JHERONIMUS ACADEMY OF DATA SCIENCE (JADS)

MASTER THESIS

Generation of Synthetic Health Data with Sequential Treatments: a Case Study on Colorectal Cancer Patients Data in the Netherlands Cancer Registry

Author:
Myrthe WOUTERS
Student number:
1273195

Main Supervisor:
Dr. Murat FIRAT
Second Supervisor:
Dr. Ir. Joaquin VANSCHOREN

Company Supervisor:
Daan KNOORS

A thesis submitted in fulfillment of the requirements
for the degree of Master of Science

Master Data Science & Entrepreneurship

Eindhoven University of Technology (TU/e)
&
Tilburg University (TiU)

July 1, 2021

JHERONIMUS ACADEMY OF DATA SCIENCE (JADS)

Abstract

Eindhoven University of Technology (TU/e)
Tilburg University (TiU)

Master of Science

Generation of Synthetic Health Data with Sequential Treatments: a Case Study on Colorectal Cancer Patients Data in the Netherlands Cancer Registry

by Myrthe WOUTERS

Sharing medical and patient data with a larger (research) community is beneficial to further stimulate and accelerate advances in medical knowledge and patient care. However, medical patient data is often highly sensitive and private, making it impossible to distribute the data to a wider audience. The recent literature proposes the generation of synthetic medical data that mimics the statistical properties of the original data to overcome these privacy concerns. This thesis focuses on the generation and evaluation of a type of data largely unexplored in the existing literature: health event data. More specifically, we focus on the specific application of cancer patient data, including both static patient covariates and treatment sequences, in the Netherlands Cancer Registry maintained by the Netherlands Comprehensive Cancer Organization. Based on a literature review, we adapted or directly applied two existing generative methods to our use case. We find that both generative methods are able to generate high-quality synthetic data by considerably outperforming the naive baseline on a set of seven quality metrics. We defined these seven (existing) metrics to evaluate the quality of the specific type of health event data from several aspects and confirm their validity. With these metrics, we aim to motivate and enable healthcare organizations to participate in the synthetic health data movement while ensuring and evaluating the quality of the generated data. In a follow-up experiment, we implement the differentially private version of one of the selected generative methods. The results show that - using additional techniques including generalization and post-processing - we can obtain a differentially private synthetic data set with reasonable quality for a privacy budget generally considered just acceptable. For stricter privacy guarantees, characteristics and statistical properties of the original data are lost in the synthetic data. We encourage future work to further investigate the privacy aspect of synthetic health event data.

Acknowledgements

I would like to express my gratitude to all people that made the completion of my master thesis possible. Here, I would like to take the opportunity to thank some people in special.

First and foremost, I would like to thank my main supervisor Murat Firat and company supervisor Daan Knoors, without whom my research would have been impossible. Murat, for always taking the time to give critical feedback and advice during our discussion sessions, which most certainly improved the quality of my research, while at the same time reassuring me that we were taking steps into the right direction. Daan, not only for having your "online doors" open at any time to discuss relevant aspects of my thesis, giving me the freedom, trust and steering where necessary in defining the research, but also for encouraging me to acknowledge my accomplishments along the way. I am very grateful for both supervisors being this actively involved in my research, where our weekly meetings provided me with useful insights during the process. Furthermore, I would like to thank Joaquin Vanschoren for being the second academic assessor of this thesis.

Besides my supervisors, I would also like to thank my colleagues from both the data science and software development teams at IKNL. Our weekly update meetings encouraged me to actively present all the work I had completed so far and existing challenges that lied ahead. Other than that, joining some of you at the office at least once a week made the experience a lot more enjoyable.

Then, my thanks go to my fellow JADS students, where Lieske Trommelen, Bernard Wezeman, Jules Huisman and Paulo Rijnberg deserve special mention. We have worked together on most projects during the master program and without you this entire journey would have been a lot less fun. Of course, I should not forget to mention all the table football games we played together before the COVID-19 lockdown.

Last, but definitely not least, I would like to thank my friends and family. First and foremost, my parents, Paul and Anne-Marie, for their unconditional support, faith and love, and for giving me the opportunity to study and develop myself. While I have moved out at the age of 18, your place in Schijndel still feels like home. My brother Luuk, for playing all kinds of different (sports) games together to clear my mind. My grandmother, for always lighting a candle whenever I needed it. A special thanks goes out to my boyfriend, Maurits, who supports me with anything I do, despite our very different thoughts on where to set the bar for myself. Thank you for comforting me at times, but also for all the enjoyable things we do together. Finally, my friends, for both the support and fun times during this past year and making me forget about my thesis every once in a while.

Myrthe Wouters

Tilburg, July 1, 2021

Contents

List of Figures

List of Tables

List of Abbreviations

AUC-ROC	**A**rea **U**nder the **C**urve - **R**eceiver **O**perating **C**haracteristic
DG	**D**oppel**G**anger
DG-NP	**D**oppel**G**anger - **N**o **P**rivacy
DP	**D**ifferential **P**rivacy
GAN	**G**enerative **A**dversarial **N**etwork
IKNL	**I**ntegraal **K**ankercentrum **N**eder**L**and (Netherlands Comprehensive Cancer Organization)
JS	**J**ensen **S**hannon
MS	**M**arginal **S**ynthesizer
MS-NP	**M**arginal **S**ynthesizer - **N**o **P**rivacy
NCR	**N**etherlands **C**ancer **R**egistry
NP	**N**o **P**rivacy
PB	**P**riv**B**ayes
PBS	**P**riv**B**ayes **S**equential
PBS-NP	**P**riv**B**ayes **S**equential - **N**o **P**rivacy
PBS-DP	**P**riv**B**ayes - **D**ifferential **P**rivacy
PCD	**P**airwise **C**orrelation **D**ifference
RMSE	**R**oot **M**ean **S**quare **E**rror
TB-TOH	**T**rain on **B**oth, **T**est on **O**riginal **H**oldout set

Chapter 1

Introduction

1.1 Research motivation

Progressively, large amounts of clinical patient data are electronically collected by healthcare organizations in the form of Electronic Health Records (EHR). However, the availability of these clinical records does not automatically lead to access to clinical data for researchers, due to privacy concerns caused by the presence of personally identifiable information in combination with sensitive medical data. Even if researchers are able to access the medical data through application procedures, these procedures are lengthy processes with strict legal requirements. As a result of this limited access, advances in biomedical knowledge and patient care are relatively slow compared to other industries (Nass et al., 2009).

When real medical data is not available, data is often anonymized by healthcare organizations using de-identification methods. However, studies have shown that this approach is vulnerable to privacy attacks and re-identification risks (El Emam et al., 2011). Therefore, an alternative approach to de-identification is proposed: generation of *synthetic* data. The advantage of using synthetic data is that the data is artificially created, meaning that there is no direct one-to-one mapping between original and synthetic data. Therefore, in theory, synthetic data is more resistant to re-identification compared to de-identified data (Baowaly et al., 2019).

Prior research has investigated several synthetic data generation algorithms in the medical domain, on various data modalities, each coming with its own challenges. The majority of prior research has focused on cross-sectional EHR data, in which there is one row per patient including demographic or disease-specific variables (Camino et al., 2018; Choi et al., 2017; Yale et al., 2020), or regularly sampled time series data (Esteban et al., 2017). With regards to the cross-sectional EHR data, this data is not fully representative of real medical records, as in real life patient data is longitudinal in nature, and partly consists of (treatment) events through time. These events are generally influenced by patient covariates such as age group and gender. While there exist seminal methods to model time series data, it becomes increasingly complex to extend these methods to medical event time series data combined with patient covariates. To date, the generation of event-based time series health data remains largely unexplored in the literature (Dash et al., 2019). In this thesis, we aim to take the first steps in closing this gap by exploring and evaluating applicable methods for medical events data, in a case study on treatment sequences of colorectal cancer patients.

1.2 Research questions

This study is centered around the Netherlands Comprehensive Cancer Organization (IKNL). IKNL collects sensitive information of cancer patients in the Netherlands Cancer Registry (NCR). Data in the NCR is actively collected and maintained by IKNL and includes clinical data on nearly all cancer incidences in the Netherlands since 1989. The NCR supports health care professionals, academic researchers, patients, and policymakers with valuable data for their research and clinical guidelines. However, again, the NCR contains sensitive, medical data and therefore data sharing is limited. Currently, data can only be shared with external parties after the submission of a data request and approval by a supervisory committee. The aim of IKNL is to enable safe data sharing by developing algorithms that can generate synthetic data sets that mimic the properties of the real data, while at the same time protect the privacy of cancer patients.

As indicated in Section 1.1, prior research has investigated the generation of synthetic cross-sectional (patient) data. As such, IKNL has implemented several existing methods proposed in the literature for the generation of the cross-sectional cancer patient data in the NCR (e.g., PrivBayes (Zhang et al., 2017) and MWEM (Hardt et al., 2012)). Nonetheless, the NCR also contains *sequential* data on the treatments received by cancer patients. What remains largely uninvestigated - both in theory and in practice - is the generation of event-based time series data such as treatment sequences of cancer patients. Therefore, the focal aim of this thesis will be on the generation of such event-based time series data, in the specific context of treatment sequences of colorectal cancer patients in the NCR. More specifically, the main research question is:

To what extent can we generate synthetic health data with sequential treatments that resembles the statistical properties of the original data, in a case study of colorectal cancer patients in the NCR maintained by IKNL?

In order to structurally answer the main research question, we defined several sub-questions. First of all, we will investigate what generative methods are proposed in the academic literature and/or implemented at IKNL. Importantly, we will assess the applicability or needed adaptations for these models to be able to deal with the generation of sequential treatment events. Moreover, as health events are generally dependent upon patient covariates (Dash et al., 2019), we will focus on the applicability of existing methods for the *co-generation* of synthetic treatment sequences and static patient covariates. For this purpose, the first sub-question is defined:

1. What generative methods are proposed in the academic literature that can be used for the generation of synthetic health data with sequential treatments?

Given the generative - as opposed to discriminative - nature of synthetic data generation algorithms, evaluation of model output is a non-trivial task. Therefore, quality evaluation techniques for synthetic health data are needed to validate the generative model obtained, and subsequently substantiate its generated output. The synthetic data output should accurately capture the properties of the original data. Several techniques for evaluating the quality of synthetic data are presented in the academic literature. However, to our knowledge, research on both the generation of synthetic health event data as well as evaluating its quality is limited in the literature. The second sub-question will focus on the quality evaluation of synthetic health data with sequential treatments and is formulated as follows:

2. What techniques exist for evaluating the quality of synthetic data and what (new) approaches need to be incorporated for a quality evaluation of synthetic health data with sequential treatments?

As indicated, the generated data should accurately capture the (statistical) properties of entities and their relations in the original data. At the same time, the motivation for creating synthetic medical data is to preserve the privacy of individuals. Hence, we should ensure that the synthetic data is not too similar to the original data, in the sense that the data should not disclose private information on individuals in the original data set. There is a general consensus that incorporating privacy constraints and maintaining the quality of synthetic data might be two conflicting goals, commonly referred to as the privacy-utility trade-off (Jordon et al., 2019; Xie et al., 2018). Therefore, we - due to time limitations, shortly - cover the final sub-question related to privacy. In particular, the third sub-question will focus on existing ways for incorporating or evaluating the privacy of the synthetic data and is specified as:

3. What techniques or standards exist for evaluating the privacy or providing privacy guarantees of the synthetic health data with sequential treatments?

1.3 Scope

Given the limited prior research on synthetic health event data, our research is exploratory in nature. That is, we aim to provide a starting point on both the generation and the quality and privacy assessments of synthetic health event data. We do so in a case study on treatment sequences along with static patient data of colorectal cancer patients in the NCR. It is important to note that the generative methods and metrics we put forth are not all-encompassing. Future work may further develop (one of) the generative algorithms we propose, or even consider other algorithms and metrics. Moreover, our application is rather specific, and future research may apply and validate the proposed methods and metrics on other types of data sets.

1.4 Contributions

Here, we will state the general contributions of this thesis. The contributions are divided into theoretical and practical contributions.

1.4.1 Theoretical relevance

The theoretical contribution of this thesis is twofold. First of all, we propose and test both existing evaluation metrics, as well as a new metric to measure the sequential aspect of the treatment sequences, in the context of health event data. Secondly, we extend the non-private version of an existing generative method for tabular data, PrivBayes (Zhang et al., 2017), in order for the method to be able to generate sequential health event data.

1.4.2 Practical relevance

The practical contribution of this thesis is to implement the proposed extension of PrivBayes and two other generative methods for health data with sequential events. Then, we evaluate and compare their performances by obtaining a set of quality metrics, specified to the application at hand, for each generative method.

This research has several implications for practitioners. The origin of the research problem is the inability to share medical data with the broad (research) community due to privacy concerns. Currently, application procedures are lengthy and limited (Nass et al., 2009). With our research, we dive deeper into the generation of a specific type of health data: health event data. We aim to motivate parties in the cancer or general health domain to increasingly share privacy-sensitive medical (event) data by providing both generative models for the generation of health data with sequential events, as well as quality evaluation metrics. These quality evaluation metrics display to what extent the synthetic data can be used for data analytics and research purposes, from different aspects. From a practical perspective, multiple motives for generating high-quality synthetic data exist, including increased availability of medical data for research, more efficient communication on data set content to external parties, and safer algorithm development and hypothesis testing on synthetic data. In general, in order to use data for scientific research, Wilkinson et al. (2016) have designed a concise and measurable set of principles referred to as the FAIR Data Principles. IKNL, as well as other (scientific) health organizations, aim to adhere to these principles. FAIR stands for **F**indable, **A**ccessible, **F**indable, **I**nteroperable and **R**e-usable data. Synthetic data specifically contributes to increasing the accessibility and re-usability of (health) data for scientific research.

1.5 Outline

The outline of this thesis is as follows. Chapter 2 discusses related work and theoretical concepts in the field of synthetic (health) data generation. Chapter 3 details the use case and data set included in this research. In Chapter 4, the methodology regarding generative methods and quality metrics and the experimental setup are discussed. Thereafter, Chapter 5 presents the results of our experiments. Finally, we conclude our research in Chapter 6, in which we will also discuss limitations and suggestions for further research.

Chapter 2

Literature Review

In this chapter, we discuss prior work and theoretical concepts related to synthetic health data generation. The chapter is divided into three relevant aspects. Section 2.1 focuses on existing synthetic data generation algorithms and their applicability for our current use case. Subsequently, Section 2.2 provides a detailed explanation of two existing methods used in this research, as well as a short description of a baseline method. Then, Section 2.3 discusses prior work and metrics regarding the quality evaluation of synthetic health data. Finally, Section 2.4 covers privacy concepts in the synthetic data generation literature.

2.1 Synthetic data generation algorithms

This section presents an overview of synthetic data generation algorithms, with a focus on synthetic data generation methods within the health context. Nonetheless, some more generic methods or methods outside of the health context, yet potentially interesting for the use case of health event data, are discussed. The overview is divided into three subsections: algorithms for cross-sectional data (or static) data, algorithms for sequential (or longitudinal) data, and a concluding paragraph on the suitability of these algorithms with respect to the use case at hand.

2.1.1 Synthetic data generation algorithms for cross-sectional data

Several algorithms - varying in nature - have been proposed in the literature for the generation of cross-sectional synthetic health data. Goncalves et al. (2020) investigate three classes of synthetic data generation approaches for medical data: probabilistic models, classification-based imputation models, and generative adversarial networks (GANs). Their study is limited to categorical health data. An example of probabilistic models is Bayesian networks. In the academic literature, several variants of Bayesian networks are presented for the generation of synthetic data. However, one work - outside of the scope of medical data - stands out: PrivBayes (Zhang et al., 2017). The proposed PrivBayes method is a (differentially) private method that approximates a joint distribution using a first-order dependence tree (Bayesian network) for the generation of synthetic data. PrivBayes has also been implemented by IKNL for the generation of synthetic, cross-sectional patient data. Furthermore, Kaur et al. (2020) apply Bayesian networks for synthetic data generation in the context of health data. Another example of probabilistic models is Gaussian processes as presented in the Synthetic Data Vault (Patki et al., 2016). Their Gaussian Copula model is based on copula functions that describe the joint distribution of multiple random variables by analyzing the dependencies between their marginal distributions. From these learned dependencies, synthetic data can be generated. Second, an example of classification-based imputation models is Multivariate Imputation by

Chained Equations (MICE) (Azur et al., 2011). Finally, GANs are a class of neural networks, consisting of a generator that aims to produce realistic, but fake, data, and a discriminator that aims to categorize data as either real or fake. By playing an adversarial minimax game against each other, the generator can grasp the real data distribution. Goncalves et al. (2020) focus on a specific GAN: MC-MedGAN (Camino et al., 2018). MC-MedGAN is a generative adversarial network that also incorporates an autoencoder for the generation of multi-categorical synthetic data.

Other GAN-based approaches focusing on different or multiple data types are put forward in the literature. The first work to propose a GAN in the context of synthetic health data generation is the work of Choi et al. (2017). The authors present medGAN for the generation of synthetic static representations of diagnoses, medications, and procedure codes. The framework incorporates an autoencoder and is able to handle binary and count variables. The previously mentioned MC-MedGAN is an extension of medGAN that allows it to generate multi-categorical, cross-sectional data. Moreover, Baowaly et al. (2019) have improved medGAN for binary and count data by incorporating state-of-the-art techniques for enhanced GAN training into the framework. Finally, Zhang et al. (2020b) argue against incorporating an autoencoder and show that their proposed EMR-CWGAN outperforms both medGAN (Choi et al., 2017) as well as its two proposed adaptations medWGAN and medBGAN (Baowaly et al., 2019) on the generation of static, binary representations of diagnoses codes.

Next to GAN-based approaches focusing on a specific data type (e.g., binary, count, or categorical data), several approaches that handle cross-sectional data sets with mixed data types are proposed in the literature. While not evaluated on health data specifically, CTGAN (Xu et al., 2019) is a state-of-the-art GAN-based model when it comes to the generation of mixed-type (discrete and continuous) cross-sectional data. CTGAN uses a conditional generator to address several challenges related to mixed tabular data, including non-Gaussian, multimodal distributions of continuous columns and highly imbalanced, sparse one-hot encoded vectors of categorical columns. Other GAN-based approaches for mixed-type cross-sectional data exist, that - contrary to CTGAN - have design tailed to specific health data at hand or are evaluated on health data in particular. Examples include HealthGAN, which combines design ideas from medGAN with other GAN techniques to accommodate multiple data types (Yale et al., 2020), and HGAN (Yan et al., 2020), which highly builds on the architecture of the previously discussed EMR-CWGAN (Zhang et al., 2020b) to generate both binary and continuous data.

2.1.2 Synthetic data generation algorithms for sequential data

Sequential data is a distinct data modality in the synthetic data generation field, in the sense that a good generative model for sequential data should preserve temporal dynamics. That is, synthetic sequences should take into consideration the relationships between variables across time. The dominant technique used in synthetic data generation algorithms for longitudinal or sequential data is deep learning based models, including GANs. Nonetheless, some exceptions focusing on statistical models exist.

In the domain of statistical modeling procedures, Dahmen and Cook (2019) propose the use of hidden Markov Models for the generation of synthetic smart home sensor data. Here, the focus is on generating a nested sequence of events, without the inclusion of static covariates. Second, a recent study applied the PrivBayes method mentioned in Section 2.1.1, to a small data set of diabetes patient covariates and two repeated glycated hemoglobin measurements (Perkonoja, 2020). However, the repeated continuous measurements included differ considerably from the health event data considered in this study.

The first work on GANs for sequential data in the health domain is the work of Esteban et al. (2017). The authors present two models, Recurrent GAN (RGAN) and Recurrent Conditional GAN (RCGAN) for the generation of sequences of real-valued data - subject to some conditional input label in the case of RCGAN. RGAN and RCGAN focus on continuous, regularly time-stamped medical intensive care unit data, in contrast to health event data that is characterized by discrete, irregularly sampled time series data.

Similar in design to RCGAN, Yoon et al. (2019) propose TimeGAN for the generation of sequential data. Different from RCGAN, TimeGAN can generate both continuous and discrete as well as regularly sampled and irregularly sampled time series data. Yoon et al. (2019) show that TimeGAN consistently outperforms RCGAN on a diversity of data sets with different characteristics. The authors also evaluate TimeGAN on a data set on health event data, in particular, a lung cancer pathways data set consisting of sequences of events and their times, and conclude that TimeGAN can generate realistic lung cancer pathway sequences.

Where RCGAN and TimeGAN are both capable of producing sequential data, the methods do not co-generate covariates with sequential data, or their accuracy for such tasks is unknown (Lin et al., 2020). The dominant approach in the literature is to train variants of GANs called conditional GANs - as RCGAN - which learn to generate data conditioned on a user-provided input label. Generating the attributes as well is argued to be a simple extension (Esteban et al., 2017). Nonetheless, none of the prior works have shown evidence for this. TimeGAN claims to co-generate covariates in their paper, but it does not assess the model on any data set that includes covariates, nor does the released code handle covariates (Lin et al., 2020). A first recent work that does tackle the problem of co-generating sequential data with covariates is the work of Lin et al. (2020). The authors propose DoppelGANger, a generative adversarial network specifically designed for networked time series data characterized by long sequences, as opposed to health event data or treatment sequences which are generally short. Nonetheless, the provided framework is - from a technical perspective - flexible enough to handle health event data, which is presumably even simpler in the sense that it is much less dependent on long-term dependencies compared to networked time series data. However, the DoppelGANger architecture has not yet been evaluated on any data set containing health event data.

Next to the above-mentioned GAN-based methods for sequential data, there exist a few studies focusing on the generation of synthetic, longitudinal EHRs, i.e., sequences of visits to health care organizations (Che et al., 2017; Lee et al., 2020; Zhang et al., 2020a). Each visit (may) contain(s) intense information and multiple clinical events, such as diagnoses, procedures, and/or drug prescriptions. Given the variable number of events within a visit, the papers partly focus on representation learning in order to design a compact representation for each visit in a computable form using deep learning techniques as transformer encoders (Zhang et al., 2020a), Word2vec embeddings (Che et al., 2017) or generative autoencoders (Lee et al., 2020). Hereafter, the embeddings are used to train generative models including GANs. The setting of longitudinal EHRs consisting of patient visits including multiple events per visit is considerably more complex than our setting of sequential treatments, where each event includes one treatment only. Presumably, the process of representation learning using deep learning techniques is unnecessary in our simpler use-case of sequential treatments. Moreover, the three studies mentioned in this section do not focus on co-generation of patient covariates, where considered an important aspect of this thesis.

