hart. “The condensed nearest neighbor rule (corresp.)” In: *IEEE transactions on information theory* 14.3 (1968), pp. 515–516.

[43] Tawfiq Hasanin and Taghi Khoshgoftaar. “The effects of random undersampling with simulated class imbalance for big data”. In: *2018 IEEE International Conference on Information Reuse and Integration (IRI)*. IEEE. 2018, pp. 70–79.

[44] Haibo He et al. “ADASYN: Adaptive synthetic sampling approach for imbalanced learning”. In: *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. IEEE. 2008, pp. 1322–1328.

[45] Barbara Hoffman. “Employment discrimination: another hurdle for cancer survivors”. In: *Cancer investigation* 9.5 (1991), pp. 589–595.

[46] Joop J Hox and Hennie R Boeije. “Data collection, primary vs. secondary”. In: *Encyclopedia of social measurement* 1.1 (2005), pp. 593–599.

[47] Lucas Introna and Helen Nissenbaum. “Defining the web: The politics of search engines”. In: *Computer* 33.1 (2000), pp. 54–62.

[48] Nathalie Japkowicz. “Concept-learning in the presence of between-class and within-class imbalances”. In: *Conference of the Canadian society for computational studies of intelligence*. Springer. 2001, pp. 67–77.

[49] Nathalie Japkowicz and Shaju Stephen. “The class imbalance problem: A systematic study”. In: *Intelligent data analysis* 6.5 (2002), pp. 429–449.

[50] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. “PATE-GAN: Generating synthetic data with differential privacy guarantees”. In: *International conference on learning representations*. 2018.

[51] Jeff Larson Julia Angwin. *Machine bias*. 2016. URL: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

[52] Saurabh Karsoliya. “Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture”. In: *International Journal of Engineering Trends and Technology* 3.6 (2012), pp. 714–717.

[53] Niki Kilbertus et al. “Avoiding discrimination through causal reasoning”. In: *Advances in neural information processing systems* 30 (2017).

[54] Kerenaftali Klein, Stefanie Hennig, and Sanjoy Ketan Paul. “A bayesian modelling approach with balancing informative prior for analysing imbalanced data”. In: *Plos one* 11.4 (2016), e0152700.

[55] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. “Inherent trade-offs in the fair determination of risk scores”. In: *arXiv preprint arXiv:1609.05807* (2016).

[56] Naveen Kodali et al. “On Convergence and Stability of GANs”. In: *arXiv: Artificial Intelligence* (2018).

[57] Hendrik Kück. “Bayesian formulations of multiple instance learning with applications to general object recognition”. PhD thesis. Citeseer, 2004.

[58] Preethi Lahoti, Krishna P Gummadi, and Gerhard Weikum. “ifair: Learning individually fair data representations for algorithmic decision making”. In: *2019 ieee 35th international conference on data engineering (icde)*. IEEE. 2019, pp. 1334–1345.

[59] Der-Chiang Li et al. “Using mega-trend-diffusion and artificial samples in small data set learning for early flexible manufacturing system scheduling knowledge”. In: *Computers & Operations Research* 34.4 (2007), pp. 966–982.

[60] Haoran Li et al. “DPSynthesizer: differentially private data synthesizer for privacy preserving data sharing”. In: *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases*. Vol. 7. 13. NIH Public Access. 2014, p. 1677.

[61] Jianhua Lin. “Divergence measures based on the Shannon entropy”. In: *IEEE Transactions on Information theory* 37.1 (1991), pp. 145–151.

[62] Yang Liu et al. “Combining integrated sampling with SVM ensembles for learning from imbalanced datasets”. In: *Information Processing & Management* 47.4 (2011), pp. 617–631.

[63] David G Lowe. “Distinctive image features from scale-invariant keypoints”. In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[64] Abdul Majid et al. “Prediction of human breast and colon cancers from imbalanced data using nearest neighbor and support vector machines”. In: *Computer methods and programs in biomedicine* 113.3 (2014), pp. 792–808.

[65] Ninareh Mehrabi et al. “A Survey on Bias and Fairness in Machine Learning”. In: *ACM Comput. Surv.* 54.6 (2021). ISSN: 0360-0300. DOI: 10.1145/3457607. URL: https://doi.org/10.1145/3457607.

[66] Giovanna Menardi and Nicola Torelli. “Training and assessing classification rules with imbalanced data”. In: *Data mining and knowledge discovery* 28.1 (2014), pp. 92–122.

[67] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. “The numerics of gans”. In: *Advances in neural information processing systems* 30 (2017).

[68] Ibomoiye Domor Mienye and Yanxia Sun. “Performance analysis of cost-sensitive learning methods with application to imbalanced medical data”. In: *Informatics in Medicine Unlocked* 25 (2021), p. 100690.

[69] Satwik Mishra. “Handling imbalanced data: SMOTE vs. random undersampling”. In: *Int. Res. J. Eng. Technol* 4.8 (2017), pp. 317–320.

[70] Tom Mitchell. “Machine learning”. In: (1997).

[71] Giang Hoang Nguyen, Abdesselam Bouzerdoum, and Son Lam Phung. “Learning pattern classification tasks with imbalanced data sets”. In: *Pattern recognition* (2009), pp. 193–208.

[72] Sourabh Niyogi. “Bayesian learning at the syntax-semantics interface”. In: *Proceedings of the Twenty-Fourth Annual Conference of the Cognitive Science Society*. Routledge. 2019, pp. 697–702.

[73] P Russel Norvig and S Artificial Intelligence. *A modern approach*. Prentice Hall Upper Saddle River, NJ, USA: 2002, p. 739.

[74] Nir Ofek et al. “Fast-CBUS: A fast clustering-based undersampling method for addressing the class imbalance problem”. In: *Neurocomputing* 243 (2017), pp. 88–102.

[75] Allen ONeill. “Data Quality Evaluation using Probability Models”. In: *arXiv preprint arXiv:2009.06672* (2020).

[76] Nicolas Papernot et al. “Semi-supervised knowledge transfer for deep learning from private training data”. In: *arXiv preprint arXiv:1610.05755* (2016).

[77] Noseong Park et al. “Data Synthesis Based on Generative Adversarial Networks”. In: *Proc. VLDB Endow.* 11 (2018), 1071–1083. ISSN: 2150-8097.

[78] Yubin Park and Joydeep Ghosh. “PeGS: Perturbed Gibbs Samplers that Generate Privacy-Compliant Synthetic Data.” In: *Trans. Data Priv.* 7.3 (2014), pp. 253–282.

[79] Judea Pearl. “Causal inference in statistics: An overview”. In: *Statistics surveys* 3 (2009), pp. 96–146.

[80] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988, pp. 42–50.

[81] J. Ross Quinlan. “Improved estimates for the accuracy of small disjuncts”. In: *Machine Learning* 6.1 (1991), pp. 93–98.

[82] M Mostafizur Rahman and D Davis. “Cluster based under-sampling for unbalanced cardiovascular data”. In: *Proceedings of the World Congress on Engineering*. Vol. 3. 2013, pp. 3–5.

[83] Amirarsalan Rajabi and Ozlem Ozmen Garibay. “TabFairGAN: Fair Tabular Data Generation with Generative Adversarial Networks”. In: *arXiv preprint arXiv:2109.00666* (2021).

[84] Aaditya Ramdas, Nicolás García Trillos, and Marco Cuturi. "On wasserstein two-sample testing and related families of nonparametric tests". In: *Entropy* 19.2 (2017), p. 47.

[85] Farshid Rayhan et al. "Cusboost: cluster-based under-sampling with boosting for imbalanced classification". In: *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*. IEEE. 2017, pp. 1–5.

[86] Baeza-Yates Ricardo. ""Bias on the Web". In: *Communications of the ACM* 61.6 (2018), pp. 54–61.

[87] Andrea Romei and Salvatore Ruggieri. "A multidisciplinary survey on discrimination analysis". In: (2013).

[88] Takaya Saito and Marc Rehmsmeier. "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets". In: *PloS one* 10.3 (2015), e0118432.

[89] Pedro Saleiro et al. "Aequitas: A bias and fairness audit toolkit". In: *arXiv preprint arXiv:1811.05577* (2018).

[90] Tim Salimans et al. "Improved techniques for training gans". In: *Advances in neural information processing systems* 29 (2016).

[91] Samira Samadi et al. "The price of fair pca: One extra dimension". In: *Advances in neural information processing systems* 31 (2018).

[92] Tobias Schnabel et al. "Recommendations as treatments: Debiasing learning and evaluation". In: *international conference on machine learning*. PMLR. 2016, pp. 1670–1679.

[93] Max Schubach et al. "Imbalance-aware machine learning for predicting rare and common disease-associated non-coding variants". In: *Scientific reports* 7.1 (2017), pp. 1–12.

[94] John Semerdjian and Spencer Frank. "An ensemble classifier for predicting the onset of type II diabetes". In: *arXiv preprint arXiv:1708.07480* (2017).

[95] Camelia Simoiu, Sam Corbett-Davies, and Sharad Goel. "The problem of infra-marginality in outcome tests for discrimination". In: *The Annals of Applied Statistics* 11.3 (2017), pp. 1193–1216.