Another work in the context of medical sequential data is the work of Dash et al. (2019) and its follow-up work in 2020 (Dash et al., 2020). Different from the other research mentioned in this section, these two studies have an explicit focus on the co-generation of health event data and patient covariates. The authors propose a distinct approach, in which the focus lies on providing a flexible, efficient workflow for both joint modeling and synthesis of static and temporal variables. The first three steps of the workflow are as follows: (1) Identify appropriate summary static(s) for time series variables (e.g., mean, median, etc.), (2) Compute summary statistic(s) for fixed time-intervals over the whole time period, and (3) Append summary statistic(s) to static variables to create a transformed, cross-sectional data frame including both static and time series data. Hereafter, one can use an existing generative model - operating on cross-sectional data - of their choice to generate the transformed data. While seemingly comparable to our setting, Dash et al. (2020) focus on real-valued time series that can be divided into fixed time intervals, being inherently different from our use-case of discrete treatment sequences with varying time intervals.

2.1.3 Suitability of existing algorithms for the use-case of health event data

When assessing the sequential synthetic data generation algorithms mentioned in Section 2.1.2 on suitability for the use case of health event data, one algorithm stands out: DoppelGANger (Lin et al., 2020). As mentioned before, DoppelGANger satisfies our most important criterion: co-generation of covariates and sequential data. The method can - at least technically - be directly applied to the patient covariates and treatment sequences of colorectal cancer patients in the NCR. Moreover, the algorithm has shown to outperform other sequential generative models developed in the medical context, with experiments on networked time series data. Additionally, the generative methods designed for longitudinal EHR data (Che et al., 2017; Lee et al., 2020; Zhang et al., 2020a) are likely more complex than needed - especially the parts on representation learning - for the generation of treatment sequences. Next to that, these models do not allow for the co-generation of patient covariates. Therefore, we consider DoppelGANger the most suitable sequential generative model for our use case.

On the other hand, Dash et al. (2020) provide an interesting approach, in which the focus lies on the joint modeling of patient covariates and temporal data in a cross-sectional data frame, rather than designing algorithms specific for sequential data. Their joint modeling approach is applicable to continuous time series data that can be divided into fixed time intervals. Nonetheless, we can be inspired by the work of Dash et al. (2020) and develop a method to jointly model patient covariates and event sequences. Assuming that we can define such a workflow for the use case of discrete, irregularly sampled health event data, we can choose from the wide range of existing generative algorithms for cross-sectional data described in Section 2.1.1 for the co-generation of patient covariates and treatment sequences of colorectal cancer patients. Since IKNL has already implemented Bayesian networks (Zhang et al., 2017), we consider it interesting to assess how we can adapt this method and extend it to our current use case on patient data *including* sequential treatments.

2.2 Background on selected generative methods

In this section, we describe two existing generative methods that will be used in this research, as well as a baseline method. The motivation for the selection of these methods will be elaborated upon in Section 4.2.1. Here, we explain the data generation mechanism and basic principles and concepts of these methods.

2.2.1 PrivBayes

This section describes the PrivBayes method proposed by (Zhang et al., 2017). In order to gradually explain the working of the PrivBayes algorithm, we first discuss the general working of the method, leaving details (with regards to differential privacy) out. Hereafter, we cover an elaborate and formal explanation of the PrivBayes method, including details regarding differentially private data synthesis in PrivBayes.

General explanation - Data synthesis through Bayesian networks

PrivBayes (Zhang et al., 2017) is a non-parametric method for releasing high-dimensional data using Bayesian networks. In general, a Bayesian network is a method to compactly describe and approximate the full-dimensional distribution of a data set, by assuming conditional (in)dependence among the attributes in the data set. More specifically, a Bayesian network is a directed acyclic graph (DAG) that (1) portrays each attribute in the data set as a node, and (2) models relations between attributes using directed edges. If there is a directed edge from attribute Y to attribute X, Y is defined as a *parent* of X, and the set of all parents of X (i.e., all nodes with directed edge to node X) is referred to as its *parent set*.

Using the concept of Bayesian networks, PrivBayes consists of three steps (Zhang et al., 2017):

1. Construct a Bayesian Network over the attributes in the data set. The network is constructed greedily by iteratively selecting attributes that have the highest mutual information with (a subset of) the previously selected attributes. This attribute is then added to the network with this (subset of) the previously selected attributes as its parent set. Considering only the previously selected attributes for the parent set of the currently modeled attribute ensures that the network is acyclic, namely a DAG. The first node (i.e., the root of the network) is selected randomly.

2. Determine the conditional probabilities between each node and its parent set in the Bayesian network defined in step 1.

3. Use the Bayesian network (constructed in step 1) and the conditional distributions (constructed in 2) to derive an approximate distribution of the data set. Then, sample instances from the approximate distribution to generate a synthetic data set.

An example Bayesian network on four attributes in a fictionary data set is shown in Figure 2.1. Here, ’Age’ was randomly selected as the first node (i.e., the root of the network). Then, the attribute ’Education’ is selected as the second node, with parent

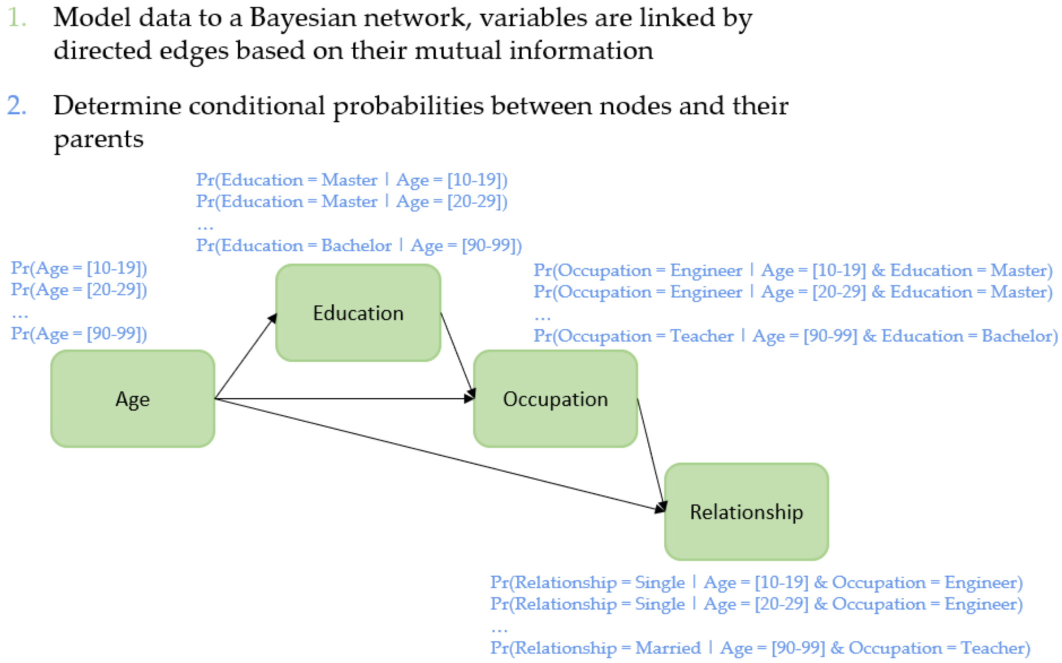

1. Model data to a Bayesian network, variables are linked by directed edges based on their mutual information
2. Determine conditional probabilities between nodes and their parents
Pr(Education = Master | Age = [10-19])
Pr(Education = Master | Age = [20-29])
...
Pr(Education = Bachelor | Age = [90-99])
Pr(Age = [10-19])
Pr(Age = [20-29])
...
Pr(Age = [90-99])
Education
Age
Occupation
Relationship
Pr(Occupation = Engineer | Age = [10-19] & Education = Master)
Pr(Occupation = Engineer | Age = [20-29] & Education = Master)
...
Pr(Occupation = Teacher | Age = [90-99] & Education = Bachelor)
Pr(Relationship = Single | Age = [10-19] & Occupation = Engineer)
Pr(Relationship = Single | Age = [20-29] & Occupation = Engineer)
...
Pr(Relationship = Married | Age = [90-99] & Occupation = Teacher)

FIGURE 2.1: Example Bayesian network.

set {'Age'}, meaning that this attribute-parent (AP) pair had the highest mutual information among the candidates [('Education', {'Age'}), ('Occupation', {'Age'}), ('Relationship', {'Age'})]. This procedure of adding attributes based on mutual information continued until all attributes were added to the network. Then, during step 2, the conditional probabilities (shown in blue in the figure) for each node given its parents were computed. As the node 'Age' - as the root of the network - has no parents, its marginal probability distribution is determined. Using the Bayesian network structure and the conditional (or, marginal) probabilities obtained, we can sample a synthetic data set in order of attribute addition to the network. In this network, this would imply sampling the attributes for an instance in the following order: 'Age', 'Education', 'Occupation', 'Relationship'.

Detailed explanation - Differentially private Bayesian networks

In this section, we detail the PrivBayes algorithm, including components designed for satisfying differential privacy. While we did not cover any details related to differential privacy in this thesis yet, for the explanation of PrivBayes it is important to understand the general idea of differential privacy. Differentially private algorithms introduce a small amount of noise into the output that ensures that the presence of an individual cannot be inferred. The amount of noise introduced is determined by the sensitivity of the output (i.e., the change in output when one individual is added or removed from the data set), as well as a user-defined parameter ε: the privacy budget. Smaller values of ε correspond with a higher scale of noise to be added and thus imply a stronger privacy guarantee. Noise may be introduced using several mechanisms, where the Laplace mechanism (Dwork et al., 2006) and the exponential mechanism (McSherry & Talwar, 2007) can be used for noise addition to continuous outputs and outputs consisting of a discrete set of alternatives respectively. Finally, a property of differential privacy used in PrivBayes is its composability property.

The composability property ensures that when a set of m sub-tasks obtain differential privacy with privacy budgets $\varepsilon_1, \varepsilon_2, ..., \varepsilon_m$ respectively, the algorithms as a whole satisfies $(\sum_i \varepsilon_i)$-differential privacy. For a more elaborate explanation on differential privacy, we refer to Section 2.4.1.

Altogether, taking into account the composability property, PrivBayes satisfies $(\varepsilon_1 + \varepsilon_2)$-differential privacy. More specifically, the first (Bayesian network learning) and second (conditional distribution construction) step require direct access to the input dataset, and each consumes ε_1 and ε_2 privacy budget respectively. The third step (sampling) requires no access to the original data set and therefore does not incur any privacy cost. The parameter β partitions the total privacy budget ε into ε_1 and ε_2 by assigning $\varepsilon_1 = \beta\varepsilon$ and $\varepsilon_2 = (1 - \beta)\varepsilon$, and hereby balances the quality of the network learning and distribution learning in PrivBayes (Zhang et al., 2017). In the following two paragraphs, we cover an in-depth description of step 1 (differentially private Bayesian network learning) and step 2 (differentially private generation of conditional distributions) of the PrivBayes algorithm. For this explanation, we use the notation employed by Zhang et al. (2017), shown in Table 2.1.

Notation	**Definition**
$\mathcal{D}$	A sensitive data set to be published
n	The number of tuples in $\mathcal{D}$
$\mathcal{A}$	The set of attributes in $\mathcal{D}$
d	The number of attributes in $\mathcal{A}$
$\mathcal{N}$	A Bayesian network over $\mathcal{A}$
$Pr[\mathcal{A}]$	The distribution of tuples in $\mathcal{D}$
$Pr_{\mathcal{N}}[\mathcal{A}]$	An approximation of $Pr[\mathcal{A}]$ defined by $\mathcal{N}$
dom(X)	The domain of random variable X

TABLE 2.1: Notation used for explanation of PrivBayes

Step 1: Differentially private Bayesian network learning

Algorithm 1 presents the greedy algorithm used for differentially private Bayesian network learning in Zhang et al. (2017). At the start of the algorithm (Line 1), the Bayesian network $\mathcal{N}$ is defined as an empty set of AP pairs. Let V represent the set of attributes whose parent set has been defined at the current point in the partial construction of $\mathcal{N}$. Thus, at the beginning of the algorithm, V is defined as an empty set. Then, the algorithm randomly picks an attribute from $\mathcal{A}$ (denoted as X_1) as the root of $\mathcal{N}$, i.e., its parent set Π_1 is set to $\emptyset$ (Line 2). Next, the algorithm consists of $d - 1$ iterations, in which it greedily selects an AP with ε_1-differentially private highest mutual information score (derived with the Exponential Mechanism) from a candidate set Ω and adds it to $\mathcal{N}$ (Lines 3-12). Details on the scoring function for determining the mutual information of an AP pair are beyond the scope of this research and we refer to the article of Zhang et al. (2017) here for.

The `MaximalParentSets` function is used to limit the parent set size of an attribute X in the Bayesian network and is highly relevant for the differentially private version of PrivBayes. This aspect is related to the addition of noise to construct a set of noisy (i.e., ε_2 differentially private) conditional distributions (step 2). Intuitively, the reasoning is as follows. A higher degree Bayesian network (i.e., lower restriction on the parent set size of attributes) retains more information from the original distribution $Pr[\mathcal{A}]$. For example, a $(d-1)$-degree Bayesian network estimates $Pr[\mathcal{A}]$ without suffering from any information loss. However, increasing the size of the

Algorithm 1: GreedyBayes (Algorithm 4 in Zhang et al. (2017))

```
initialize $\mathcal{N} = \emptyset$ and $V = \emptyset$;
randomly select an attribute $X_1$ from $\mathcal{A}$; add $(X_1, \emptyset)$ to $\mathcal{N}$; add $X_1$ to $V$;
for $j = 2$ to $d$ do
    initialize $\Omega = \emptyset$;
    foreach $X \in \mathcal{A} \setminus V$ do
        find all maximal parent sets of $X$, that is $\top(X) =$ MaximalParentSets$(V, \frac{n\varepsilon_2}{2d\theta|dom(X)|})$;
        if $\top(X) = \emptyset$ then
            add $(X, \emptyset)$ to $\Omega$;
        else
            foreach $\Pi \in \top(X)$ do add $(X, \Pi)$ to $\Omega$;
    select $(X_j, \Pi_j)$ from $\Omega$, using a mutual information scoring function with privacy budget $\varepsilon_1/(d-1)$;
    add $(X_j, \Pi_j)$ to $\mathcal{N}$; add $X_j$ to $V$;
return $\mathcal{N}$;
```

parent set of an attribute also increases the dimensionality of the joint distributions. High-dimensional distributions are more vulnerable to noise and thus the noisy distributions become less useful, especially when the privacy budget ε is small, leading to a synthetic data set full of randomness. Hence, the choice of the size of the parent set of an attribute should balance the informativeness of the constructed Bayesian network and the robustness of the probability distributions. Zhang et al. (2017) denote this balance as the *signal-to-noise ratio* of the joint probabilities. This balancing act is influenced by three parameters: the privacy budget ε, the total numbers of tuples n in the data set, and the defined usefulness θ of each noisy distribution. θ-usefulness can be defined as follows: a noisy distribution is θ-useful if the ratio of the average scale of information to the average scale of noise (i.e., signal-to-noise ratio) is no less than θ. More specifically, given an Attribute-Parent (AP) pair (X, Π) with m cells in the joint distribution $Pr[X, \Pi]$ - where m is the product of the domain sizes of all attributes in $\{X\} \cup \Pi$ - the average scale of information in each cell is $1/m$. As will be further explained in the following paragraph on step 2, the average scale of noise in each cell is $2d/n\varepsilon_2$. Therefore, $Pr[X, \Pi]$ is θ-useful only if $m \leq n\varepsilon_2/2d\theta$.

It is important to note that when using a non differentially private version of the PrivBayes algorithm (i.e., $\varepsilon = \varepsilon_1 = \varepsilon_2 = \infty$), there exists no limitation on the parent set sizes of attributes. In other words, the maximal parent set $\top(X) = V$ for every X. That is, each attribute that is not yet added to the Bayesian network at this point ($X \in \mathcal{A} \setminus V$) is added to the candidate set Ω with parent set V (i.e., (X, V) is added to Ω for every $X \in \mathcal{A} \setminus V$). This means that in the non differentially private version of PrivBayes, the GreedyBayes algorithm returns a $(d-1)$-degree Bayesian network that estimates $Pr[\mathcal{A}]$ without suffering from any information loss.

Step 2: Generation of differentially private conditional distributions

Once we have defined the Bayesian network in step 1, we can construct the approximate distribution of tuples in $\mathcal{D}$ ($Pr_{\mathcal{N}}[\mathcal{A}]$). For this, we need d conditional distributions $Pr[X_j|\Pi_j](j \in [1, d])$. The algorithm for generating the conditional distributions by Zhang et al. (2017) first derives the joint distributions $Pr[X_j, \Pi_j]$, which can be calculated from data set $\mathcal{D}$ and the parent sets for each attribute X in $\mathcal{N}$ defined by the GreedyBayes algorithm in step 1. Then, Laplace noise (with scale such that each joint distribution satisfies ε_2/d-differential privacy) is injected into $Pr[X_j, \Pi_j]$ to retrieve a noisy joint distribution $Pr^*[X_j, \Pi_j]$. Given that $Pr[X_j, \Pi_j]$ has sensitivity $2/n$ (i.e., the maximum change in output when changing one record's information), the exact needed scale of Laplace noise added to $Pr[X_j, \Pi_j]$ is $2d/n\varepsilon_2$ and is thus influenced by the number of attributes in $\mathcal{A}$ (d), the number of instances in the data set n and the user-defined privacy budget ε. In order to ensure that $Pr^*[X_j, \Pi_j]$ is a proper probability distribution, after the addition of noise, all negative numbers in $Pr^*[X_j, \Pi_j]$ are set to zero, and then all values are normalized to preserve a total probability mass of 1. Finally, from $Pr^*[X_j, \Pi_j]$, the algorithm materializes a noisy version of the conditional distribution $Pr[X_j|\Pi_j]$, indicated by $Pr^*[X_j|\Pi_j]$.

2.2.2 DoppelGANger

This section covers an explanation of the DoppelGANger framework proposed by Lin et al. (2020). It starts with a preliminary on the general method used in DoppelGANger: Generative Adversarial Networks (GANs), followed by a description of the specific DoppelGANger method designed by Lin et al. (2020).

Preliminary: Generative Adversarial Networks (GAN)

Generative adversarial networks (GANs) (Goodfellow et al., 2014) are a class of deep neural networks. GANs have originally been proposed for the generation of high-quality synthetic images. This GAN framework has further been extended to other applications as text-to-image synthesis, video generation, music generation, tabular data generation, or sequential data generation. A GAN consists of two components: a generator G that makes an effort to produce realistic, but fake, data and a discriminator D that intends to distinguish between the fake data outputted by the generator and the real data. The generator takes as input a noise vector from which it outputs a fake sample. Then, the discriminator takes samples from both the real training data and the fake samples generated by the generator to determine whether they are real or fake.

By playing an adversarial minimax game against each other, the generator can learn the distribution of the real samples. Errors in the classification task are used to train the parameters of both G and D using backpropagation. Formally, the minimax game between G and D with value function $V(G, D)$ is described in Equation 2.1, where p_{data} is the real data distribution and p_z is the simple noise (input) distribution.

$$\underset{G}{min}\underset{D}{max} V(G, D) = \mathbf{E}_{x \sim p_{data}(x)}[log D(x)] + \mathbf{E}_{z \sim p_z(x)}[log(1 - D(G(z)))] \tag{2.1}$$

DoppelGANger architecture

As mentioned in Section 2.1.2, DoppelGANger (DG) is designed for the co-generation of metadata and networked time series data, characterized by long sequences. Some of the proposed design elements by Lin et al. (2020) are specifically focused on the long-term temporal correlations within networked time series data and are not relevant to our use case of short treatment sequences of colorectal cancer patients. Fortunately, the implementation of DoppelGANger is flexible, and we can select which aspects of the architecture to keep or remove for our use case. Here, we will focus on the design elements proposed by Lin et al. (2020) that are relevant to our use

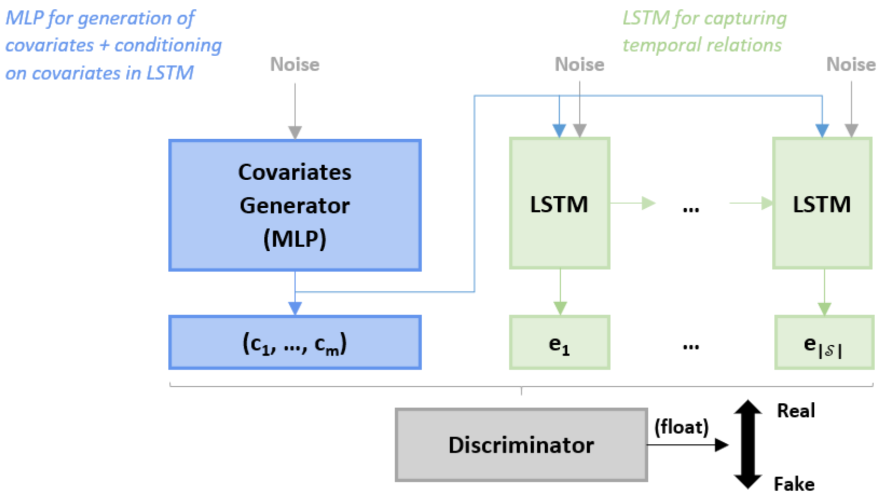

MLP for generation of covariates + conditioning on covariates in LSTM
LSTM for capturing temporal relations
Noise
Noise
Noise
Covariates Generator (MLP)
LSTM
...
LSTM
$(c_1, ..., c_m)$
e_1
...
$e_{|\mathcal{S}|}$
Discriminator
(float)
Real
Fake

FIGURE 2.2: DG architecture (Lin et al., 2020) highlighing key concepts relevant to this research. e represents a single (generated) event (i.e., time step) within the event sequence and $|\mathcal{S}|$ represents the length of the event sequence.

case, mainly centered around capturing relationships between covariates and event sequences.

Most importantly, the DG architecture decouples the co-generation of covariates (referred to as metadata by Lin et al. (2020)) and sequences (referred to as measurements by Lin et al. (2020)) into two sub-tasks: (1) generating covariates (c) and (2) generating sequences ($\mathcal{S}$) *conditioned* on covariates: $P(c_p, \mathcal{S}_p) = P(c_p) \cdot P(\mathcal{S}_p|c_p)$. For each task, a dedicated generator is used that is most applicable for the task at hand. More specifically, a standard multi-layer perceptron (MLP) network is used for generating the covariates. MLP networks are state-of-the-art for modeling tabular data. For sequence generation, DG uses a variant of RNN called long short-term memory (LSTM) that is designed and widely used to model sequential data. Instead of generating the entire sequence at once, LSTMs generate one time step at a time and then run several passes to generate the entire sequence. Different from MLPs, LSTMs have an internal state that implicitly encodes and remembers the past states of the time series. Thus, when generating a specific time step, the LSTM unit can incorporate the patterns in all time steps before. To preserve the relationships between the covariates and the event sequence, the generated metadata c_p is added as a conditional input to the LSTM at every step.

A schematic representation of the overall DG architecture and the relevant concepts included in this research is in Figure 2.2. The decoupled generation of covariates and event sequences is shown as the MLP network for the generation of covariates in blue and an unfolded representation of the LSTM network for the generation of event sequences in green. Also, the conditioned generation of events based on the covariates generated is shown, indicated by the blue arrows going into the LSTM cells.

Besides capturing relations between covariates and sequences, another challenge is that sequences may have varying lengths. Lin et al. (2020) designed DG to deal with the generation of varying length sequences using a generation flag. Along with the original events, DG adds this generation flag to each time step: $[1, 0]$, if the sequence does not end at this time step, and $[0, 1]$ if the sequence ends exactly at this time step. The generator learns and outputs the generation flag $[p_1, p_2]$ through a softmax output layer, so that $p_1, p_2 \in [0, 1]$ and $p_1 + p_2 = 1$. Then, $[p_1, p_2]$ is used to control whether to continue unrolling the LSTM to the next time step. More specifically, if $p_1 < p_2$, generation is stopped and all future events are padded with 0's; if $p_1 > p_2$, LSTM unrolling is continued to generate events for the next time step.

2.2.3 Marginal Synthesizer

The Marginal Synthesizer makes use of a histogram representation when generating synthetic data. It represents each feature in the data set as a histogram and then samples synthetic data according to the histogram probabilities (or, marginal probabilities) for each feature. Given the fact that each feature is modeled independently (according to its marginal distribution), the Marginal Synthesizer does not take into account relations between features. This method is included in our research solely to serve as a baseline for the other two more complex approaches that do take relations between features into account. In general, we expect that the synthetic data generated by Marginal Synthesizer captures individual feature distributions well, while poorly capturing relations between features.

Differential privacy guarantees can be implemented within the Marginal Synthesizer by introducing noise within the marginal distributions. The differentially private Marginal Synthesizer will add noise that is desired for a defined parameter ε to the histogram probabilities.

2.3 Evaluating the quality of synthetic data

When evaluating the generative model and its synthetic data output, the goal is to validate that the learning process has led to a sufficiently close approximation of the original data set. However, evaluating the quality of synthetic health data is a nontrivial task in practice. The concept of "realism" of samples finds a fairly natural application to images or text but becomes more opaque when faced with the complexity of (a combination of) cross-sectional or sequential (health) data. Where synthetic images and text can be evaluated by humans through simply observing individual samples, evaluating cross-sectional or sequential data focuses more on comparing aggregate statistics and patterns in the data.

Related studies define various - sometimes overlapping - methods for evaluating the quality of the generated synthetic data. Typically, quality metrics differ between papers - in general, or based on the data they focus on - and do not always give a comprehensive view. Georges-Filteau and Cirillo (2020) categorize quality metrics for synthetic data. While their overview focuses on synthetic health data generated by GANs, these quality metrics focus on the synthetic data itself and can therefore be applied irrespective of the generative model in place.

Quality metrics can broadly be categorized into qualitative and quantitative quality metrics. Qualitative metrics are mainly centered around expert judgments. Tasks are focused on (clinical) experts and include choosing the most realistic of two data points in pairs of one real and one synthetic (e.g. Choi et al., 2017), classifying data points as real or synthetic one by one (e.g. Beaulieu-Jones et al., 2019) or rating data points for realism according to a predefined numerical scale (e.g. Beaulieu-Jones et al., 2019). Another type of qualitative evaluation may be comparing visual representations of original and synthetic data set statistics. In general, qualitative quality metrics based on visual inspection (by experts) are considered weaker measures for data quality, mostly used as a supporting role for quantitative metrics. However, strengths of (visual) qualitative measures include their allowance for gaining quick insights into underlying reasons for synthetic data set quality and their property of often being easier to communicate to non-expert users.

On the other hand, quantitative measures provide a more convincing indication of synthetic data quality (Georges-Filteau & Cirillo, 2020). Georges-Filteau and Cirillo (2020) further divide quantitative metrics into three loosely defined categories: (1) comparing and evaluating distributions over the full data set (*Data set distribution metrics*), (2) comparing the statistical properties of the real and synthetic data, including marginal statistics as well as relations between features (*Statistical metrics*), and (3) evaluating the quality of the data indirectly by assessing the work that can be done with the synthetic data, also called utility (*Utility metrics*).

Firstly, data set distribution metrics include aggregates of feature distribution metrics over the entire data set. Examples of metrics that compare the distribution of a feature are Kullback-Leibler (KL) divergence (e.g. Goncalves et al., 2020) and Jensen-Shannon (JS) distance (Knoors, 2018). Additionally, prior work has proposed statistical tests as data set distribution metrics, with the null hypothesis that real and synthetic samples originate from the same distribution. Examples include Maximum Mean Discrepancy (MMD) (Esteban et al., 2017) and Kolmogorov-Smirnov (KS) tests (Baowaly et al., 2019). While these data set distribution metrics give a general representation of the overall quality of the entire synthetic data set, the metrics do not provide transparency on which features are (not) comparable between the real and synthetic data.

Second, statistical metrics focus on statistical similarity between real and synthetic data. Examples of statistical measures include feature distribution divergence metrics mentioned above (JS distance and KL divergence), but in this case on a per feature basis. Another metric that compares real and synthetic data feature-wise is dimension-wise distribution (e.g., success probability for binary features (Baowaly et al., 2019; Choi et al., 2017)). Next to evaluation metrics on univariate feature distributions, more extensive quality metrics focused on the *relations* between features are proposed. Extensions of the univariate metrics include higher-order marginals that compare the joint distributions of features, using for example j-way marginals (Zhang et al., 2017). Another statistical metric focusing on the relations between features is inter-dimensional correlation. Often, correlation matrices are both visually compared, together with a quantitative metric on the difference between real and synthetic correlation matrices (e.g. Beaulieu-Jones et al., 2019; Dash et al., 2019; Goncalves et al., 2020; Yoon et al., 2020). Other statistical measures proposed in related work focused on a specific aspect of the data include support coverage (Goncalves et al., 2020), which penalizes synthetic data sets if less frequent categories are not well represented, and log cluster metrics (Goncalves et al., 2020) which represents the (dis)parity of cluster memberships among real and synthetic samples. While statistical metrics are often relatively interpretable, these metrics are unlikely to paint a full picture (Georges-Filteau & Cirillo, 2020). Prior works mostly agree that no single statistical metric on its own was sufficient. Therefore, often a combination of statistical metrics is needed to allow a deeper understanding of the quality of synthetic data.

Finally, while lacking the interpretability of statistical measures, utility metrics often provide the strongest indication of data realism (Georges-Filteau & Cirillo, 2020). Esteban et al. (2017) propose two novel techniques to evaluate the synthetic data from a utility - or more specifically, a predictive - perspective. Firstly, Esteban et al. (2017) define "**T**rain on **S**ynthetic, **T**est on **R**eal" (TSTR), in which the synthetic data set is used to train a prediction model and subsequently this model is tested on a held-out set of real examples. The TSTR evaluation metric is a useful metric for demonstrating the ability of the synthetic data to be used for real applications. On the other hand, Esteban et al. (2017) come up with a second evaluation metric "**T**rain on **R**eal, **T**est on **S**ynthetic" (TRTS). In this approach, the real data is used to train a prediction model on a data set specific task, and subsequently, this model is tested on a synthetic data set generated by the generative model. TRTS is a useful metric for evaluating if the relations with the target variable in the original data set are retained in the synthetic data set. However, the downside of TRTS is that its performance will not degrade when there exists limited diversity in the synthetic data set, i.e., the synthetic data set only represents a subspace of the real data distribution. Both metrics require the real data to have meaningful labels.

Given the limitation of TRTS to identify limited diversity in the synthetic data, TSTR is considered the most interesting evaluation (Esteban et al., 2017). Presumably given this consideration, the most widely used predictive evaluation metric in the literature is an extension of TSTR. This most commonly used metric works as follows: (1) train (and optimize) two prediction models - one on the original data set and one on the synthetic data set), and (2) evaluate both models on the hold-out set of real examples and report the difference in test performance (**T**rain on **B**oth, **T**est on **O**riginal **H**oldout set (TB-TOH)). The difference in the performance of the models trained (one on the original and one on the synthetic data set) can be considered a measure of how much the quality of the synthetic data is affected. Most related studies perform this extension of the TSTR measure on one prediction task that is considered appropriate for the data set at hand (e.g. Mendelevitch & Lesh, 2021; Xu & Veeramachaneni, 2018; Yoon et al., 2020), while a few studies take a more extensive approach and repeat the procedure for each variable as the target and report the average value (e.g. Choi et al., 2017; Goncalves et al., 2020).

Some research goes even further by using metrics related to prediction performance to get a more in-depth representation of the quality of the synthetic data. For example, Jordon et al. (2019) argue that in certain specific contexts, it is important that the relative performance of two algorithms when trained and tested on the synthetic data is similar to their relative performance when trained and tested on the original data. Therefore, the authors propose a relative algorithm performance metric that compares the performance ranking of different machine learning models trained on the original versus the synthetic data set. Another example of a metric related to prediction performance that gives a more extensive representation of synthetic data quality is feature importance comparison, which evaluates the extent to which classifiers trained on the real versus the synthetic data were relying on the same features while making their predictions (Beaulieu-Jones et al., 2019).

Finally, machine learning tasks other than predictive metrics that evaluate the utility of the synthetic data are proposed. Baowaly et al. (2019) were the first to propose a machine learning metric focusing on Association Rule Mining (ARM) in their use-case of static (binary) vectors of patient's diagnosis and procedures. ARM is used to discover commonly occurring variable-value pairs among a large set of variables (Agrawal, Srikant, et al., 1994). ARM is widely used on health data to identify patterns and co-occurrences among clinical variables (Yadav et al., 2018), and is therefore considered an appropriate evaluation task in the domain of synthetic health data generation, in order to examine whether rules present in the original data are also present in the synthetic data. In particular, the ARM algorithm is employed on both the original and synthetic data set to identify frequent association rules in each of the two data sets. To compare the rules found from the original data set to the rules found from the synthetic data set, precision and recall metrics are used.

Multiple other studies evaluate the synthetic data using - amongst others - this ARM evaluation metric (Kaur et al., 2020; Yan et al., 2020).

2.4 Privacy of synthetic data

As mentioned before, incorporating privacy constraints and maintaining the utility of synthetic data might be two conflicting goals. In general, two approaches to privacy incorporation in the synthetic data generation framework exist: 1) conferring traditional privacy guarantees in the training process, and 2) evaluation of synthetic data privacy after generation. Section 2.4.1 further describes the first approach, where the latter approach is detailed in Section 2.4.2. Notably, some papers do not take into account any evaluation of privacy, nor do they incorporate privacy guarantees (e.g., Xu et al., 2019). For IKNL, the preferred approach is the first approach in which theoretical privacy guarantees are incorporated, as the cancer data at hand is inherently sensitive medical data. Additionally, differential privacy - as opposed to post-generation privacy metrics - protects against database linkage attacks, which may occur when hospitals that store (additional) data on a subset of patients in the NCR encounter a data leak.

2.4.1 Differential privacy

Differential privacy (DP) (Dwork & Roth, 2014) is an important concept in the literature on privacy as well as synthetic data generation and has emerged as a state-of-the-art standard for managing privacy risks. Essentially, differential privacy is a mathematical definition of privacy that requires that the answer to any query (or, synthetic data output) be "probabilistically indistinguishable" with or without the presence of a particular individual in the original data set (Dankar & El Emam, 2013). Let D be a sensitive data set to be released. Differential privacy claims that any release of information about D should be done via a randomized algorithm G, such that the output of G reveals very little information about any particular individual in D. Formally, differential privacy can be defined as follows:

Definition 1. ε-Differential Privacy (Dwork et al., 2006). A randomized algorithm G is ε-differentially private if for any two data sets D_1 and D_2 that differ only by one individual, and for any possible output O of G, we have

$$Pr[G(D_1) = O] \leq exp(\varepsilon) * Pr[G(D_2) = O] \tag{2.2}$$

Thus, the formula considers a randomized algorithm, i.e., one in which noise is added to the output. The ratio of the probability that the randomized algorithm outputs any possible output O when the data set is D_1 and the probability that the randomized algorithm outputs any possible output O when one individual is added/removed from the data set (i.e., the data set is D_2) is at most $exp(\epsilon)$. In other words, $exp(\epsilon)$ constrains how different the probabilities of obtaining any specific output of the randomized algorithm of any two data sets differing by one individual are. In other words, it quantifies the maximum effect any single individual may have on the output of the algorithm. In general, the definition states that the lower the value of ϵ (given that it should be non-negative), the closer the value of $exp(\epsilon) = 1$, the closer the probabilities of obtaining any possible output O for D_1 and D_2 differing in one individual and thus the stronger the privacy guarantee.

In short, differentially private algorithms introduce a small amount of randomness (noise) into the output that ensures that the presence of an individual cannot be inferred. The amount of noise introduced is determined by a privacy parameter ε (privacy budget) that can be set by the user. Lower values of ε imply stronger privacy guarantees. In general, differential privacy introduces some kind of uncertainty or "plausible deniability", from which privacy follows. The need for differential privacy can best be explained by a simple example of the release of statistical information on a data set. Imagine a teacher that conveys the average grade for an assignment to the class through their online learning environment. Say the average grade of the class ($n = 20$ students) is 8.0. The next day, someone decides to drop out of the class, and in the online learning environment, the mean grade of the class (now consisting of $n = 19$ students) is automatically updated to 8.1. Based on this information, the grade for the drop-out student can easily be inferred ($8.0 * 20 - 8.1 * 19 = 6.1$). If the online learning environment had included a proper differentially private algorithm for computing the mean class grade, the inclusion of noise would have made it impossible to infer the grade of the drop-out student, as there is some uncertainty involved (i.e., the maximum change in output, or mean grade, after drop out of the student is bounded, and there exists a "plausible deniability" that the change is caused by pure randomness of the algorithm).

Two interesting properties of differential privacy are worth noting for this research. The first one is its composability property (Dwork & Roth, 2014). The composability property ensures that when a set of m sub-tasks obtain differential privacy with privacy budgets $\varepsilon_1, \varepsilon_2, ..., \varepsilon_m$ respectively, the algorithms as a whole satisfies ($\sum_i \varepsilon_i$)-differential privacy. The second property of differential privacy is that differentially private algorithms are 'future proof', meaning that the privacy guarantee of the output will not be affected by side information or post-processing (Dwork & Roth, 2014). Thus, it is safe to perform post-processing, which might reduce the noise or improve the signal of the output of the differentially private algorithm, without affecting the privacy level.

There is no specific guidance on the exact value of ε needed to provide "enough" privacy. This is partly considered a social question by Dwork and Roth (2014). The original authors consider low values of ε around 0.01 or 0.1 acceptable. Nonetheless, other literature also examines an $\varepsilon = 1$ acceptable (Arnold & Neunhoeffer, 2020; Lin et al., 2020). Values of ε higher than 5 are typically considered a potential privacy leak but might be still worthwhile exploring, depending on the data at hand.

Differential privacy mechanisms

In order to achieve differential privacy, a mechanism used for introducing the needed amount of noise is required. Several mechanisms used for introducing noise may be used. Here, we explain two of the most widely used ones, the Laplace mechanism (Dwork et al., 2006) and the exponential mechanism (McSherry & Talwar, 2007). These are also both relevant for the differentially private version of the PrivBayes algorithm used in this research. Both mechanisms build on the sensitivity of a function between two data sets D_1 and D_2 differing in at most one individual, where the sensitivity of a function f is formally defined as:

$$S(f) = \underset{D_1,D_2}{max} ||f(D_1) - f(D_2)||_1 \tag{2.3}$$

Here, $||\cdot||$ denotes the L_1 norm. In words, $S(f)$ presents the maximum possible change in the output of the function f when one individual is added or removed from the data set.

The Laplace Mechanism (Dwork et al., 2006) then achieves differential privacy by adding noise sampled from the Laplace distribution Laplace(σ) with mean 0 and scale σ to the output of f. The scale of Laplace depends on the sensitivity of f ($S(f)$) and the user-defined privacy budget ε: $\sigma = S(f)/\varepsilon$. Thus, the higher the sensitivity of f, the higher the scale of noise needed to obtain differential privacy. For the privacy budget ε the opposite holds, where the lower the value of ε, the higher the scale of noise needed to achieve differential privacy.

The exponential mechanism (McSherry & Talwar, 2007) is used when the output of f is a discrete set of alternatives, instead of a continuous value. The exponential mechanism delivers a differentially private version of f, by sampling from its output domain Ω. The sampling probability for each alternative $w \in \Omega$ is determined by a case-specific score function f_s, which takes as input any data set D and any element $w \in \Omega$ and outputs a continuous score $f_s(D, w)$ that measures the quality of alternative w, where a larger score corresponds with w being a better output with respect to D. An example of a score function is a function that determines what option would result in the highest mutual information, as in step 1 of PrivBayes (Zhang et al., 2017). The exponential mechanism now samples $w \in \Omega$ with a probability proportional to $exp(f_s(D, w))/2\Delta$). Δ controls the degree of privacy protection and the exponential mechanism achieves ε-differential privacy if $\Delta \geq S(f_s)/\varepsilon$. Again, the same relations between the scaling factor and the sensitivity of f_s and ε hold as for the Laplace mechanism.

2.4.2 Post-generation privacy metrics

Generative methods that do not incorporate the differential privacy standard take widely varying approaches to privacy evaluations of synthetic data. It is important to note that, unlike differential privacy, the post-generation synthetic data privacy evaluation approaches do not give any *theoretical* privacy guarantees. Instead, these privacy evaluations use ad-hoc notions of privacy which are only validated empirically (Jordon et al., 2019).

These empirical privacy analyses are often based on the definitions of presence disclosure and attribute disclosure. Presence disclosure occurs when an attacker can determine that the generative algorithm was trained on a data set including the record from a specific person p. More recently, presence disclosure has been termed as a membership inference attack (Shokri et al., 2017). Membership disclosure metrics often use certain distance measures between original and synthetic data samples. Examples of practical implementations of membership inference metrics are shown in Zhang et al. (2020a) and Goncalves et al. (2020). Here, one claims that a record p was present in the original data training data set if there exists at least one sample with a certain distance to the record p in the synthetic data set. Else, it is claimed not to be present in the original training set. Then, to compute the membership disclosure metric of a generative method m, a set of r records used to train the generative model (train records) and another set of r records from the holdout test set (test records) are defined. For each of these $2r$ patient records, the closest distance to a record in the synthetic data set generated by m is determined. If this distance is lower than a prescribed threshold, the record is claimed to be in the training set. For each claim, there are four possible scenarios: true positive (correct claim that the record is in the training set), false positive (incorrect claim that the record is in the training set), true negative (correct claim that the record is not in the training set), or false negative (incorrect claim that the record is not in the training set). Subsequently, precision and recall for the claim outcomes can be computed, where a precision of 0.5 and low values of recall correspond with high privacy of the synthetic data set.

Attribute disclosure on the other hand occurs when attackers can derive additional attributes about a specific person p based on a known subset of attributes about p. Practical computations of attribute disclosure metrics are presented in Choi et al. (2017), Zhang et al. (2020a) and Goncalves et al. (2020). In this case, the attacker can first extract the k nearest neighbors for all records in the synthetic data set based on the subset of known attributes. Then, the unknown attributes can be estimated via a majority voting rule. Subsequently, the precision of this task is used as the main metric for attribute disclosure. Studies often use varying numbers of known attributes and the number of nearest neighbors k to study the attribute disclosure of a generative method.

Platzer and Reutterer (2021) propose another interesting post-generation measure of privacy - mostly related to membership inference - centered around making a strong case for plausible deniability for any individuals present in the original training data set, even for cases of a strong resemblance of a particular synthetic record with an original subject. The authors propose to calculate for each synthetic record its Distance to Closest Record (DCR) with regards to the original training data set as well as to an equally sized holdout test set. Then, the share of synthetic records that have DCR closer to a training than to a holdout record is the suggested measure for privacy risk. If this share is close to 50%, there exists an empirical claim for plausible deniability. However, as mentioned, it is important to note that, as for the other post-generation measures, this privacy metric does not give any theoretical privacy guarantee.

Chapter 3

Case study and data

3.1 Application background

This research is conducted in collaboration with IKNL. IKNL collects information of cancer patients in the Netherlands Cancer Registry (NCR), including patient, hospital, tumor, diagnostic, and treatment-related information. For this research, an anonymized (limited) data set is provided after signing an NDA.

3.2 Data set

3.2.1 Basic notation

We formally describe our data set as follows. The data set $P = \{1, 2, ..., p, ..., |P|\}$ is defined as a set of samples p (i.e., the patients). Each patient $p = (c_p, \mathcal{S}_p)$ contains m covariates $c_p = [{c_p}^1, {c_p}^2, ..., {c_p}^m]$. For example, covariate ${c_p}^1$ could represent the age at diagnosis of patient p and covariate ${c_p}^2$ the gender of patient p, and so on. Additionally, each patient record contains an ordered sequence of events (e.g., treatments) $\mathcal{S}_p = (e_{1,p}, e_{2,p}, ..., e_{|\mathcal{S}_p|,p})$, where $e_{i,p}$ is the i-th event of patient p. Different samples may contain a different number of events. That is, different patients may receive a different number of treatments. While general treatment guidelines exist, treatments are patient-specific due to for example a patient's medical history, the clinician's advise and the preferences of patients that are taken into account when making treatment decisions. The number of events for sample p is given by $|\mathcal{S}_p|$. Note that $\mathcal{S}_p$ is an ordered vector of events, i.e., these events followed each other chronologically in time.

3.2.2 General description

The NCR contains data on nearly all cancer incidences in the Netherlands, covering various cancer types. For this research, IKNL provided a data set with a more narrow scope focused on colorectal cancer patients. Colorectal cancer can be subdivided into three topographies: colon, rectosigmoid, and rectum. Nearly all incidences of colorectal cancer patients during the years 1998-2018 are included in the data set. The data set consists of two types of information: (1) patient and tumour related variables (e.g., age at diagnosis, tumour stage) and (2) treatment-related variables (e.g., treatment code, start date) of treatments received by patients in the data set. In total, the data set consists of 322,349 tumour episodes belonging to 268,764 unique patients. For registration purposes, multiple episodes may belong to the same primary tumour and patient. In addition, 518,469 treatments related to these incidences are included. These treatments cover 346 different treatment codes.

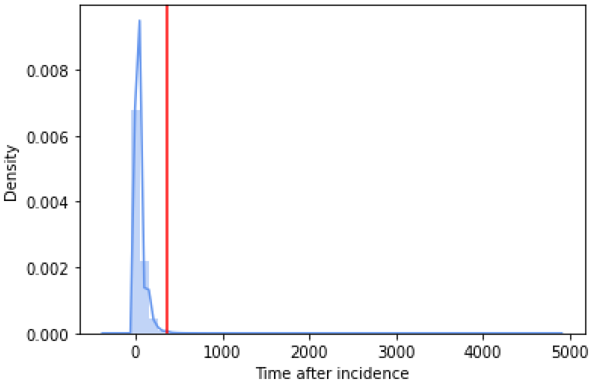


FIGURE 3.1: Time after incidence (days) distribution of treatments in data set. The red vertical line represents one year after diagnosis.