[96] Jack W Smith et al. "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus". In: *Proceedings of the annual symposium on computer application in medical care*. American Medical Informatics Association. 1988, p. 261.

[97] Paria Soltanzadeh and Mahdi Hashemzadeh. "RCSMOTE: range-controlled synthetic minority over-sampling technique for handling the class imbalance problem". In: *Information Sciences* 542 (2021), pp. 92–111.

[98] Harini Suresh and John Guttag. "A Framework for Understanding Sources of Harm throughout the Machine Learning Life Cycle". In: *Equity and Access in Algorithms, Mechanisms, and Optimization*. New York, NY, USA: Association for Computing Machinery, 2021. ISBN: 9781450385534. DOI: 10.1145/3465416.3483305. URL: https://doi.org/10.1145/3465416.3483305.

[99] John A Swets. "Measuring the accuracy of diagnostic systems". In: *Science* 240.4857 (1988), pp. 1285–1293.

[100] Muhammad Atif Tahir, Josef Kittler, and Fei Yan. "Inverse random under sampling for class imbalance problem and its application to multi-label classification". In: *Pattern Recognition* 45.10 (2012), pp. 3738–3750.

[101] Muhammad Atif Tahir et al. "A multiple expert approach to the class imbalance problem using inverse random under sampling". In: *International workshop on multiple classifier systems*. Springer. 2009, pp. 82–91.

[102] Georg Thimm and Emile Fiesler. "High-order and multilayer perceptron initialization". In: *IEEE Transactions on Neural Networks* 8.2 (1997), pp. 349–359.

[103] Robert J Tibshirani and Bradley Efron. "An introduction to the bootstrap". In: *Monographs on statistics and applied probability* 57 (1993), pp. 1–436.

[104] Ivan Tomek. "Two modifications of CNN". In: *IEEE Trans. Systems, Man and Cybernetics* 6 (1976), pp. 769–772.

[105] Sahil Verma and Julia Rubin. "Fairness definitions explained". In: *2018 ieee/acm international workshop on software fairness (fairware)*. IEEE. 2018, pp. 1–7.

[106] Qianwen Wang et al. "Visual analysis of discrimination in machine learning". In: *IEEE Transactions on Visualization and Computer Graphics* 27.2 (2020), pp. 1470–1480.

[107] Xiaokang Wang, Huiwen Wang, and Yihui Wang. "A density weighted fuzzy outlier clustering approach for class imbalanced learning". In: *Neural Computing and Applications* 32.16 (2020), pp. 13035–13049.

[108] Xiaokang Wang et al. "A Fuzzy Consensus Clustering Based Undersampling Approach for Class Imbalanced Learning". In: *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*. 2019, pp. 133–137.

[109] James Wexler et al. "The what-if tool: Interactive probing of machine learning models". In: *IEEE transactions on visualization and computer graphics* 26.1 (2019), pp. 56–65.

[110] Liyang Xie et al. "Differentially private generative adversarial network". In: *arXiv preprint arXiv:1802.06739* (2018).

[111] Catherina Xu et al. "Fairness Indicators Demo: Scalable Infrastructure for Fair ML Systems". In: (2020).

[112] Le Xu and Mo-Yuen Chow. "A classification approach for power distribution systems fault cause identification". In: *IEEE Transactions on Power Systems* 21.1 (2006), pp. 53–60.

[113] Lei Xu et al. "Modeling Tabular Data Using Conditional GAN". In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2019, 7335—7345.

[114] Show-Jane Yen and Yue-Shi Lee. "Cluster-based under-sampling approaches for imbalanced data distributions". In: *Expert Systems with Applications* 36.3 (2009), pp. 5718–5727.

[115] Xiaoxin Yin and Jiawei Han. "CPAR: Classification based on predictive association rules". In: *Proceedings of the 2003 SIAM international conference on data mining*. SIAM. 2003, pp. 331–335.

[116] Lian Yu and Nengfeng Zhou. "Survey of Imbalanced Data Methodologies". In: *arXiv preprint arXiv:2104.02240* (2021).

[117] Bianca Zadrozny and Charles Elkan. “Learning and making decisions when costs and probabilities are both unknown”. In: *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining.* 2001, pp. 204–213.

[118] Du Zhang. “Inconsistencies in big data”. In: *2013 IEEE 12th International Conference on Cognitive Informatics and Cognitive Computing.* IEEE. 2013, pp. 61–67.

[119] Indre Zliobaite. “On the relation between accuracy and fairness in binary classification”. In: *arXiv preprint arXiv:1505.05723* (2015).