3.2.3 Preprocessing

In order to conduct this research, we took the following preprocessing steps to mitigate noise in the data. A complete overview of the exact steps taken and their effects on data set size is listed in Table 3.1.

1. In order to avoid having multiple episodes of one tumour or multiple tumour incidences per patient in the data set, we kept only the first tumour episode per patient.

2. For one of our evaluation metrics elaborated upon in Section 4.3.2, we perform a 1-year survival classification task. Our data set contains cancer incidences up until the end of 2018. However, for patients with cancer incidences during the year 2018, 1-year survival is unknown. In particular, we cannot ensure that for all patients with cancer incidence during the year 2018 a follow-up was conducted on whether they are still alive. Therefore, we removed all cancer incidences in 2018 from our data set.

3. For practical reasons, IKNL collects treatments of cancer patients up to 12 months after diagnosis. Data in the NCR is collected manually and therefore a threshold of one year after diagnosis is set as the collection date of all data related to a new cancer incidence, including treatment-related data. Nonetheless, this remains a limitation of the data set at hand, and the NCR in general. Incidentally, some later treatments are included for e.g. specific clinical trials. Figure 3.1 shows that the majority of treatments in the data set occur within one year after diagnosis (the red vertical line represents one year after diagnosis). We removed all treatments with a start date after 12 months after the cancer diagnosis of the patient to keep comparisons between patients fair.

4. Sequence lengths vary in the data set, with patients receiving more than 10 treatments. However, the vast majority of patients receive only a few treatments (Figure 3.2). We limit the sequence lengths present by removing all patients with more treatments than the 99% quantile of the sequence length distribution. The 99% quantile is a sequence length of 5 and thus we removed all patients with sequence lengths of 6 or more from the data set.

5. As indicated in Section 3.2.2, 346 different treatments are included in the data set. However, most of these treatments occur very infrequently. Following

> prior research (Yan et al., 2020), we ranked treatment codes based on their prevalence and removed those with an occurrence rate less than 1/1000. More specifically, we removed patients that have received a treatment with a rate of less than 1/1000. Thus, if a patient had 5 treatments, of which 4 are common (rate higher than 1/1000) and one is rare (rate less than 1/1000), we remove all data from this patient from the data set.

After all the preprocessing steps are conducted, the data set contains 396,654 treatments associated with 239,903 patients. The final data set includes 47 different treatment codes.

Step	Main action	Effect on data set size
1	Remove tumour epsiodes that are not the first epsiode per patient	Removal of 55,765 (17.30%) tumours
2	Remove patients with incidence date in 2018	Removal of 15,339 (5.79%) patients
3	Remove all treatments with start date after 12 months after diagnosis	Removal of 1,090 (0.24%) treatments
4	Remove patients with treatment sequence length longer than 5	Removal of 2,147 (0.86%) patients
5	Remove patients that received a treatment with an occurrence rate less than 1/1000	Removal of 7,655 (3.10%) patients

TABLE 3.1: Preprocessing steps including their main actions and effects on data set size

For this research, we limit the features included in the data set to reduce its dimensionality. The selection of features related to patient covariates (i.e., features in c) is based on domain knowledge (i.e., the influence of patient covariates on treatment decisions, based on clinical expert knowledge and guidelines) and the ability to predict 1-year survival with reasonable quality using these features. The data set includes two engineered features: *Age at diagnosis* (years between patient's date of birth and tumour incidence date) and *1-year survival* (binarized version of survival time: years between patient's date of death and tumour incidence date). Table 3.2 shows the included features and their respective data types and value ranges. Regarding treatment data, we only included treatment code as a feature within the

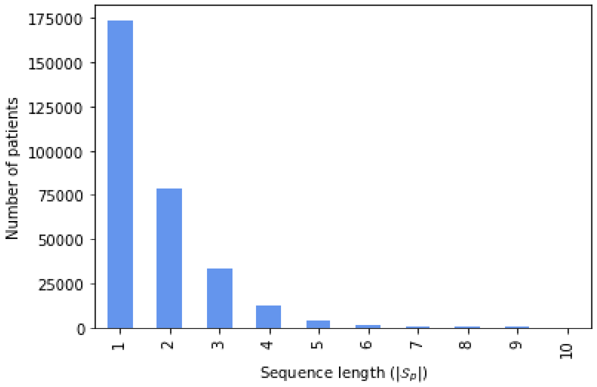


FIGURE 3.2: Sequence length distribution in raw data set. Actual sequence length distribution contains longer sequence lengths with low occurrences. These are omitted here for confidentiality.

Feature	Topology code	Gender at birth	Differentiation grade	Nr positive lymph nodes	Age at diagnosis	Sublocation	Stage	1 year survival
Type	Categorical	Binary	Categorical	Numerical	Numerical	Categorical	Categorical	Binary
Value range	(C18, C19, C20)	(Female, Male)	(1-4, 9)	(0-89)	(0-99)	(0-9)	(0-4, M, X)	(0,1)

TABLE 3.2: Selected patient covariates (c) including their data types and value ranges in the data set

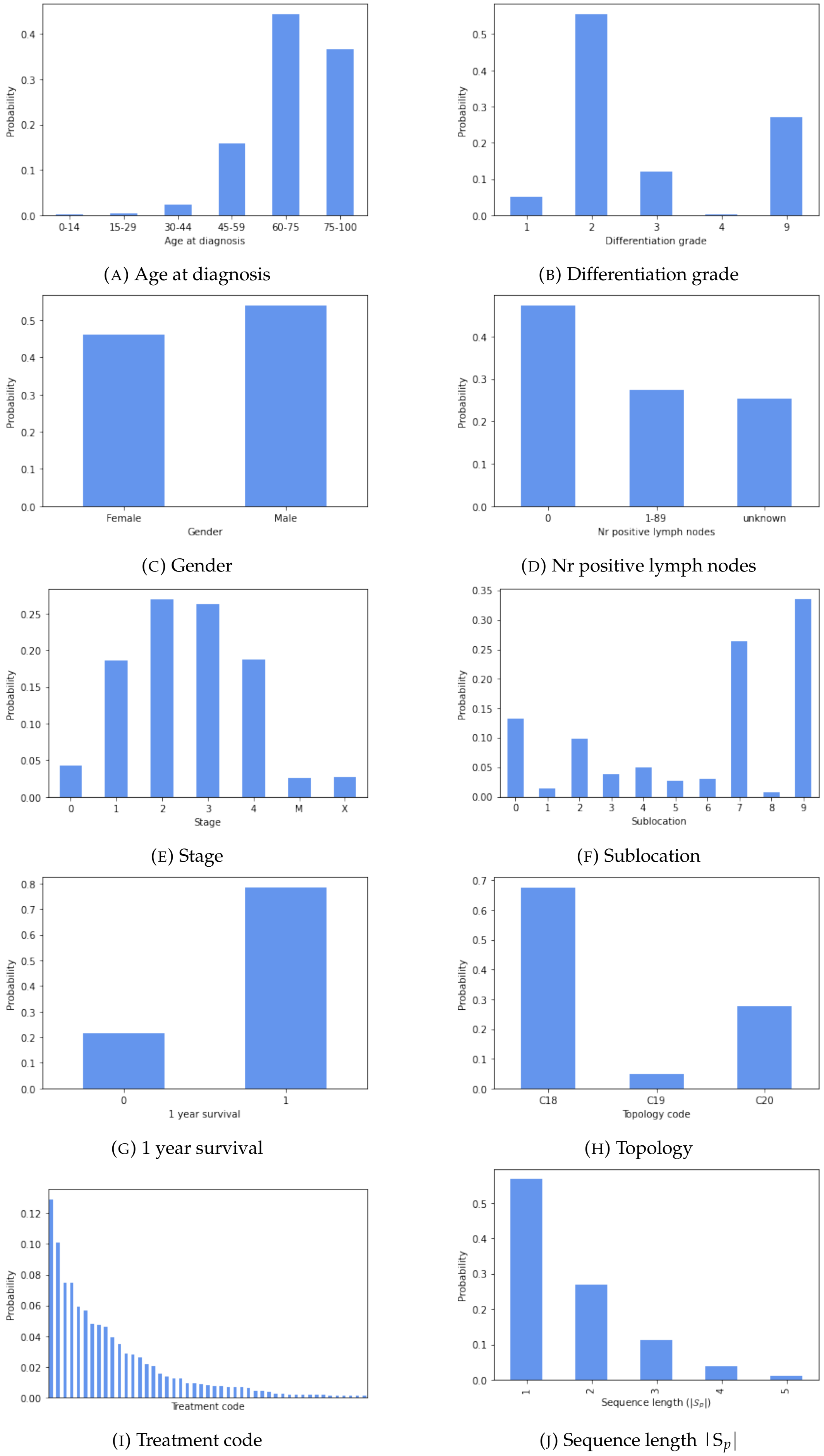
treatment sequences. We did not include treatment dates, as for 24.48% of all treatments in the data set the treatment date is unknown. This high percentage of missing treatment dates is likely due to the collection process of the data at IKNL, where registration practices might differ per hospital. Nonetheless, the data set provides information on the order of treatments, even those without a date, so that we can reliably construct chronologically ordered treatment sequences. Finally, treatment-related information such as treatment outcome or drug dosage is not included in this research, mainly based on the problem statement provided by IKNL and to reduce the complexity of the problem. In addition, treatment-related information is mostly collected from 2015 onwards only, causing it to be missing for many patients in the data set.

One of the aforementioned methods presented in Section 2.2.1 - PrivBayes - requires discrete features as input. In order to enable a fair comparison between generative methods, we discretize the two numerical features in our data set: *Age at diagnosis* and *Nr positive lymph nodes*, regardless of the generative method in place. Such discretization of numerical features is a common task in the field of data privacy literature (e.g., k-anonymity (Sweeney, 2002a, 2002b)) and can often be done based on domain knowledge (Zhang et al., 2017). While other methods for discretization can be considered, this is outside the scope of this research and we will focus on discretization based on domain knowledge. In particular, *Age at diagnosis* is discretized in 6 categories using age categories used on the informative website of cancer in the Netherlands[1] (age ranges: [0-14, 15-29, 30-44, 45-59, 60-74, >75]), and *Nr positive lymph nodes* is discretized into three categories: [0, 1-89, unknown]. Additionally, we reduce the dimensionality of the feature *Stage* by excluding its suffix (A, B, or C) as its 'broad' stage covers the main meaning of the feature.

3.2.4 Descriptives

Figure 3.3 shows the probability plots for all patient covariates in the final data set. The lower two subplots show probability plots of treatment-related features treatment code occurrence rate and sequence lengths respectively. It shows that most features have an imbalanced distribution, except for gender (slightly more male than female) (Figure3.3c). This class imbalance is specifically important to take into consideration for the feature 1-year survival (Figure 3.3g), as this is the target variable for one of the quality metrics proposed in Section 4.3.2. Furthermore, Figure 3.3a shows that most patients in the data set are older patients (age between 60-100 years old), as colorectal cancer is mostly a disease that affects people of middle age and beyond. Regarding the topology of the cancer, Figure 3.3h shows that colon cancer (C18) is most prevalent in our data set, followed by rectal cancer (C20). Rectosigmoid cancer (C19) is the least prevalent in our data set. Figure 3.3e shows the probability plot of cancer stage. It can be seen that stages 1-4 are most prevalent. This is due to the fact that stage 0 corresponds to an incidental finding, whereas stage M and X imply stage unknown or not enough information to provide stage respectively. Most

[1] https://www.kanker.nl/kankersoorten/dikkedarmkanker/wat-is/overlevingscijfers-dikkedarmkanker



(A) Age at diagnosis

(B) Differentiation grade

(C) Gender

(D) Nr positive lymph nodes

(E) Stage

(F) Sublocation

(G) 1 year survival

(H) Topology

(I) Treatment code

(J) Sequence length $|S_p|$

FIGURE 3.3: Probability plots of both static patient covariates and treatment-related features (treatment codes and sequence lengths)

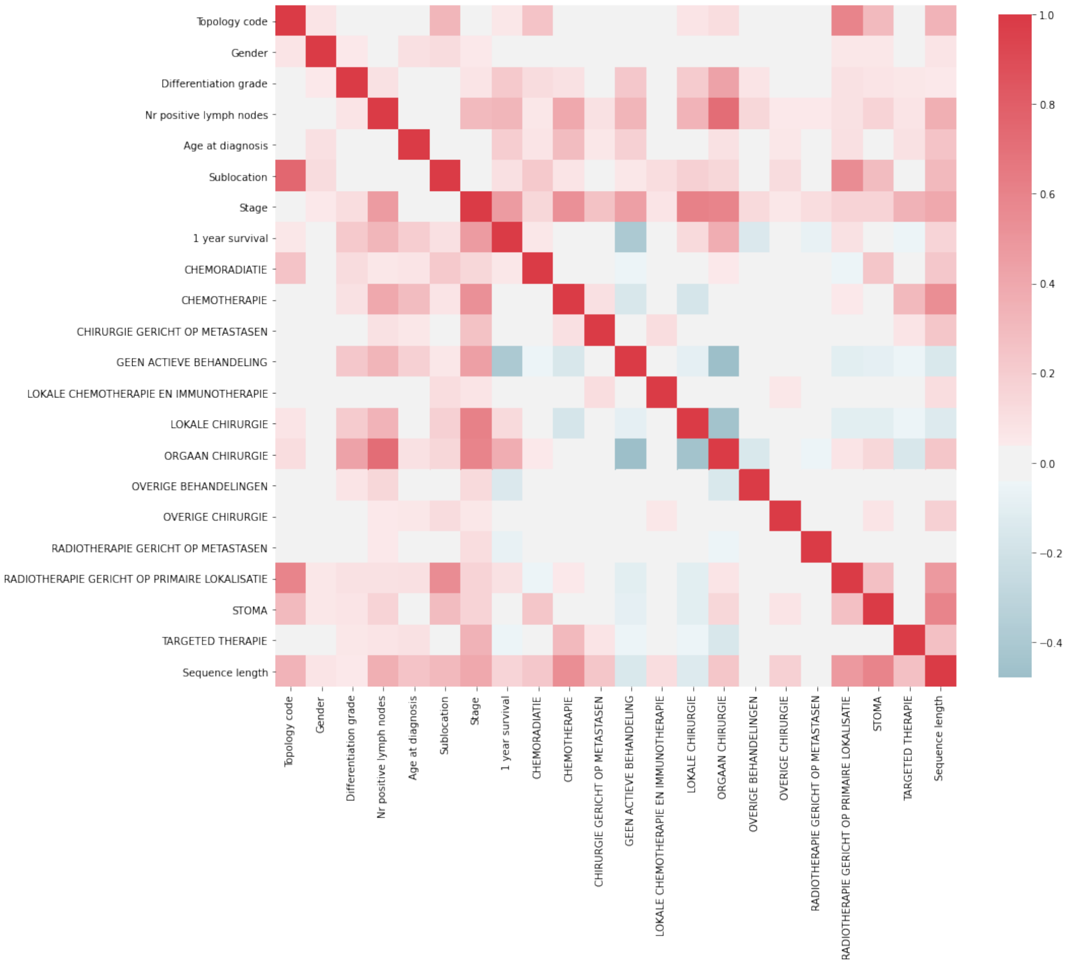


FIGURE 3.4: Association matrix of the preprocessed data set. Exact association values are omitted for confidentiality.

patients in our data set are diagnosed with stage 2 or stage 3 cancer. Concerning differentiation grade (Figure 3.3b), grade 2 is the most prevalent, which corresponds to moderately differentiated tumor tissue. The second-most prevalent category is 9, which implies an unknown differentiation grade. Then, for number of positive lymph nodes (Figure 3.3d), most patients have zero positive lymph nodes, where unknown and some (1-89) positive lymph nodes are less prevalent. Then, the most common sublocations (Figure 3.3f) are 9 and 7, where all the others occur considerably less often. Finally, for the treatment-related features, Figure 3.3i and Figure 3.3j show that treatment occurrence rates and treatment lengths are imbalanced.

Figure 3.4 shows the associations between (and within) patient covariates and derived treatment features. In Section 4.3.2, we thoroughly explain the methods used to construct the derived treatment features and calculations on the associations between features. In short, we extend the static patient covariates with a binarized treatment vector and sequence length for the patient. In the data set, each of the specific treatment codes can be mapped to a main group of lower cardinality. For example, the specific treatment "Rectal amputation / Abdominal Perineal Resection (APR)" belongs to the main group "Organ surgery", where the specific treatment "Capecitabine" belongs to the main group "Chemotherapy". The binarized treatment vector then has one attribute per main group, where the value of this main group is 1 if the patient encountered one or more treatment(s) belonging to this main group,

Code	Meaning	Main group	Occurrence (%)
132C18	Hemicolectomy / ileocecal resection	Organ surgery (ORGAAN CHIRURGIE)	13%
135C18	Low anterior resection	Organ surgery (ORGAAN CHIRURGIE)	10%
420000	Systemic chemotherapy	Chemotherapy (CHEMOTHERAPIE)	7%
133C18	Sigmoid resection	Organ surgery (ORGAAN CHIRURGIE)	7%
210000	External radiotherapy	Radiotherapy (RADIOTHERAPIE GERICHT OP PRIMAIRE LOKALISATIE)	6%
712000	Colostomy	Stoma (STOMA)	6%
121000	Endoscopic resection / polypectomy	Local surgery (LOKALE CHIRURGIE)	5%
100000	Surgery unspecified	Organ surgery (ORGAAN CHIRURGIE)	5%
000000	No treatment	No active treatment (GEEN ACTIEVE BEHANDELING)	5%
142C20	Rectal amputation / Abdominal Perineal Resection (APR)	Organ surgery (ORGAAN CHIRURGIE)	4%

TABLE 3.3: Top 10 most frequent treatment codes in the preprocessed data set, including their meaning, main group, and occurrence percentage. Occurrence percentages are rounded to the nearest percentage for confidentiality.

and zero otherwise.

We can see that most associations within the data set are not very strong. Some exceptions include for example the associations between topology code and sublocation, topology code and radiotherapy (RADIOTHERAPIE GERICHT OP PRIMAIRE LOKALISATIE), and positive lymph nodes and organ surgery (ORGAAN CHIRURGIE). From this association matrix, we also see that, as expected, relations between patient covariates and treatments exist. For example, the stronger association between Topology code and radiotherapy (RADIOTHERAPIE GERICHT OP PRIMAIRE LOKALISATIE) can be explained by the fact that radiotherapy is one of the standard therapies for patients with rectal (topology code C20) or rectosigmoid cancer (topology code C19), where it is much less frequently given to patients with colon cancer (topology code C18). Another example is the stronger association between Stage and local surgery (LOKALE CHIRURGIE), as local surgery is often only possible for low-stage tumours. Associations between binarized treatment features are limited, as most patients in the data set have received very few (or even only one) treatments. Regarding our target variable (1-year survival), the highest associations of this feature are with Stage, no treatment (GEEN ACTIEVE BEHANDELING), and organ surgery (ORGAAN CHIRURGIE).

Table 3.3 lists the top 10 most frequently occurring treatment codes along with their meaning, main group, and occurrence percentage in the preprocessed data set. We can observe that most frequently occurring treatments belong to the main group organ surgery (ORGAAN CHIRURGIE). Additionally, Table 3.4 specifies the top 10 most frequently occurring treatment sequences. Here, we observe that most frequently occurring sequences have length one, with two exceptions of length two. Both these sequences of length two include a surgery, one followed by chemotherapy and the other preceded by radiotherapy.

Sequence	Sequence length	Occurrence (%)
Hemicolectonmy / ileocecal resection	1	13%
No treatment	1	7%
Sigmoid resection	1	7%
Endoscopic resection / polypectomy	1	6%
Surgery unspecified	1	6%
Low anterior resection	1	5%
(Extended) Hemicolectomy right	1	2%
Hemicolectonmy / ileocecal resection → Systemic chemotherapy	2	2%
External radiotherapy → Low anterior resection	2	2%
Systemic chemotherapy	1	1%

TABLE 3.4: Top 10 most frequent treatment sequences in the preprocessed data set, including their sequence length and occurrence percentage. Occurrence percentages are rounded to the nearest percentage for confidentiality.

Chapter 4

Methodology

In this chapter, we describe the approach to answer the research question. First, the adaptions to and implementation details of the chosen generative methods for this thesis are described. Second, we elaborate upon metrics to evaluate the quality of the generated synthetic data sets. Finally, we introduce the experimental setup.

4.1 General notations

Throughout the methodology, several re-occurring elements related to the data set or quality metrics are introduced. In order to provide a comprehensive overview, the notation for these aspects is presented in Table 4.1.

Notation	Definition
P	The data set defined as a set of samples $p \in P$
c_p	Vector of static covariates for patient p
$\mathcal{S}_p$	Event sequence of patient p
$e_{i,p}$	Treatment code of the i-th event of patient p
$g(i,p)$	Main group of $e_{i,p}$
P_{train}	The part of the original data set P used for synthetic data generation
P_{test}	The holdout test set ($P_{test} \subset P$)
Z	A synthetic data set generated from P_{train}
F_p	A vector of patient covariates c_p and treatment sequence $\mathcal{S}_p$, such that every $e_{i,p}$ in $\mathcal{S}_p$ is added to c_p with respect to order of occurrences
l	Maximum event sequence length in data set P
H	The subset of patient covariates ($H \subset c$) that are known to influence (treatment) events based on domain knowledge
V_P	An extended static version of a data set P (concatenation of patient covariates c_p, binarized treatment vector indicating for each main group whether the patient had a treatment belonging to that main group, and sequence length $\lvert\mathcal{S}_p\rvert$ for each patient $p \in P$)
$\Psi_{a\mid b}$	A classification model trained on data set a and tested on data set b
α	AUC-ROC score

TABLE 4.1: General notation used in the methodology

4.2 Generative methods

4.2.1 Motivation for chosen generative methods

In this thesis, we evaluate three synthetic data generation methods for the generation of the synthetic version of the original patient data, including both static covariates and treatment sequences. We first elaborate upon the motivation of the choice for the three generative methods.

First of all, based on our literature review in Section 2.1, DoppelGANger is the only existing algorithm that satisfies our most important criterion: co-generation of covariates and sequential data. The method can be applied to the patient covariates and treatment sequences of colorectal cancer patients in the NCR with little modifications only. Additionally, GANs have emerged as the state-of-the-art for producing realistic tabular and time-series data. Compared to more traditional statistical modeling approaches to generate synthetic data that require strict probabilistic model assumptions, GANs are more flexible in their modeling procedure and can therefore often better represent (latent) relations between features (Georges-Filteau & Cirillo, 2020; Goncalves et al., 2020). For these reasons, we consider it straightforward to include DoppelGANger as one of the generative methods in this research.

Then, as elaborated upon in Section 2.1.3, if we can develop a method to jointly model patient covariates and health event sequences, we can deploy a wide range of generative algorithms designed for cross-sectional data on this jointly modeled data frame. We introduce a method for joint-modeling of patient covariates and treatment sequences in Section 4.2.2. However, given the limited time available in this thesis, we will not consider a wide range of existing generative methods for cross-sectional data. Instead, we focus on PrivBayes as a generative method for the jointly modeled patient covariates and treatment sequences for the following reasons. First of all, the other method deployed in this thesis - DoppelGANger - is a black-box method. PrivBayes is a more transparent method focusing on Bayesian networks, which allows for both user interpretation of the generation process as well as (with some slight adaptations to the existing algorithm) user input on network construction based on domain knowledge. As argued by Kaur et al. (2020), interpretability is crucial in health analytics applications and it is therefore interesting to compare a more transparent method (PrivBayes) with a black-box model capable of detecting latent relationships (DoppelGANger). Secondly, recent work by Kaur et al. (2020) has shown that, in contrast to the general consensus and results presented in Xu et al. (2019), Bayesian networks can outperform deep (GAN) methods in the context of cross-sectional health data. It is relevant to evaluate if this finding also holds for the co-generation patient covariates and treatment sequences, which can be considered a more complex use case from a data perspective. Finally, from a practical standpoint, IKNL is ultimately interested in generating synthetic data with differential privacy guarantees. PrivBayes provides a way to include differential privacy guarantees in their algorithm.

Finally, we include Marginal Synthesizer as a baseline algorithm for the co-generation of patient covariates and treatment sequences. We consider Marginal Synthesizer an appropriate baseline method, as it is relatively straightforward to hypothesize its expected behavior with regards to the quality metrics presented in Section 4.3. Herewith, Marginal Synthesizer serves as an additional check for the expected behavior and purposes of quality metrics. As it will be elaborated upon further in Section 4.2.4, Marginal Synthesizer is expected to score well on individual feature distributions, while it does not capture any relations between features in the synthetic data set and therefore serves as a lower bound on quality metrics involving relations between features.

4.2.2 PrivBayes for sequential data

In this section, we will describe an approach to utilize the PrivBayes algorithm (Zhang et al., 2017) described in Section 2.2.1 for generating the synthetic version of the original patient data including covariates and treatment sequences. In general, the event sequence is characterized as a relatively short sequence with irregularly sampled events, that depends on a subset of static covariates, later referred to as context attributes. As PrivBayes can only handle tabular data (i.e., one row per entity), we first present a way to preprocess and jointly model the covariates c and event sequence S into one cross-sectional data set (later referred to as sequential data pivoting). Then, we suggest a slight adaptation of the PrivBayes algorithm in order to handle the transformed cross-sectional data set, by utilizing both the temporal ordering of events and clinical domain knowledge on context attributes.

Sequential data pivoting

We propose the following workflow for joint modeling of the covariates c and event sequence $\mathcal{S}$ into one cross-sectional data set F and term it as *sequential data pivoting*. This procedure is shown visually in Figure 4.1. In short, the workflow entails creating a pivoted data set F as a combined vector of patient covariates c and treatment sequences $\mathcal{S}$, such that every event $e_{i,p}$ (i.e., treatment) in $\mathcal{S}_p$ is added to the end of c_p with respect to order of occurrences. A detailed description of the workflow is as follows:

1. Define the maximum number of events (i.e., maximum sequence length) $|\mathcal{S}_p|$ for a patient p in P. This maximum sequence length is obtained by trimming the sequence length distribution of treatments of patients during data preprocessing (Section 3.2.3, step 4). We denote the maximum sequence length as $l = Q_\gamma(|\mathcal{S}|)$, where γ is the trimming percentage. Thus, in our data set, $\gamma = 0.99$ and $l = Q_{0.99}(|\mathcal{S}|) = 5$.

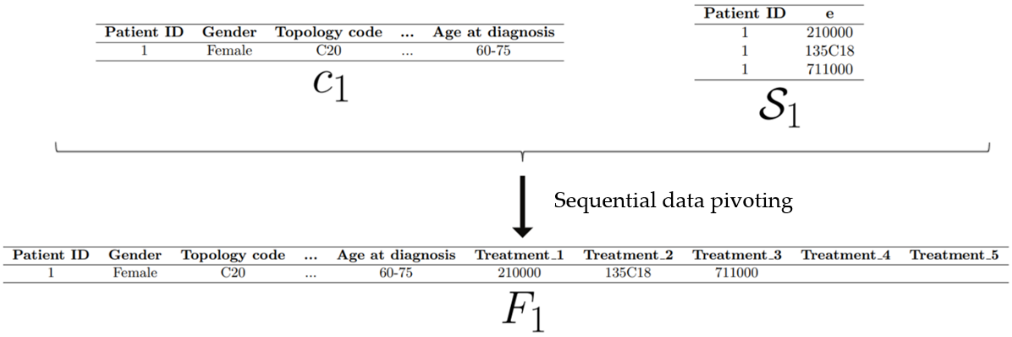
2. Append l attributes to the static covariates part of the data set c, one per event: $Treatment_i$ $(i \in [1, l])$.

3. For each patient p, fill the $Treatment_i$ attributes for all $e_{i,p}(i \in [1, |\mathcal{S}_p|])$ in S_p. Additionally, leave all $Treatment_i$ attributes $i \in (|\mathcal{S}_p|, l]$ empty.

The output of the sequential data pivoting is a vector F_p consisting of both static covariates c_p and a pivoted representation of sequential information for each patient p. As mentioned, we denote this pivoted, cross-sectional data set as F.

We acknowledge that this sequential data pivoting is best suited for data where the event sequences are relatively short, i.e., l is a low number, in order to prevent the number of attributes to append to the patient covariates from becoming too large. An increased number of attributes d in the data set will influence the effectivity of the differentially private version of PrivBayes (i.e., $\varepsilon \neq \infty$), by increasing the level of noise to be added at a fixed privacy level. In our use-case scenario, colorectal cancer treatment sequences have the characteristic of being relatively short. This also relates to our application scenario in which only treatments up to 12 months after diagnosis are collected. Thus, in other application domains, this sequential data pivoting may be inappropriate for several reasons. The characteristic of short treatment sequences within the first 12 months after diagnosis holds for most other cancer types, making our approach generalizable within - at least - the current domain.

Figure 4.1 shows a visual representation of sequential data pivoting applied to the static covariates c_1 and the event sequence $\mathcal{S}_1$ of a fictionary patient 1, with output the pivoted vector F_1.



Patient ID	Gender	Topology code	...	Age at diagnosis
1	Female	C20	...	60-75

c_1

Patient ID	e
1	210000
1	135C18
1	711000

$\mathcal{S}_1$

Patient ID	Gender	Topology code	...	Age at diagnosis	Treatment_1	Treatment_2	Treatment_3	Treatment_4	Treatment_5
1	Female	C20	...	60-75	210000	135C18	711000		

F_1

FIGURE 4.1: Visual representation of sequential data pivoting applied to a fictionary patient 1

Adapted PrivBayes algorithm for sequential data

Now that we have proposed a way to transform the static covariates c and the event sequences $\mathcal{S}$ into one cross-sectional data frame F, we can naively apply PrivBayes as proposed by Zhang et al. (2017) on this data set to generate a synthetic version. However, we propose to extend the PrivBayes method by leveraging our knowledge on the temporal ordering of treatment sequences and domain knowledge on context attributes (H) influencing treatment sequences. This possibility for a user to enter a (partly) ready-made network of hypothesized connections into the PrivBayes algorithm was also recognized by prior work as an experiment worthwhile studying (Perkonoja, 2020). The adapted PrivBayes procedure for the pivoted data set F is as follows:

1. Execute step 1 (GreedyBayes) of the original PrivBayes algorithm on $\mathcal{A}_c$ - the set of attributes in the patient covariates part of F, i.e., c - which outputs a Bayesian network $\mathcal{N}_c$.
2. Model the AP pairs for all attributes in $\mathcal{A}_\mathcal{S}$ - the set of attributes in the pivoted representation of sequential information in F (i.e., *Treatment* attributes) - based the temporal ordering of treatments sequences and domain knowledge on context attributes (H) influencing treatment events to define the sequential part of the network $\mathcal{N}_\mathcal{S}$.
3. Append $\mathcal{N}_\mathcal{S}$ to $\mathcal{N}_c$ to obtain the Bayesian network $\mathcal{N}$ over all attributes $\mathcal{A}$ in F.
4. Execute steps 2 (conditional distribution generation) and 3 (sampling) of the original PrivBayes algorithm using $\mathcal{N}$.

Step 2: Sequential network ($\mathcal{N}_\mathcal{S}$) definition

The main novelty in our adaptions to the original PrivBayes method by Zhang et al. (2017) is in step 2 of the procedure. In this paragraph, we explain step 2 of the adapted PrivBayes procedure for the pivoted data set F in more detail.

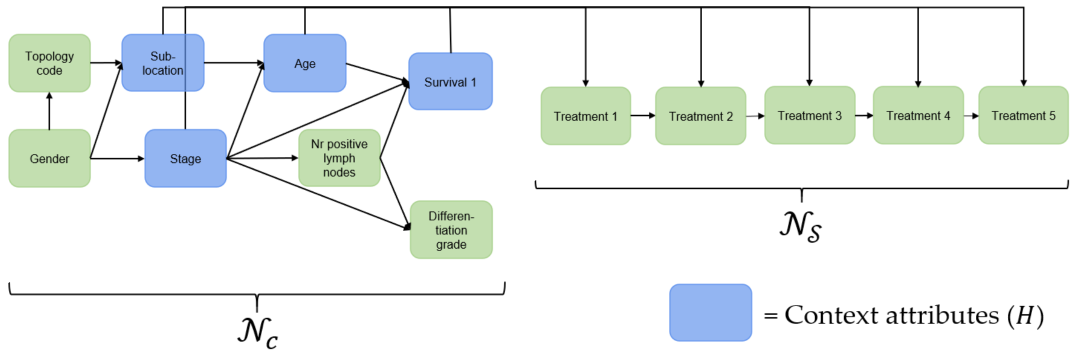


FIGURE 4.2: Bayesian network $\mathcal{N}$ including its components $\mathcal{N}_c$ and $\mathcal{N}_{\mathcal{S}}$

We let the user manually define the parent set of each $Treatment_i$ ($i \in [1,l]$) attribute in $\mathcal{A}_{\mathcal{S}}$, but provide a structured way to do so. First, we let the user define a set of context attributes $H \subset c$, that are known to influence the event sequence from domain knowledge (e.g., clinical knowledge on treatment choices). We add H to the parent set Π_i of each $Treatment_i$ ($i \in [1,l]$ attribute in $\mathcal{A}_{\mathcal{S}}$.

Second, we take into account the temporal ordering of the events in the sequence while defining $\mathcal{N}_{\mathcal{S}}$. In general, we take into consideration the temporal ordering of treatment events while constructing the parent set of all *Treatment* attributes. That is, we condition the current treatment on the previous treatment. In other words, we add $Treatment_{i-1}$ to the parent set Π_i of $Treatment_i$. We only consider a time window size of 1 in the construction of the sequential network (i.e., we assign the previous treatment only to the parent set of the current treatment), as (patterns within) sequences are generally short and increasing the time window will intuitively give less privacy. Furthermore, we model the *Treatment* attributes in temporal order, so that $Treatment_i$ attributes are added to the network in increasing order of $i \in [1,l]$. That is, we first add $Treatment_1$ attribute to the network, then $Treatment_2$ attribute, and so on up until the $Treatment_l$ attribute. This ordered addition of treatment attributes to the network ensures that $\mathcal{N}_{\mathcal{S}}$ is a directed acyclic graph (DAG) and hence the entire network $\mathcal{N}$ remains a DAG.

To sum up, the parent set Π of each $Treatment_i$ attribute in $\mathcal{A}_{\mathcal{S}}$ is set to the union of context attributes H and the previous treatment attribute $Treatment_{i-1}$. Mathematically, $\Pi_i = H \cup \{Treatment_{i-1}\}$.

An example of a Bayesian network $\mathcal{N}$, including its components $\mathcal{N}_c$ and $\mathcal{N}_{\mathcal{S}}$ is shown in Figure 4.2. Boxes and arrows represent attributes and conditional dependencies respectively, where each attribute is conditioned on its parent set. In this example, the attributes 'Sublocation', 'Stage', 'Age' and 'Survival 1' are the set of context attributes H and thus included in the parent sets Π_i of each $Treatment_i$ ($i \in [1,l]$) attribute.

After we generate data using PrivBayes Sequential, we revert the sequential data pivoting so that we obtain c_p and $\mathcal{S}_p$ separately for each synthetic patient p.

Parameters

Recall from Section 2.2.1 that the PrivBayes algorithm includes two internal parameters: the budget allocation parameter, β (partitions the total privacy budget ε into ε_1

(the privacy used for network learning) and ε_2 (the privacy budget used for distribution learning)) and the usefulness of noisy joint distributions, θ (the ratio of the average scale of information to the average scale of noise in the joint distributions). Based on an empirical evaluation, Zhang et al. (2017) found that an appropriate value for β should be in the range of $[0.2, 0.5]$ and the authors suggest a default value of $\beta = 0.3$. In addition, Zhang et al. (2017) find that PrivBayes is robust against small changes in θ, where the authors define suitable values in the range of $[3, 6]$ and a default value of $\theta = 4$. Based on the extensive empirical evaluation of the authors, we decided to adopt the default values of $\beta = 0.3$ and $\theta = 4$ - where relevant - in this thesis.

4.2.3 DoppelGANger

In this section, we describe data preparation steps taken and implementation details for our use case of the original DoppelGANger (DG) architecture by Lin et al. (2020) described in Section 2.2.2.

Data preparation

In order to prepare our data set for the DG algorithm, several steps were taken. For the event sequences, we transformed the sequences to an array termed `data_sequence` of size [(number of training samples) x (maximum length) x (total dimension of treatment codes)]. The treatment codes were one-hot encoded. As we have 47 unique treatment codes, this leads to a dimensionality of 47. Given that our training set consists of 191,913 patients (elaborated upon in Section 4.4), recall from Section 3.2.2 that the maximum length (l) of treatment sequences in our data set after preprocessing is 5, and as just mentioned the dimensionality of the one-hot encoded treatment codes is 47, the `data_sequence` array is of size $[191913, 5, 47]$. All treatment sequences with lengths shorter than 5 were padded to length 5 using zero padding.

Patient covariates are transformed to an array termed `data_covariates` of size [(number of training samples) x (total dimension of attributes)]. Again, all categorical covariates (which equals all covariates in our case) are stored by one-hot encoding. The dimensionality of the one-hot encoded attributes equals the sum of all cardinalities of the covariates, which is 38 in our case. Thus, the `data_covariates` array is of size $[191913, 38]$.

Finally, related to the generation flag described in Section 2.2.2, DG requires as input the generation flags for the training data in an array termed `data_generation_flag` of size [(number of training samples) x (maximum length)], i.e., [(191913 x 5)] in our case. The value of this generation flag for a patient p is 1 if the there is a treatment for patient p at this time step, and 0 otherwise. For example, a patient p with treatment sequence length $|\mathcal{S}_p| = 3$ has generation flag $[1, 1, 1, 0, 0]$.

Implementation details

In this section, we describe the implementation details of DG described in Section 2.2.2 for our research.

Auxiliary discriminator In their paper, Lin et al. (2020) propose the incorporation of an auxiliary discriminator to improve the quality of generated covariates, especially needed when the average sequence length is long. As our sequences are short, particularly compared to the time series lengths considered by Lin et al. (2020), we do not include an auxiliary discriminator and use one discriminator for the combination of patient covariates and the event sequence only (in line with the architecture presented in Figure 2.2).

Mode collapse After conducting some initial experiments with the DG architecture on our data, we noticed a well-known problem in GANs occurred on the treatment sequences: *mode collapse*. Mode collapse is the situation in which the GAN generates data that only covers a few classes of data samples, rather than producing diverse synthetic outputs. In particular, in our preliminary experiments, only about 50% of all treatment codes present in the original data set were present in the synthetic data set generated by the original DG architecture and parameters. In their paper, Lin et al. (2020) propose a technique called *autonormalization* to cope with mode collapse. However, this approach is only relevant to continuous data, where our data is categorical. Interestingly, however, in their implementation, Lin et al. (2020) allow the user to include the PacGAN (Lin et al., 2018) framework within the DG architecture. PacGAN effectively handles mode collapse by offering a simple modification to the original GAN architecture in which the discriminator classifies individual samples as either real or synthetic. Instead, the main idea of PacGAN is to modify the discriminator to make decisions based on multiple samples from the same class (e.g., 5 samples in each pac means that the discriminator is given 5 samples of the real data or 5 samples of the synthetic data as input) and then classifies this pac of samples as either real or synthetic. After some experimentation, we set the number of samples in each pac to 10, a value also used by prior research on cross-sectional data (Xu et al., 2019).

GAN loss function DG adopts Wasserstein loss (Arjovsky et al., 2017) for improving training stability and alleviating mode collapse. Lin et al. (2020) empirically find that DG with Wasserstein loss is better than DG with original GAN loss for generating categorical variables. As our research is centered around categorical variables only, it is straightforward to use the implementation of Lin et al. (2020) and use DG with Wasserstein loss.

Batch generation Lin et al. (2020) find that LSTM generators struggle to capture temporal correlations for long time series. The reason for this is that for long time series, LSTMs take too many passes to generate the entire sample. The more passes are taken, the more temporal correlations the LSTM tends to forget. Therefore, the authors propose batch generation, in which the LSTM generates S records at each pass, instead of generating one time step per pass. Empirically, they find that setting S so that the number of steps for an RNN to take is maximally 50 gives good results. However, as our treatment sequences have a length of maximally 5, using batch generation is unnecessary. Therefore, we can simply use the original LSTM framework in which one time step is generated per LSTM pass (i.e., $S = 1$).

Other architectural details The most important architectural details and parameters are summarized in Table 4.2. Softmax layer is applied for the categorical covariates and events output. In addition, Lin et al. (2020) mention that GAN convergence required up to 200,000 batches on their data sets. When translating that to our data set P_train with 191,913 instances and a batch size of 100, the proposed number of epochs equals $(200000 * 100)/191913 = 105$ (rounded up to the nearest integer). For

Parameter	Value
Covariates generator	MLP with 2 hidden layers and 100 units in each layer
Sequence generator	1-layer LSTM with 100 units
Discriminator	MLP with 4 hidden layers and 200 units in each layer
Samples per pac	10
Autonormalization	False
Batched generation parameter (S)	1
Loss function	Wasserstein distance
Gradient penalty weight	10
Optimizer	Adam optimizer
Learning rate	0.001
Batch size	100
Epochs	105

TABLE 4.2: Parameters used for training DoppelGANger. Any parameter not listed in this table was left as proposed by Lin et al. (2020).

other parameters not present in the aforementioned table, default parameters proposed by Lin et al. (2020) were used.

4.2.4 Marginal Synthesizer for sequential data

In order for Marginal Synthesizer to co-generate patient covariates and treatment sequences, we use the pivoted representation of our data set F (explained in Section 4.2.2) as input for the algorithm. Due to the fact that Marginal Synthesizer does not take relations between features into account, it might happen that we sample an 'empty' treatment (i.e., no treatment) for a specific treatment column (e.g., *Treatment*$_2$), and sample a treatment for one of the next treatment columns (e.g., *Treatment*$_3$). We post-process the synthetic data such that the first 'empty' treatment sampled by Marginal Synthesizer marks the end of the treatment sequence for that patient (and thus we handle the next treatments as 'empty' regardless of the value sampled by Marginal Synthesizer).

4.3 Quality evaluation

This section describes metrics to evaluate the quality of the generated synthetic data. As quantitative measures provide more convincing evidence of synthetic data quality compared to qualitative evaluation metrics (Georges-Filteau & Cirillo, 2020), we primarily consider quantitative evaluation approaches. Nonetheless, some visual additions to these quantitative evaluation metrics are presented, as these are generally considered more interpretable for non-expert (medical) users. We examine quantitative evaluation measures from all three categories defined by Georges-Filteau and Cirillo (2020): data set distribution metrics, statistical metrics, and utility metrics. From a domain perspective, we consider three important evaluation criteria of the generated synthetic data set:

1. Univariate statistics of patient covariates and (static representations of) treatment sequences

2. Relations between and within patient covariates and (static representations of) treatment sequences

3. Temporal aspects of and patterns within treatment sequences

In the following sections, we will formulate the evaluation metrics used in this thesis and evaluate what criterion of synthetic data set quality they cover. Table 4.3 presents an overview of the metrics detailed below.

Metric	Summary	Evaluation criterion
Metric 1: Jensen-Shannon (JS) Distance patient covariates	Average JS distance between patient covariates distributions in original and synthetic data set	1: Univariate statistics
Metric 2: RMSE treatment occurrences	RMSE between treatment code occurrences in original and synthetic data set	1: Univariate statistics
Metric 3: Support Coverage treatment occrurences	Ratio of unique treatment codes (domain size of treatment codes) in synthetic and original data set	1: Univariate statistics
Metric 4: JS Distance sequence lengths	JS distance between treatment sequence length distribution in original and synthetic data set	1: Univariate statistics
Metric 5: Associations (PCD)	Pair-wise Correlation Difference (Frobenius norm) between association matrices of the original and synthetic data set on an extended version of the static data sets V including both patient covariates and static information on treatment sequences	2: Relations
Metric 6: TB-TOH	Train survival prediction model on (extended static versions V as in Metric 5 of) original data set used for training and synthetic data set, test both on original holdout set and compare prediction performance (AUC-ROC)	2: Relations
Metric 7: Jaccard similarity sequential pattern mining	Jaccard similarity between sequential patterns mined from treatment sequences in the original and synthetic data set	3: Temporal aspects

TABLE 4.3: Concise overview of metrics used to evaluate synthetic data set quality

4.3.1 Univariate metrics

Patient covariates

Metric 1: Covariates distributions (JS Distance)
One basic measure of synthetic data set quality is centered around comparing the statistical distributions of the features in the data set. A simplistic way of comparing statistical distributions of features is by plotting the distributions of the features in the original and the synthetic data set and comparing them visually. In our data set, all features are discretized and hence the probability distributions can be plotted as discrete probability distribution plots. However, this approach might not be extendable to data sets with many features and its feasibility is limited in cases where multiple synthetic data sets are generated. A quantitative way of evaluating the statistical distribution resemblance of the synthetic data set is by computing the distance between the probability plot of a feature in the original and synthetic data set. The distance can be evaluated per feature, or an aggregate measure over all features can be shown. In this research, we use an aggregate distance between the original and synthetic data set by computing the mean of all distances between each patient covariate (i.e., each feature in c).

As mentioned in Section 2.3, different distance measures are used in the literature. In this research, we chose to adopt the Jensen-Shannon (JS) distance. The JS distance is computed as the square root of the JS divergence, which in turn is a symmetric and smoothed version of the Kullback-Leibler (KL) divergence. Here, we prefer the use of the JS distance over the KL divergence, as the KL divergence requires absolute continuity in the probability distributions it compares. In our use case of comparison of original and synthetic data sets, it may happen that a (minority) category is not represented in the synthetic data set. In this case, we would not be able to compute the KL divergence, while the JS distance can be computed.

When the probability distribution of a feature in the synthetic data set is identical to the probability distribution of that feature in the original data set, the JS distance for that feature is 0. In the case of our aggregate JS distance metric, a value of 0 implies that the probability distributions of all patient covariates in the synthetic data set are identical to their respective probability distributions in the original data set. Thus, the lower the value of the JS distance, the better the quality of the synthetic data set in this perspective.

Given that the JS distance evaluates univariate statistical distributions of patient covariates in the data set, the JS distance metric falls under evaluation criterion 1.

Sequential data statistics

In order to get a first impression of the statistical quality of the treatment sequences, we introduce three measures that show the similarity between the original and synthetic treatment sequences. These measures do not take the order of treatments, i.e. sequential information, into account and thus focus on evaluation criterion 1.

Metric 2: Treatment occurrences (RMSE)
First of all, we look at the general occurrences of treatment codes in the entire data set. This metric is comparable to the dimension-wise probability metric suggested by Choi et al. (2017). For each unique treatment code x (in our data set, 47 unique treatment codes exist), we calculate its probability of occurrence in a data set D $p_D(x)$ as shown in Equation 4.1.

$$p_D(x) = \sum_{p=1}^{|D|} \sum_{i=1}^{|\mathcal{S}_p|} [e_{i,p} = x] \tag{4.1}$$

Then, we can quantitatively compare the occurrence probabilities of each treatment code $x \in dom(e)$. More specifically, we compute the Root Mean Square Error (RMSE) between $p_{P_{train}}(x)$ and $p_Z(x)$ for each $x \in dom(e)$. Formally, the formula for computing the RMSE is shown in Equation 4.3. The lower the RMSE, the more similar the occurrence probabilities of the treatment codes in the original and synthetic data set, and hence the better the quality of the synthetic data set.

$$RMSE(P_{train}, Z) = \sqrt{\sum_{k=1}^{|dom(e)|} \frac{(p_Z(x_k) - p_{P_{train}}(x_k))^2}{|dom(e)|}} \tag{4.2}$$

Next to this quantitative RMSE score, we can visually compare $p_{P_{train}}(x)$ and $p_Z(x)$ for each treatment code $x \in dom(e)$.

Metric 3: Treatment codes covered (Support Coverage)
As mentioned in Section 2.2.2, mode collapse is a known issue for some generative methods, including GANs. To evaluate if the synthetic data set covers all treatment codes in the original data set, we use the support coverage metric (Goncalves et al., 2020). The support coverage metric measures how much of the treatment code

variable's support (i.e., the domain size of the variable) in the original data is covered in the synthetic data. Mathematically, the support coverage is defined:

$$Support\ Coverage(P_{train}, Z) = \frac{|dom(e)_{P_{train}}|}{|dom(e)_{Z}|} \tag{4.3}$$

where $dom(e)_{P_{train}}$ and $dom(e)_Z$ are the domain of the treatment codes in the original (P_{train}) and synthetic (Z) data set, respectively. A support coverage equal to 1 means that all the treatment codes present in the original data set are represented in the synthetic data set. Hence, the higher the value for the support coverage metric, the better the quality of the synthetic data set in this respect. We recognize that this metric conveys only a limited indication of synthetic data set quality, and it should mostly be used as a necessary condition for useful synthetic data.

Metric 4: Sequence lengths (JS Distance)
Third, we compare the distribution of treatment sequence lengths $|\mathcal{S}|$ in the original and synthetic data set. We can visually compare the lengths of treatment sequences $|\mathcal{S}|$ in P_{train} and Z using a grouped bar chart. For a quantitative comparison, we use the Jensen-Shannon (JS) distance - as described in Metric 1 (paragraph 4.3.1)) - between the sequence length distributions in the original versus the synthetic data set. Again, like Metric 3, this metric conveys a limited indication of synthetic data set quality and should therefore also mostly be used as a necessary condition.

4.3.2 Relational metrics

Associations

Metric 5: Associations (PCD)
Where the first four metrics focus on individual feature distributions or flat data set characteristics only, correlations matrices can be used to evaluate more extensive aspects of synthetic data set quality in terms of the extent to which the relations between features are retained in the synthetic data. Two aspects of our specific use case need to be taken into account to compute the correlation matrices in our data set: (1) the existence of both categorical and binary variables, and (2) the combination of static and sequential data, where computing correlations between the static and sequential data is not straightforward.

Related to the existence of both categorical and binary variables in the data set, three different functions are applied to compute correlations between pairs of variables, depending on the data types of the features. We will further refer to correlations as *associations*. Note that binary variables in the data set are treated as continuous variables for the calculation of associations. The three methods used are as follows.

Continuous/binary - Continuous/binary: Pearson correlation
In the case of a pair of continuous (or in our case, binary) features, Pearson's correlation (Pearson's R) is used. Pearson's R is a symmetric correlation function that quantifies the strength of a linear association between two features. Its values range from -1 to 1, where a score of 1 implies a perfect positive correlation between the features and a score of -1 entails a perfect negative correlation. A score of 0 implies that there exists no correlation between the features.

Patient ID	CHEMOTHERAPIE	CHEMORADIATIE	ORGAAN CHIRURGIE	...	TARGETED THERAPIE
1	1	0	1	...	0

TABLE 4.4: Example binarized treatment vector for an fictionary patient 1

Continuous/binary - Categorical: Correlation ratio

In the case of a pair of features, including one continuous (binary) and one categorical feature, we use a specific measurement of association: Correlation ratio (Fisher, 1992). Intuitively, the correlation ratio assesses, given a continuous (binary) value, how certain we are which category it is associated with. The range of the correlation ratio is [0,1], where 0 means that a category cannot be determined by a continuous value, whereas a score of 1 implies that we can determine the category based on a numeric variable with absolute certainty. The correlation metric is non-symmetric.

Categorical - Categorical: Theil's U

In the case of a pair of categorical features, Theil's U and Cramer's V are two commonly used methods to assess the associations between two categorical features. Cramer's V is a symmetric measure that captures an overall evaluation of the strength of association. Theil's U (the uncertainty coefficient) (Shannon, 1948) is an asymmetric measure based on the conditional entropy between two variables. Given the ability of Theil's U to capture asymmetry between feature associations, it is the preferred metric for this research as it gives a more detailed representation of associations between features. Intuitively, it measures how well we can predict the value for one variable, given the value for the other variable. Theil's U ranges from 0 to 1, where 0 corresponds to the situation in which one feature provides no information on the value of another feature, and a score of 1 implies that one feature can perfectly predict (i.e., provides full information about) the other feature.

Secondly, for the latter aspect relating to the data set consisting of both static and sequential data, we use a static representation of the treatment sequences with which we extend the static data set. This allows us to compute associations both within and between static and sequential data. Our procedure is as follows. To reduce dimensionality and allow for a visual interpretation of the association matrix, we assign each treatment $e_{i,p}$ to its main group $g(i, p)$. For example, consider a fictionary patient 1 with $\mathcal{S}_1 = $ ('Sigmoidresectie', 'Capox'). Mapping these treatments to their main groups corresponds with ('ORGAAN CHIRURGIE', 'CHEMOTHERAPIE'). Then, for each patient, we create a binarized treatment vector with one attribute per main group (13 in total in our case). The value of this attribute is either zero or one, where one represents that the patient has undergone one or more treatment(s) belonging to this main group, and zero means that the patient has not encountered a treatment belonging to this main group. An example of this binarized treatment vector (for the fictionary patient 1 discussed before) is shown in Table 4.4. This example shows that the patient has undergone treatments belonging to the main groups CHEMOTHERAPIE and ORGAAN CHIRURGIE. Furthermore, we consider the sequence length $|\mathcal{S}_p|$ (i.e., number of treatments a patient received) as an additional static feature derived from the treatment sequence. Finally, the static covariates c_p, the binarized treatment vector and the sequence length $|\mathcal{S}_p|$ for each patient p are concatenated, leading to the extended static version V_P of a data set P.

We then use V_{Ptrain} and V_Z to compute the association matrices $M_{V_{P_{train}}}$ and M_{V_Z} of the original and synthetic data set respectively. These association matrices can be compared visually. However, to quantitatively assess the distance between the association matrices of the original and synthetic data, we use the pair-wise correlation difference (PCD) (Goncalves et al., 2020). The PCD metric calculates the Frobenius norm of the two matrices, which is the sum of Euclidean distances of every $m_{V_{P_{train}}ij}$ in $M_{V_{P_{train}}}$ and m_{V_Zij} in M_{V_Z}. The formula for calculating the PCD is shown in Equation 4.4.

$$PCD\left(M_{V_{P_{train}}}, M_{V_Z}\right) = \sqrt{\sum_{i=1}^{d}\sum_{j=1}^{d}\left(m_{V_{P_{train}}ij} - m_{V_Zij}\right)^2} \tag{4.4}$$

where d is the number of features in the extended data set V.

The PCD evaluates the relations between and within patient covariates and a static representation of treatment sequences in terms of association values and therefore falls under evaluation criterion 2. The smaller the PCD, the more similar the associations between and within patient covariates and derived treatment features, and hence the better the quality of the synthetic data set. A PCD of 0 means that the association matrices of the extended static versions of the original and synthetic data set are identical and thus all relations between features are perfectly kept. However, it is important to note that the exact value of the PCD is dependent upon the existing associations present in the original data set. For example, if the original data set has only a few high associations between features, the PCD might still be relatively low if the synthetic data set captures no associations at all. Although we can interpret the PCD score of various synthetic data sets (generated by different synthetic data generation algorithms) as we compare all synthetic data sets to the same original data set, we cannot discriminate between "good" or "bad" PCD scores in an absolute way. Therefore, a visual comparison will give meaningful additional information on the quality of the synthetic data set and we stress the importance of considering both a quantitative evaluation of the associations through PCD as well as a visual evaluation of the computed association matrices.

Machine learning: Train on Both, Test on Original Holdout set (TB-TOH)

Metric 6: 1 year survival prediction (TB-TOH)

As mentioned in Section 2.3, while lacking the interpretability of statistical measures, utility metrics often provide the strongest evidence of data realism. Section 2.3 describes that most related studies perform a prediction task considered appropriate for the data set at hand, but some prior works consider multiple target features or in the extreme case train prediction models for each feature in the data set. However, due to time restrictions in this research, we decided to focus on one specific prediction task. It is important that the target variable in this prediction task is of clinical significance to the data set being considered (Mendelevitch & Lesh, 2021). In (colorectal) cancer research, one prevalent prediction task is survival prediction (e.g. Wang et al., 2019). Survival prediction can be considered as a regression task (i.e., predicting the exact survival time in for example days or months) or a binary classification task (i.e., survival time less/more than a specific threshold, for example, 5 years). For this thesis, we took a binary classification approach towards survival prediction. More specifically, we predict whether or not a patient survived more than one year after diagnosis. While five-year survival might be a more prevalent task in colorectal cancer research, our data set only includes treatment information up to one year after diagnosis. Therefore, 1-year survival is considered a more appropriate target variable.

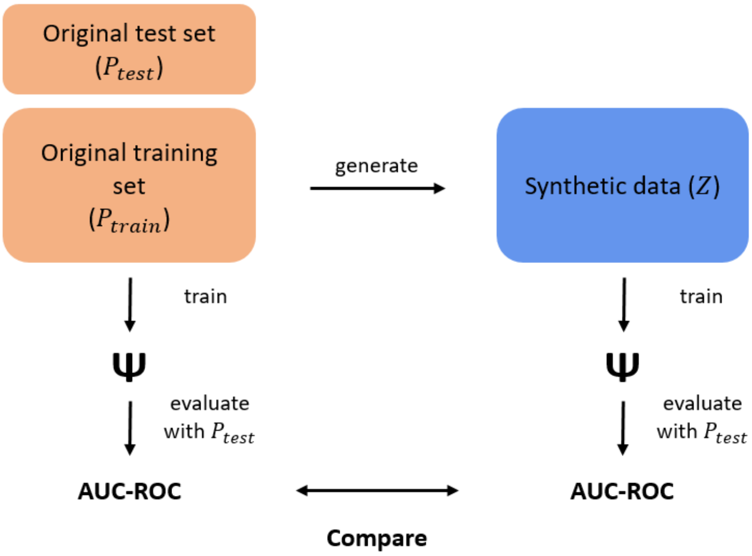


FIGURE 4.3: Schematic view of the TB-TOH metric

Additionally, as illustrated in Section 2.3, several approaches towards comparing the performance of predictors on real and synthetic data sets exist. As the extension of TSTR (TB-TOH) is considered the most interesting evaluation (Esteban et al., 2017), we will employ this metric for our research. Thus, to conclude, our evaluation metric will consist of a TB-TOH evaluation of a 1-year survival prediction task.

In order to include both patient covariates as well as treatment data as input features for the classification task, we use the extended static version V (described in more detail in the paragraph related to Metric 5 (Section 4.3.2)) of the data sets for this classification task. In short, this extended version of the data set includes patient covariates, a binarized treatment vector per main group, and the sequence length $|\mathcal{S}_p|$ for each patient p.

In the TB-TOH method, one model is trained on the synthetic data set (V_Z) and one model is trained on the part of the original data set used for training ($V_{P_{train}}$). Both models are evaluated on the original test set ($V_{P_{test}}$). A schematic representation of the TB-TOH method is shown in Figure 4.3. Recall from Section 3.2.4 that the feature 1-year survival is imbalanced in the original data set, and hence the prediction task is an imbalanced classification task. In this case, accuracy is not considered an appropriate evaluation metric. Instead, we use the Area Under the ROC Curve (AUC-ROC) as our evaluation metric, as it is an appropriate metric for binary, imbalanced classification tasks. Then, the TB-TOH score is computed as the ratio between the AUC-ROC score on the original test set of the classifier trained on the synthetic data set versus the classifier trained on the original data set used for training. Formally, the TB-TOH metric is described in Equation 4.5.

$$TB\text{-}TOH\left(V_{P_{train}}, V_Z | V_{P_{test}}\right) = \frac{\alpha(\Psi_{V_Z|V_{P_{test}}})}{\alpha(\Psi_{V_{P_{train}}|V_{P_{test}}})} \tag{4.5}$$

In case the classifier trained on the synthetic data set performs identically on the independent test set in terms of AUC-ROC score compared to the classifier trained on the original data used for training (i.e., $\Psi_{V_Z|V_{P_{test}}} = \Psi_{V_{P_{train}}|V_{P_{test}}}$), the TB-TOH score equals exactly one. Hence, the closer the TB-TOH score is to one, the higher the quality of the synthetic data. As a prediction model takes into account the relations between and within all input features with respect to the target feature, the TB-TOH evaluation metric falls under evaluation criterion 2.

Details of the implemented prediction model in the TB-TOH approach are as follows. We use the Random Forest (RF) algorithm for the classification task. While Decision Trees (DT) are widely used in clinical research due to their interpretability, transparency of the prediction model is not a requirement in this research. The implementation of a prediction model in our research guides as a way to evaluate the quality of the synthetic data set. Mendelevitch and Lesh (2021) define principles that are important when applying classifiers for evaluating predictive model performance to compare synthetic to real data sets. The authors emphasize the importance of an optimized, stable model. As RF models are an ensemble technique, they suffer less from the overfitting risks related to DT, often outperform DT models in terms of score, and are considered to be more stable. Therefore, we decided to use the RF algorithm for the 1-year survival prediction task in this research.

Secondly, as indicated before, the 1-year survival prediction is an imbalanced classification task. This imbalance in the training data leads to the model having problems with correctly predicting minority class instances. Several techniques exist to improve the performance of classification models on imbalanced data sets, including oversampling the minority class (using for example synthetic samples through SMOTE (Chawla et al., 2002)), undersampling the majority class, and cost-sensitive learning. For the Random Forest algorithm, cost-sensitive learning will affect the splitting criterion to penalize differently a false classification from the minority and majority class. In the RF implementation in scikit-learn[1], we can use the parameter 'class_weight' for this purpose. Setting the 'class_weight' parameter to 'balanced" implies that the weight (i.e., the penalty) applied is inversely proportional to the class frequency. After empirical experiments on the original data set, we concluded that setting the 'class_weight' to 'balanced' led to the best performance compared to oversampling or undersampling.

Thirdly, as mentioned, using optimized models is a key principle when applying classifiers to compare synthetic to real data sets. Therefore, we optimize the RF algorithms on the respective data sets. We use grid search with k-fold cross-validation ($k = 5$) for hyperparameter optimization. We fix the values for some hyperparameters based on empirical experiments on the original training set (e.g., 'class_weight') to speed up computation time. Additionally, as we aim to build a stable classifier, overfitting is to be avoided. Therefore, we limit the maximal tree depth to a maximum of 10 and use a minimum leaf size of 0.005. The exact hyperparameters tested (or fixed) for the RF algorithm in this research are shown in Table 4.5.

Finally, it is important to note that for TB-TOH to be an informative measure of synthetic data set quality, the model trained on the original data has to have (better than random) predictive performance. The reason behind this is that if the model trained on the original data set has close to random performance, a model trained on the synthetic data set with close to random performance will obtain a high score on TB-TOH of around 1.0, while indicating little on synthetic data set quality. The AUC-ROC score provides an intuitive measure of predictive performance, as an AUC-ROC score of 0.5 corresponds to no predictive power, irrespective of the class (im)balance in the data set. AUC-ROC scores of at least 0.7 are generally considered acceptable (Hosmer et al., 2013) and this value may thus guide as a minimum threshold for strong enough predictive performance of the model trained on the original data set.

[1] https://scikit-learn.org

Hyperparameter	Values
n_estimators	[100, 150, 200]
criterion	[gini, entropy]
max_depth	[3, 5, 10]
min_samples_leaf	0.005
max_features	[sqrt, log2]
class_weight	balanced

TABLE 4.5: Hyperparameter values used or tested for RF to select best hyperparameter values combination using grid search with cross validation

4.3.3 Temporal metric

Metric 7: Sequential pattern mining (Jaccard Similarity)

In order to examine whether the synthetic data consists of meaningful treatment sequences $\mathcal{S}$, we propose an evaluation metric focusing on the temporal aspect of the data. Inspired by the evaluation metric on association rule mining proposed by Baowaly et al. (2019), we suggest a more suitable metric for sequential data: sequential pattern mining. As opposed to association rule mining that identifies co-occurrences among a set of features (i.e., which items appear together frequently, regardless of their ordering), sequential pattern mining identifies patterns in sequences by taking into account the ordering of items or events. Intuitively, in order for a synthetic data set to be of good quality, sequential patterns mined from the original data set and synthetic data set should (largely) overlap. As this metric considers and evaluates the temporal aspects and patterns of the treatment sequences, it is focused on evaluation criterion 3.

The SPADE algorithm (Zaki, 2001) is used for extracting frequent sequential patterns of the treatment sequences $\mathcal{S}$. As it is not meaningful to extract sequential patterns from treatment sequences with length 1, we exclude all patients with $|\mathcal{S}_p| = 1$ from the original and synthetic data sets for this evaluation metric (i.e., only include those patients with $2 \leq |\mathcal{S}_p| \leq l$).

The SPADE algorithm includes some parameters. Firstly, the support of a sequential pattern is the number of sequences where the pattern occurs divided by the total number of sequences in the database. Here, we consider the database the subset of the data set (either original or synthetic) with only patients with $2 \leq |\mathcal{S}_p| \leq l$ included. The minimum support parameter defines the minimum support needed for a pattern to be included in its output. After some experimentation, and to evaluate the robustness of the results with respect to the chosen value for minimum support, we include three values for minimum support: [0.01, 0.02, 0.03]. Second, we set the minimum number of items in an extracted sequential pattern (i.e., the minimum length of a sequential pattern) to two. Hereby we avoid including single, frequently occurring treatments to be mined as frequent sequential patterns. It is not meaningful to include frequent sequential patterns of length 1, as it does not capture any temporal aspect and the general occurrence probability of treatment codes is already evaluated in Metric 2.

The output of the sequential pattern mining algorithm is a set of extracted frequent sequential patterns R mined from the data set. We define $R_{P_{train}}$ as the sequential patterns mined from the original data set and R_Z as the sequential patterns mined from the synthetic data set. We can then quantitatively assess the quality of the synthetic data set in terms of temporal patterns captured by calculating the Jaccard similarity between $R_{P_{train}}$ and R_Z. The Jaccard similarity coefficient is defined as the size of the intersection divided by the size of the union of two sets. We use the Jaccard similarity metric as opposed to precision and recall metrics proposed by Baowaly et al. (2019). The reason for this can best be illustrated by two examples. Consider a scenario in which 50 frequent sequential patterns are extracted from the original data set (i.e., $|R_{P_{train}}| = 50$). Then, as a first example, imagine that from the synthetic data set, only one frequent sequential pattern is extracted (i.e., $|R_Z| = 1$). This pattern is also in $R_{P_{train}}$ (i.e., $|R_{P_{train}} \cap R_Z| = 1$). Intuitively, a synthetic data set in which only 1 out of 50 sequential patterns is reproduced is considered a synthetic data set of low quality. However, the precision (the number of common patterns extracted from both original and synthetic data sets divided by the number of patterns extracted in the synthetic data set) is 1, which is considered the best value possible for precision. It would require the interpreter of the score to also look at the recall (the number of common patterns extracted from both original and synthetic data sets divided by the number of patterns extracted from the original data set) score (which would, in this case, be low) to make a just claim on synthetic data set quality. On the other hand, as a second example, imagine that from the synthetic data set 300 frequent sequential patterns are extracted (i.e., $|R_Z| = 300$), in which all frequent sequential patterns in $R_{P_{train}}$ are included (i.e., $|R_{P_{train}} \cap R_Z| = 50$). In this case, recall is 1, while a synthetic data set with a large amount of spurious frequent sequential patterns can intuitively not be considered a high-quality synthetic data set. Again, it would require the interpreter to also look at the precision, which in this case would be low. Instead, the Jaccard similarity of the synthetic data sets in both of these examples is low, and therefore it serves as a single and unambiguous metric of synthetic data set quality with respect to sequential patterns.

The Jaccard similarity coefficient is formally specified in Equation 4.6. The Jaccard similarity coefficient is bounded between zero and one. Jaccard similarity of one means that the mined sequential pattern sets from the original and synthetic data sets are exactly the same, where Jaccard similarity equal to zero means that there is no overlap between the two mined frequent sequential pattern sets. The closer the Jaccard similarity is to one, the more similar the frequent sequential pattern set of the synthetic data R_Z is to the frequent sequential pattern set of the real data $R_{P_{train}}$, and hence the better the quality of the synthetic treatment sequences.

$$Jaccard\ Similarity\,(R_{P_{train}}, R_Z) = \frac{|R_{P_{train}} \cap R_Z|}{|R_{P_{train}} \cup R_Z|} \tag{4.6}$$

4.4 Experimental setup

This section describes the experimental setup for the two experiments conducted in this study, followed by the implementation.

For our experimental procedure in general (i.e., for both experiments), we first split the preprocessed, original data set P into a training set P_{train} and a test set P_{test} with a train-test split of 80%-20%, stratified on 1-year survival. P_{test} functions as the separate test set for Metric 6 (TB-TOH). We use P_{train} to train our generative models.

4.4.1 Experiment 1 - No Privacy (NP)

We started experimenting with a differentially private version of PrivBayes in our initial experiments. However, it appeared that even for high epsilon values of $\varepsilon = 10$, the quality of the synthetic data was very poor. This can be accounted to the fact that in our use case of patient data with covariates and treatment sequences, both temporal relations as well as relations between treatment sequences and selected patient covariates (i.e., context columns) exist. Our proposed network structure for PrivBayes Sequential was by far unable to satisfy the θ-usefulness constraints described in Section 2.2.1. Therefore the noise completely dominates the original signal, making the high-dimensional conditional distribution next to useless. On the other hand, limiting the parent set size of treatment attributes such that they would satisfy the θ-usefulness constraint has very restricting consequences. In particular, it would imply that for θ in the range $[3-6]$ (values suggested appropriate by the original authors (Zhang et al., 2017)), the maximum parent set size of the treatment attributes is between 17 and 35 for $\varepsilon = 1$, a value considered the highest acceptable ε value for IKNL. A maximum parent set size of at most 35 means that we cannot even use the previous treatment attribute (with dimensionality 47) as a parent of the current treatment attribute, and thus the algorithm cannot capture temporal relations, let alone both temporal and covariate (context columns) relations. For these reasons, we decided to focus on an evaluation of the non differentially private versions (abbreviated NP) of the selected methods in Experiment 1. Non differentially private essentially means setting $\varepsilon = \infty$.

For Experiment 1, we generate one synthetic data set using Marginal Synthesizer (MS-NP) and one using DoppelGANger (DG-NP) using the parameters defined in the respective sections (if applicable). For the non differentially private version of PrivBayes Sequential (PBS-NP), there are no boundaries on the parent set sizes of the patient covariates as explained in Section 2.2.1. Consequently, step 1 (GreedyBayes) will output a $(d-1)$-degree Bayesian network on the set of attributes in the patient covariates part $\mathcal{N}_c$ that estimates $Pr[\mathcal{A}_c]$ without suffering from any information loss. This means that only copies of existing original records (with respect to patient covariates) are generated (i.e., it is comparable to random sampling with replacement from the patient covariates part of the original data set to generate a synthetic data set). From a privacy perspective, this is highly undesirable. Therefore, we use a method introduced by Zhang et al. (2017) called "NoPrivacy". While this method was proposed only for evaluation purposes to test how much differential privacy was affecting the output quality, it is a useful way of limiting the network size in a non differentially private version of PrivBayes. The "NoPrivacy" method uses an ε value in order to define the θ-usefulness criterion to limit the maximum parent set size of the attributes, without perturbing the mutual information choices and conditional probabilities (so, no differential privacy). As PrivBayes gives no straightforward way to evaluate the most appropriate network size (i.e., parent set sizes for attributes), we use three values of $\varepsilon : [10, 1, 0.1]$, where smaller values of ε correspond to a lower network degree (i.e., stricter limits on parent set sizes) for $\mathcal{N}_c$. In these experiments, we use the default value of $\theta = 4$. To avoid confusion with differential privacy (which - as mentioned - this experimental setup does *not* cover), we term these values as network sizes [bigger network size, medium network size, smaller network size] respectively. We ensure that each of these networks is initialized with the same (randomly chosen) root note, so that variation between networks of different degrees cannot originate from pure randomness. This also allows us to evaluate the claim by Zhang et al. (2017) that a network of low degree is sufficient to approximate the data. In short, we use PBS-NP to generate three synthetic data sets, one for each different network size of $\mathcal{N}_c$: [bigger network size, medium network size, smaller network size].

Regarding the context attributes H used in PBS-NP, we consulted the 'Richtlijnendatabase'[2] (Database of guidelines) from Federatie Medisch Specialisten for the guidelines on treatment choices for colorectal cancer patients[3] as well as a clinical expert on colorectal cancer from IKNL. From a qualitative evaluation of the guidelines, it appeared that 'Stage' and the location of the tumor (i.e., 'Sublocation' in its more specific version) have the most important influence on treatment choices for patients diagnosed with colorectal cancer. The important influence of these two patient covariates on treatment choices was confirmed by the clinical expert. Additionally, the expert mentioned that 'Age at diagnosis' can play a role in the choice of treatment (e.g., through higher chances of post-operative mortality[4]), while not always defined as an absolute contraindication in the guidelines. Also, expected survival time may be taken into account when making treatment decisions, partly based on patient preferences (e.g., in case of metastatic (Stage 4) cancer, patients often have a say themselves in whether to go for life-prolonging treatments). Finally, treatment choices may vary strongly per patient and typically more criteria are considered and clinical. These are not always recorded in the data of IKNL and thus cannot be included in the network. Hence, also with the intention to keep the network small(er), we include the following set of patient covariates as context attributes $H = \{$'Stage', 'Sublocation', 'Age at diagnosis', '1-year survival'$\}$.

In conclusion, five synthetic data sets are generated and evaluated: one using Marginal Synthesizer (MS-NP), one using DoppelGANger (DG-NP), and three using PrivBayes ([PBS-NP bigger network size, PBS-NP medium network size, PBS-NP smaller network size]).

Expectations

In order to confirm the validity of the proposed quality metrics, it is important to hypothesize their expected behavior on the synthetic data sets generated by the different methods employed in this research. This section will describe the expectations on the quality metrics for the non-private (NP) generative methods.

First of all, it is relatively straightforward to hypothesize the expected results of the quality metrics on a synthetic data set generated by MS-NP. MS-NP models each feature independently, and thus relations between features will largely be lost. On the other hand, as the method focuses solely on retaining individual feature distributions, we expect MS-NP to perform well (i.e., serve as an upper bound) for quality metric 1 regarding individual patient covariate distributions. For the metrics on univariate statistics regarding static treatment features (quality metrics 2, 3, and 4), its behavior is expected to be worse due to the post-processing we conduct (i.e., handle the treatments after the first sampling of an 'empty' treatment as 'empty', regardless of the sampled value). This effect is expected to be most evident for quality metric 4 regarding sequence lengths, as it becomes harder to generate long treatment sequences when using the aforementioned post-processing step. Finally, MS-NP is expected to perform poorly on quality metrics involving (temporal) relations between features (i.e., quality metric 5, 6, and 7) and it serves as a baseline for the other two generative methods here.

[2] https://richtlijnendatabase.nl/

[3] https://richtlijnendatabase.nl/richtlijn/colorectaal_carcinoom_crc/startpagina_-_crc.html

[4] https://richtlijnendatabase.nl/richtlijn/colorectaal_carcinoom_crc/primaire_behandeling_coloncarcinoom_bij_crc/obstructief_coloncarcinoom_bij_crc.html

Secondly, given that we use a non differentially private version PBS-NP, the method is expected to reliably construct a network with relations between patient covariates. Additionally, the construction of the sequential part of the network is based on reliable domain knowledge and temporal information. All conditional probabilities are calculated from our training set without adding noise, causing them to closely approximate the original data distribution. For these reasons, we expect PBS-NP to perform well both on quality metrics regarding univariate statistics as well as quality metrics regarding (temporal) relations between features. More specifically, we expect the performance of PBS-NP to come close to the performance of MS-NP on quality metric 1. Moreover, as a treatment is conditioned on the previous treatment, we do not need a post-processing step as all treatments after the first 'empty' treatments will be 'empty' by definition of the algorithm. Therefore, we expect PBS-NP to outperform MS-NP on quality metrics regarding static treatment features (quality metrics 2, 3, and 4). Furthermore, we expect PBS-NP to closely capture relations between features by reliably constructing the network and sampling from the actual conditional probability tables (i.e., without adding noise). Hence, its performance on quality metric 5 and 6 are expected to be high. Also, this is where the difference between the different variants of PBS-NP is expected to emerge. As PBS-NP bigger network size can capture more relations between patient covariates directly through allowing larger parent set sizes, we expect higher values for quality metrics 5 and 6 for this variant. The values for quality metrics 5 and 6 are expected to decline - or at most remain similar - as network sizes become smaller. Lastly, as we condition a treatment attribute on the previous treatment attribute for all PBS-NP variants, temporal relations are expected to be kept. More specifically, we hypothesize that especially sequential patterns with length 2 are accurately kept. Through indirect conditioning, sequential patterns of length 3 or more might be retained, but they might also be partially lost as we only directly condition on the previous treatment. In conclusion, we expect that PBS-NP can capture most sequential patterns in the original data set and thus perform well on quality metric 7, albeit some sequential patterns (especially those longer than 2) might disappear.

Finally, the behavior of the quality metrics on the synthetic data set generated by DG-NP is harder to hypothesize, as DG-NP is a fully black-box method. However, by using a separate, dedicated generator for patient covariates and treatment sequence generation, together with the conditioning on patient covariates in treatment sequence generation, we expect to realize high performance on quality metrics involving (temporal) relations between features (quality metrics 5, 6 and 7). Additionally, the use of an LSTM network in sequence generation allows the generative methods to capture longer-term correlations in the treatment sequences. Therefore, DG-NP might be more effective in capturing temporal relations compared to PBS-NP. On the other hand, PBS-NP can directly capture relations between consecutive treatments, potentially being at an advantage here. Finally, we expect DG-NP to capture univariate distributions less closely than MS-NP and PBS-NP as it generates data from noise instead of directly sampling from marginal or conditional probability distributions. Nonetheless, we expect DG-NP to resemble the univariate distributions in the original data to a reasonable extent.

4.4.2 Experiment 2 - PrivBayes Sequential with Differential Privacy (DP)

In our initial experiments, we naively applied a differentially private version of PrivBayes Sequential. With a naive way of implementing differential privacy, we mean using differential privacy as proposed by the original authors of PrivBayes

(Zhang et al., 2017) in our adapted version PrivBayes Sequential. More specifically, differential privacy is implemented in GreedyBayes for construction of $\mathcal{N}_c$ (step 1 PrivBayes Sequential) and in conditional distribution generation (step 4 PrivBayes Sequential). Recall from Section 2.2.1 that for differentially private conditional distribution construction, noise is added to the joint probability distributions of each node X_j and its parents Π_j. Through the addition of noise in the joint probability distributions in the differentially private version of PrivBayes Sequential, it may happen that we sample an 'empty' treatment (i.e., no treatment) for a specific treatment column, and sample a treatment for one of the next treatment columns. We post-process the synthetic data generated by the differentially private version of PrivBayes Sequential in the same way as for Marginal Synthesizer (Section 4.2.4) by cutting the treatment sequence once the first 'empty' treatment is sampled and thus handling the next treatments as 'empty' regardless of the value sampled by PrivBayes Sequential. Recall from Section 2.4.1 that this post-processing operation will not affect the privacy level of the outputted synthetic data.

However, we found that the naive way of implementing differential privacy with the aforementioned post-processing did not result in synthetic data with reasonable quality. In particular, the synthetic data set generated using the naive way of implementing differential privacy lead to poor results for Metric 2 (treatment occurrence percentages), Metric 4 (Sequence length distribution), Metric 5 (PCD, especially PCD for associations between patients covariates and derived treatment features, and PCD for associations within derived treatment features) and Metric 7 (Sequential pattern mining). For exact performance values for the naive way of implementing differential privacy, we refer to Section 5.2.

This poor performance of the naive implementation of differential privacy in PrivBayes Sequential can be explained as follows. As briefly mentioned in Section 4.4.1, our proposed network structure for PrivBayes Sequential was by far unable to satisfy the θ-usefulness constraints for the treatment attributes. Recall from Section 2.2.1 that the θ-usefulness constraint is used to control the signal-to-noise ratio of the differentially private joint probability distributions. By substantially violating the θ-usefulness constraints in PrivBayes Sequential, the dimensionality of the joint probability distributions for the treatment attributes was this large that the noise needed to obtain ε-differential privacy completely dominated the original signal.

Therefore, we started experimenting with ways to increase the signal-to-noise ratio for acceptable values of ε. Straightforwardly, the signal-to-noise ratio may be increased by (1) increasing the signal, i.e., the average scale of information in each cell of the joint probability distribution, or (2) decreasing the scale of noise added. In general, as derived from the explanation of the original PrivBayes algorithm in Section 2.2.1, the signal-to-noise ratio of the joint probability distributions $Pr[X, \Pi]$ is determined by the following parameters:

1. The privacy budget ε, where higher values of ε lead to a higher signal-to-noise ratio keeping all else equal (decreases the amount of noise added).

2. The number of individuals in the data set n, where higher n leads to a higher signal-to-noise ratio keeping all else equal (increases average amount of information in each cell of the conditional probability distribution).

3. The number of attributes in the data set d, where lower d leads to a higher signal-to-noise ratio keeping all else equal (decreases the amount of noise added due to composability property of differential privacy).

4. The number of cells in the joint distribution $Pr[X, \Pi]$ m, where lower m leads to a higher signal-to-noise ratio keeping all else equal. Lowering m can be done in two non-exclusive ways: (1) by decreasing the domain size of X and/or the domain size of the attributes in Π, (2) by decreasing the parent set size Π (decreases the size of joint probability distribution and hereby increases the average scale of information in each cell of it).

Regarding the first parameter, ε, we cannot simply increase the privacy budget as values of $\varepsilon > 1$ are generally unacceptable, especially for medical data. Also, the PrivBayes algorithm (Zhang et al., 2017) is considered the state-of-the-art for obtaining differentially private synthetic data through Bayesian networks. Developing alternative ways to decrease the amount of noise needed to be added to the joint probability distributions while still obtaining differential privacy is - if possible at all - outside of the scope of this research. Second, increasing n is also considered unfeasible, as chancing our inclusion criteria or focusing on a wider range of cancer types in the NCR at the same time increases the cardinality of the attributes, in turn having a negative impact on the signal-to-noise ratio. Thirdly, decreasing the number of attributes in our data set is unreasonable, since we already limited the number of patient covariates included in this research, which will likely only increase when IKNL plans to publish a synthetic data set in practice.

This leaves us with the fourth option: decreasing the number of cells in the joint distributions $Pr[X, \Pi]$, specifically for the treatment attributes in our data set. We took three steps to decrease the number of cells in the joint distributions of the treatment attributes: (1) Decrease the domain size of the treatment attributes by replacing the specific treatment codes $e_{i,p}$ (47 unique values) by their main groups $g(i, p)$ (13 unique values), using the mapping provided by IKNL, (2) Replace the context column 'Sublocation' (10 unique values) by its broader version 'Topology code' (3 unique values), which is considered enough detail for generating treatment *main groups*, and (3) Remove the context column 'Age at diagnosis' (considered the context column with the weakest relation to treatment choices) from the parent sets of the second treatment attributes onward, that is $Treatment_i$ attributes, $i \in [2, l]$. The context column 'Age at diagnosis' is considered the context column with the weakest relation to treatment choices by the clinical experts, also indicated by it not being defined as an absolute contraindication in the guidelines. Moreover, the context column 'Age at diagnosis' has a strongest correlation with any of the derived treatment attributes equal to 0.29, being the lowest among all four context columns (a strongest correlation with any of the derived treatment attributes 0.6 for 'Topology code', 0.61 for 'Stage' and -0.40 for '1-year survival'). Moreover, we still apply the post-processing of the treatment sequences by marking the first 'empty' treatment attribute sampled as the end of the treatment sequence for a patient.

Still, after these steps were taken, a peculiar characteristic of the synthetic data set appeared when including a reasonable privacy budget (i.e., $\varepsilon \leq 1$). This is shown in Figure 4.4 for $\varepsilon = 1$, where the low signal-to-noise ratio causes the sequence length distribution in the synthetic data set to deviate significantly from the original data set. This can be explained through a combination of the sequential data pivoting and the way of adding noise in the original PrivBayes algorithm. The category 'empty' is an increasingly dominant category for $Treatment_i$ attributes, $i \in [2, l]$, as generally treatment sequences are short. The smaller, yet still too large to satisfy θ-usefulness, joint probability distributions for these attributes contain many small values or values equal to 0. The way PrivBayes adds noise is by sampling from the same scale of Laplace noise for each of the cells in the joint probability distributions, regardless

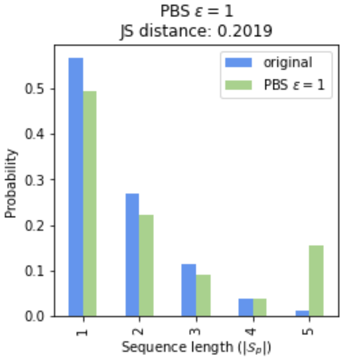


FIGURE 4.4: Comparison of original and synthetic sequence length distribution after first adaptations to the differentially private version ($\varepsilon = 1$) of PrivBayes Sequential

of the probability value of this cell. This way of adding noise is done to ensure that the algorithm obtains differential privacy, and herewith a form of "plausible deniability" that a unique pattern in the synthetic data set is caused by it also being in the original data set, as any pattern now has a possibility of being sampled simply by chance through the addition of noise. Note that the longer the sequence, the more unique it probably will be. Hence, by the nature of differential privacy, statistics that are very specific to individuals are masked, while patterns about groups are more likely and meant to be preserved. In general, the way of noise addition to the joint probability distributions (using the Laplace mechanism) and obtaining differential privacy suggested by Zhang et al. (2017) implies that for approximately half of the zero/small probability values, a positive amount of noise is added, while for the other half a negative amount of noise is added (leading to negative probability values). PrivBayes then sets the negative probability values to zero and normalizes the noisy distribution, causing the (normalized) probabilities for the dominant ’empty’ category to decrease for the sake of the small increases in value for approximately half of the zero/small probability values of treatment groups. This phenomenon explains why the chances of sampling ’empty’ categories for $Treatment_i$ attributes, $i \in [2, l]$ becomes smaller. In short, the larger the joint probability distributions, the more cells have small values or values equal to 0 (especially given the category ’empty’ being dominant for $Treatment_i$ attributes, $i \in [2, l]$) and the more the probabilities for the ’empty’ categories decrease for the sake of adding small amounts of noise to these cells with small or zero probability value. Not sampling any ’empty’ values for any of the *Treatment* attributes directly implies a sequence length of 5, explaining the high peak in the synthetic sequence length distribution compared to the original sequence length distribution for sequence length equal to 5.

In order to mitigate this issue, we could further decrease the number of cells in the joint distributions by removing other context columns. However, removing other context columns is considered unreasonable, as these are most certainly known to highly affect treatment choices from domain knowledge. Thus, removing these links from the network will considerably affect output quality as well, specifically in terms of keeping relations between patient covariates and treatment sequences. Therefore, we decided to focus on another way of reducing the deviations in the synthetic sequence length distribution by slightly changing the generation procedure and post-processing of the synthetic data. Our procedure is as follows. We add an additional attribute ’Sequence length’ to the patient covariates, which represents the length of the treatment sequence for the patient. We model this attribute in $\mathcal{N}_c$

using step 1 of our adapted PrivBayes Sequential algorithm (Section 4.2.2). Then, we generate a synthetic version of the original data set using a differentially private version of PrivBayes Sequential, including our proposed and aforementioned deviations. Finally, we cut each synthetic sequence of length 5 (i.e., sequences for which no 'empty' categories were sampled for any of the treatment attributes) to the synthetic 'Sequence length' attribute sampled for that patient. In other words, we drop all treatments $e_{i,p}$ in the synthetic treatment sequence for $i >$ value of the 'Sequence length' attribute of this synthetic patient p, only for those synthetic patients p for which all treatments are sampled (i.e., no 'empty' categories in the treatment sequence).

The additional steps taken to include differential privacy in the PrivBayes Sequential method can be summarized as follows:

1. Generalize the treatment codes (dimensionality 47) to treatment main groups (dimensionality 13).
2. Replacing the context column 'Sublocation' (dimensionality 10) by its broader version 'Topology code' (dimensionality 3).
3. Removing the context column 'Age at diagnosis' from the parent sets Π_i of $Treatment_i$ attributes, $i \in [2, l]$
4. Add 'Sequence length' attribute to patient covariates.
5. Post-processing the synthetic data set by cutting the treatment sequence once the first 'empty' treatment is sampled, for synthetic patients for which the value 'empty' is sampled within the treatment sequence. For those synthetic patients with all treatments sampled (i.e., no 'empty' value sampled in the treatment attributes), the treatment sequence is cut at the synthetic value of the 'Sequence length' attribute for this patient.

By implementing these adaptations, we generate two synthetic data sets using a differentially private version of PrivBayes Sequential (PBS), one for each value of $\varepsilon : [1, 0.1]$. Due to both time and computational limitations, we were unable to generate a differentially private synthetic data set for DoppelGANger and thus Experiment 2 only covers PrivBayes Sequential. We did not have access to a GPU, and the confidentiality of the data restrained us from using a publicly available GPU. The estimated time for training DoppelGANger including DP on our machine was 210 hours for the desired 105 epochs. Also, Lin et al. (2020) found that DoppelGANger with DP leads to poor-quality synthetic data in their use case, further motivating our decision not to include the computationally heavy DoppelGANger with DP.

Expectations

In general, we expect the results for all quality metrics to deteriorate whenever the privacy budget ε decreases. Thus, we expect worse results on all quality metrics for PBS-DP compared to PBS-NP, and we expect PBS $\varepsilon = 0.1$ to perform worse compared to PBS $\varepsilon = 1$ for all quality metrics. Furthermore, as explained in Section 4.4.2, we hypothesize that our suggested pre- and post-processing adaptations improve the quality of the differentially private synthetic data, compared to the naive way of implementing differential privacy.

4.4.3 Implementation

All experiments were conducted using Python on an HP ZBook with an i7-8565U CPU and 16GB RAM. The DoppelGANger algorithm is implemented using the GitHub repository[5] by Lin et al. (2020) and PrivBayes and Marginal Synthesizer are implemented using (adaptations of) the GitHub repository[6] by IKNL.

[5] https://github.com/fjxmlzn/DoppelGANger

[6] https://github.com/daanknoors/synthetic_data_generation

Chapter 5

Results

In this section, we present the results of the generative methods in the experimental setup with regards to the quality metrics presented in Section 4.3. We show and interpret the results for the generative methods per experiment, per quality metric.

5.1 Quality evaluation - Experiment 1 (NP)

5.1.1 Univariate metrics

Patient covariates

Table 5.1 shows the values for Metric 1 (average JS distance patient covariates) for all generative methods described in the experimental procedure. Recall that lower values correspond with better synthetic data quality. It can be seen that the MS-NP and all three variants of the PBS-NP generate univariate feature distributions that are very close to the original univariate feature distributions. The smaller size of the static network seems to have only a negligible impact on the JS distance between univariate feature distributions in the synthetic and original data set. As mentioned, MS-NP serves as a lower bound for this metric. In general, we observe that both MS-NP and all variants of PBS-NP generate univariate feature distributions that are very similar to the univariate feature distributions of the original data set.

Method	**Metric 1 ↓** **JS distance covariates**
MS-NP	0.0011
PBS-NP bigger	0.0011
PBS-NP medium	0.0015
PBS-NP smaller	0.0019
DG-NP	0.0151

TABLE 5.1: Values for quality metric 1 (JS distance patient covariates) for all non-private (NP) generative methods. The symbol on the right of the metric indicates: ↓ the lower the better.

For DG-NP, the JS distance is considerably higher than for all the other generative methods. To get some more understanding of what these quantitative values of JS distance mean, Figure 5.1 shows a visual representation of the distances between the probability distributions in the original and synthetic data. Here, each bar represents a category of one of the eight features in the patient covariates part of the synthetic data set, where the line represents the probability of this category in the original data set. For a more interpretable visualization on a per-feature basis, we refer to Appendix A. Visually, we can observe that univariate feature distributions in the

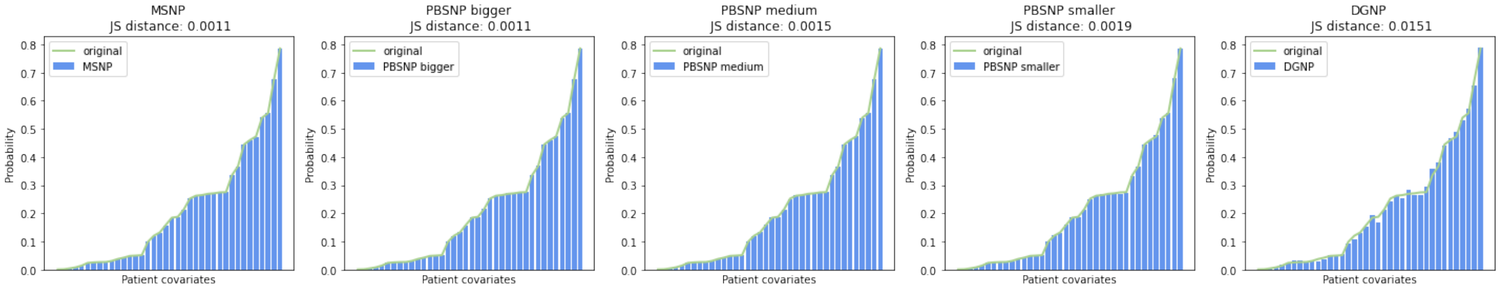


FIGURE 5.1: Visual representation of univariate feature distributions in the original (line) and synthetic (bar) data set for all non-private (NP) generative methods. Each value on the x-axis represents a category of one of the eight features in the patient covariates part of the data set. Categories are ordered by their probability of occurrence in the original data set.

synthetic data sets generated by MS-NP and PBS-NP are almost identical to univariate feature distributions in the original data set (the green line and blue bars follow each other almost perfectly). Albeit not showing this almost perfect similarity in the synthetic data set generated by DG-NP, the green line generally follows the bars closely. Thus, the DG-NP method is able to generate synthetic data in which univariate feature distributions resemble the original univariate feature distributions, although not as perfectly as MS-NP and all variants of PBS-NP.

Sequential data statistics

The values for the three quality metrics regarding sequential data statistics for all generative methods are presented in Table 5.2. First of all, PBS-NP bigger network size provides the best results for Metric 2 (RMSE occurrence percentages in the original and synthetic data set). For PBS-NP medium and smaller network sizes, the RMSE is a bit higher, but still considerably lower than the RMSE values for MS-NP and DG-NP. As mentioned before in our expectations, the worst performance of MS-NP on this quality metric can be explained by the post-processing conducted when using MS-NP for sequential data, where we consider the first ’empty’ treatment sampled the end of the treatment sequence. This post-processing slightly distorts the univariate distribution for the treatment codes. Finally, DG-NP reaches an RMSE of 0.0027, higher than RMSE for PBS-NP, but lower than RMSE for MS-NP.

Method	**Metric 2** **RMSE ↓ (CC* ↑)** **occurrence percentages**	**Metric 3 ↑** **Support coverage treatments**	**Metric 4 ↓** **JS distance sequence lengths**
MS-NP	0.0037 (0.9932)	1.0	0.1367
PBS-NP bigger	0.0002 (1.0000)	1.0	0.0007
PBS-NP medium	0.0003 (0.9999)	1.0	0.0022
PBS-NP smaller	0.0008 (0.9996)	1.0	0.0069
DG-NP	0.0024 (0.9966)	0.8511	0.0202

TABLE 5.2: Values for quality metrics related to sequential data statistics for all non-private (NP) generative methods. * We also include the correlation coefficient (CC) for metric 2, as it allows for comparison with prior research. The symbols on the right of the metrics indicate: ↓ the lower the better, and ↑ the higher the better

To get some more intuition on the quantitative RMSE values, Figure 5.2 shows scatterplots of treatment occurrence percentages in the original data set (x-axis) versus

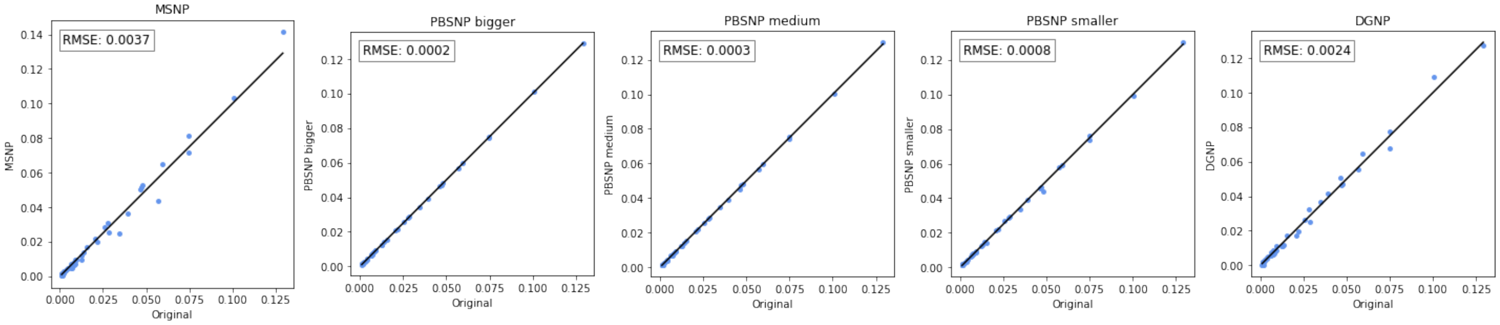


FIGURE 5.2: Scatterplots of treatment occurrence percentages in the original data set (x-axis) vs. synthetic counterpart (y-axis) for all non-private (NP) generative methods. Each dot represents one of the 47 treatment codes. The diagonal line indicates the ideal performance where the original and synthetic data set show equal occurrence percentages for each treatment.

its synthetic counterpart (y-axis) for all non-private (NP) generative methods. The black diagonal line indicates the ideal performance where the original and synthetic data set show equal occurrence percentages for each treatment. Here, the better performance of the three PBS-NP also appears. Yet, all five generative methods approximately follow the diagonal line, indicating that prevalent (infrequent) treatments are generally also prevalent (infrequent) in the synthetic data sets. Additionally, we can compare our results to the results presented in the research of Baowaly et al. (2019) on the generation of static representations of diagnosis, medications, and procedure codes. However, the representation, as well as the data set, are different, and RMSE values are influenced by the range of the occurrence percentages. Therefore, correlation coefficients (CC) between occurrence percentages in the original and synthetic data sets are a more appropriate way of comparing results across data sets and use cases. A higher value for CC means a higher correlation between the treatment occurrence percentages in the original and synthetic data set and thus higher values correspond with better synthetic data set quality in this respect. Baowaly et al. (2019) show CC values between 0.9913 and 0.9935 on different data sets for their best-performing generative method. Both DG-NP and all three variants of PBS-NP outperform these results by showing CC values between 0.9966 (DG-NP) and 1.0 (PBS-NP bigger network size), where the CC value for MS-NP (0.9932) is similar to the results found by Baowaly et al. (2019). However, it is important to note that an exact comparison between our results and the results presented by Baowaly et al. (2019) is impossible to make, as the dimensionality of the codes in their research is substantially bigger than in our study, potentially increasing the difficulty of the task. To conclude, the three variants of PBS-NP can best approximate the original treatment occurrence percentages, yet the two other generative methods (DG-NP and MS-NP) also provide acceptable approximations of the original treatment occurrence percentages.

Regarding the support coverage of treatment codes, MS-NP and all three variants of PBS-NP show support coverage of 1, meaning that all 47 treatment codes are present in the synthetic data set. For DG-NP however, support coverage is 0.8511, meaning that only 40 out of 47 unique treatment codes are present in the synthetic data. From the visual representation of treatment occurrence percentage percentages presented earlier (Figure 5.2), we can derive that the treatment codes not represented in the

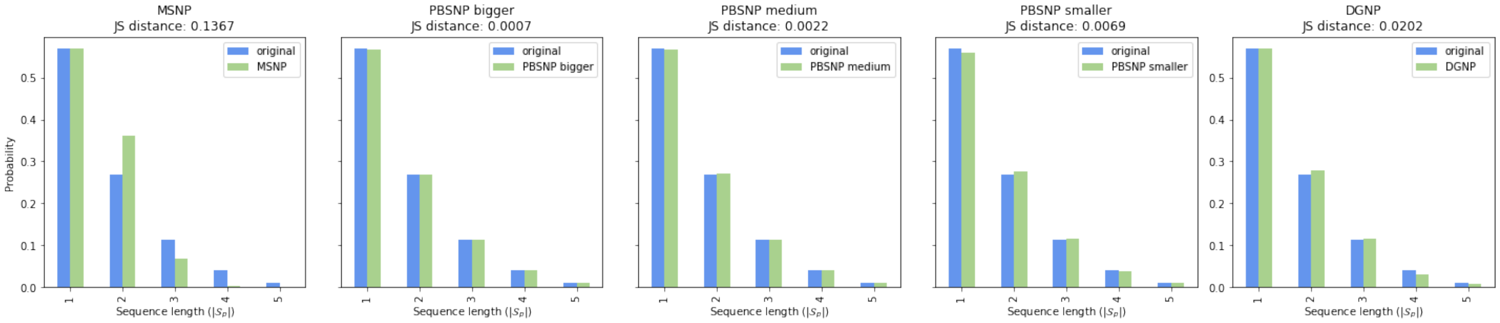


FIGURE 5.3: Bar plot of sequence length distribution in the original (blue) and synthetic (green) data set for all non-private (NP) generative methods.

synthetic data set are infrequent treatment codes in the original data set. As mentioned, the inability of GANs to learn and generate some infrequent classes in the data set is a well-known problem for GANs referred to as mode collapse. Despite our attempts to alleviate mode collapse using the PacGAN framework, DG-NP still suffered from partial mode collapse. Nonetheless, DG without PacGAN presented a support coverage of around 0.5. Thus, including the PacGAN framework did improve the results for DG-NP, yet still, it suffers from partial mode collapse, causing its inability to generate some infrequent treatment codes.

The final metric concerning sequential data statistics is the JS distance between sequence length distributions in the original and synthetic data set. As for the other metrics on sequential data statistics, PBS-NP bigger network size provides the lowest JS distance between treatment sequence distributions in the original data set. For the PBS-NP methods with smaller network sizes (medium and smaller), the JS is a bit higher, yet still considerably lower than for DG-NP.

To get some qualitative intuition on what these JS distances imply, 5.3 shows a visual representation of sequence length distributions in the original (blue bars) and synthetic (green bars) data sets for all non-private (NP) generative methods. It shows that MSNP generates shorter sequences (sequence lengths 1 and 2) and has difficulty generating longer sequences (sequence lengths 3, 4, and 5). Again, as expected, this can be accounted to the post-processing we perform when using MS-NP for sequential data and therefore we cannot consider MSNP an appropriate upper bound for this metric. Finally, from the visual plots, we observe that DG-NP can still reasonably approximate the original sequence length distribution and the general pattern of the sequence length distribution is kept, despite underperforming in this respect compared to the three variants of PBS-NP.

5.1.2 Relational metrics

Associations

In contrast with the first four quality metrics focused around univariate feature distributions, Metric 5 takes into account relations between features. As mentioned in Section 4.3.2, Metric 5 evaluates the extent to which relations between and within patient covariates and (static representations of) treatment sequences are retained in the synthetic data. Herewith, it provides a more convincing measure of synthetic data set quality for data analytics purposes beyond individual feature distributions.

Method	Metric 5 ↓ Pair-wise Correlation Distance (PCD)	PCD covariates	PCD covariates-treatments	PCD treatments
MS-NP	3.7460	1.3884	2.2158	1.5116
PBS-NP bigger	0.6872	0.0970	0.4598	0.2002
PBS-NP medium	0.7273	0.1712	0.4780	0.2063
PBS-NP smaller	0.9594	0.2042	0.6415	0.2361
DG-NP	1.1333	0.1860	0.2904	1.0398

TABLE 5.3: Values for quality metric 5 (PCD feature associations in the extended versions of the data sets V) for all non-private (NP) generative methods. The symbol on the right of the metric indicates: ↓ the lower the better.

Table 5.3 presents the PCD over the association matrices of the extended static versions of the data sets V (Metric 5) for all generative methods described in the experimental procedure. Additionally, the table provides the PCD for specific subsets of the features in the data set, namely: PCD covariates (PCD for association matrix including all patient covariates, in order to measure if the relations *within patient covariates* are kept), PCD covariates treatments (PCD for association matrix including patient covariates on the y-axis and derived treatment features on the x-axis, in order to measure if the relations *between patient covariates and (static representations of) treatment sequences* are kept) and PCD treatments (PCD for associations matrix including all derived treatment features, in order to measure if the relations *between treatments* are kept). Figure 5.4 shows what part of the association matrix each of these measures covers.

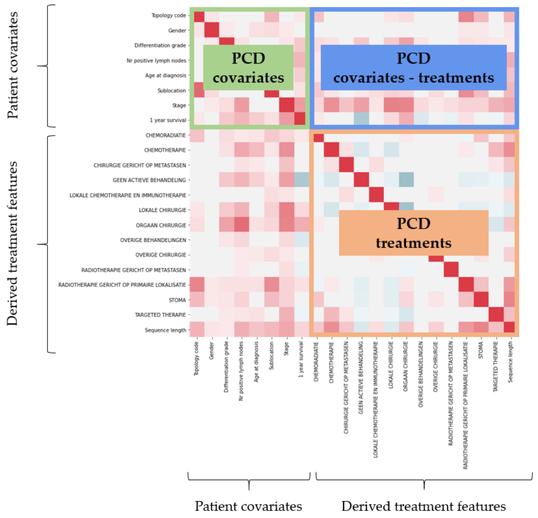
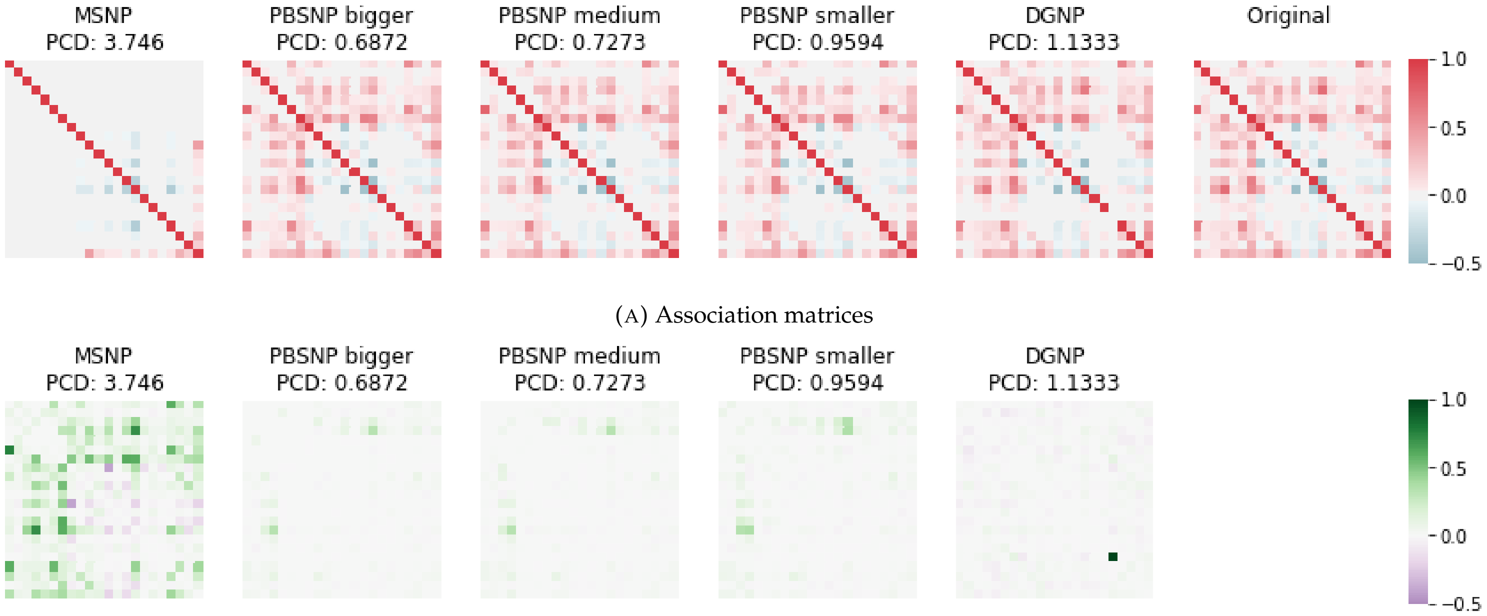


FIGURE 5.4: Visual representation of PCD for subsets of features

MS-NP shows the highest PCD for both the total association matrix as well as for each of the measures on a subset of features. This is in line with what we expected, as MS-NP models each feature independently according to its marginal distribution and therefore does not take into account relations between features. Both PSB-NP and DG-NP considerably outperform MS-NP in terms of PCD, with the best score for PBS-NP bigger network size. Again, this is in line with the expectations, as PBS-NP bigger network size is least restricted in terms of parent set sizes of (and thus, relations between) patient covariates. From the measures on subsets of features in



(A) Association matrices

(B) Differences between synthetic and original association matrices

FIGURE 5.5: Association matrices of the extended versions V of the synthetic data set for all non-private (NP) generative methods and the original data set.

the data set, we indeed observe that the difference between the PCD of the PBS-NP bigger network size and the smaller network sizes mainly comes from the PCD covariates. The smaller the network size, the fewer patient covariates directly conditionally depend on each other, and thus it becomes harder to closely retain all associations between patient covariates.

Interestingly, when comparing the results for DG-NP and PBS-NP, we observe that the PCD covariates for DG-NP is almost similar in magnitude to the PCD covariates for PBS-NP medium network size, and smaller than the PCD covariates for PBS-NP smaller network size. For larger network sizes, PBS-NP can more accurately retain associations in the original data set compared to DG-NP. Again, this is in line with the expectations, as PBS-NP bigger network size allows patient covariates to be directly conditioned on a large part of the other covariates. Therefore, associations between these features can be kept directly. Additionally, it is striking that DG-NP considerably outperforms all three variants of PBS-NP when it comes to PCD between patient covariates and derived treatment attributes. This observation can be explained as follows. Where PBS-NP can only capture direct dependencies between treatments and user-defined context columns, DG-NP seems to model (latent) relationships between all patient covariates and treatment sequences accurately. On the other hand, all three variants of PBS-NP considerably outperform DG-NP when it comes to associations within derived treatment features.

As mentioned in Section 4.3.2, it is important to visually compare association matrices next to the quantitative comparison. Figure 5.5a presents association matrices for all synthetic data sets, as well as the original data set (right-most association matrix). For visual purposes, Figure 5.5b shows heatmaps of the differences in values in the synthetic and original association matrix (association original - associations synthetic). Thus, in the lower subplot, the colors correspond with weaker (green) and stronger (purple) associations in the synthetic data compared to the original data. Bigger-sized association matrices for each of the synthetic data sets including exact association values and feature labels are included in Appendix B. We observe that the association matrices for DG-NP and all three variants of PBS-NP visually mimic the original association matrix. The association matrix of MS-NP shows - as expected - almost no relations between features. The fact that some associations between derived treatment features arise may be due to the fact that some treatment main groups only infrequently occur, and therefore are unlikely to co-occur in the synthetic data set generated by MS-NP (negative associations shown in the middle of the association matrix). In addition, we visually observe some of the aforementioned reflections, e.g., the better ability of DG-NP to capture relations between patient covariates and derived treatment features (shown by the fewer high differences in the lower subplots for that part of the association matrices for DG-NP as compared to the three PBS-NP heatmaps). Finally, we note that due to its inability to generate infrequent treatment codes (i.e., partial mode collapse), one treatment main group (RADIOTHERAPIE GERICHT OP METASTASEN) was not present in the synthetic data set generated by DG-NP, causing associations of all zeros to arise in the association matrix of DG-NP for this derived treatment feature. This explains the underperformance of DG-NP on PCD treatments (i.e., a larger distance between the synthetic and original associations within derived treatment features) compared to to PBS-NP, especially due to zero (i.e. difference of one with the original) occurring for this treatment main group on the diagonal. The diagonal simply represents associations between the same feature (which is always equal to one if the feature is present in the synthetic data) and thus represents no meaningful quality aspect of synthetic data set quality other than support coverage (Metric 3). Interestingly, when we replace this zero value on the diagonal for RADIOTHERAPIE GERICHT OP METASTASEN with a one, DG-NP reaches a PCD of 0.5333, which is the lowest among all generative methods in this research. Therefore, we can say that despite losing associations and relations through mode collapse, DG-NP closely retains associations between features it does generate. Additionally, this observation confirms our claim that it is important to visually compare association matrices next to a quantitative comparison.

TB-TOH

Recall from Section 4.3.2 that utility metrics provide the most convincing evidence of data realism and that we, therefore, include a TB-TOH evaluation of a 1-year survival prediction task. The results for TB-TOH are presented in Table 5.5 and Figure 5.6. The AUC-ROC score of an optimized RF classifier on the original test data set equals 0.837 (green dashed line in Figure 5.6), a value generally considered as good discrimination (Hosmer et al., 2013). This indicates that our original data set contains predictive power when it comes to 1-year survival classification.

As expected, MS-NP does not capture any relations between features and therefore the model trained on the data generated by MS-NP reaches an AUC-ROC score of 0.623 (a value considered poor, close to random discrimination (Hosmer et al., 2013)) on the original holdout set and a TB-TOH score of 0.745. This value thus serves as a lower bound for the TB-TOH metric. The PBS-NP bigger and medium network size perform best on this respect, both reaching a TB-TOH score of 1.0. This implies that models trained on the synthetic data sets generated by PBS-NP bigger network size and PBS-NP medium network size perform equally well on the holdout test set as a model trained on the original training data. Again, this can be explained by the fact that PBS-NP bigger network size and PBS-NP medium

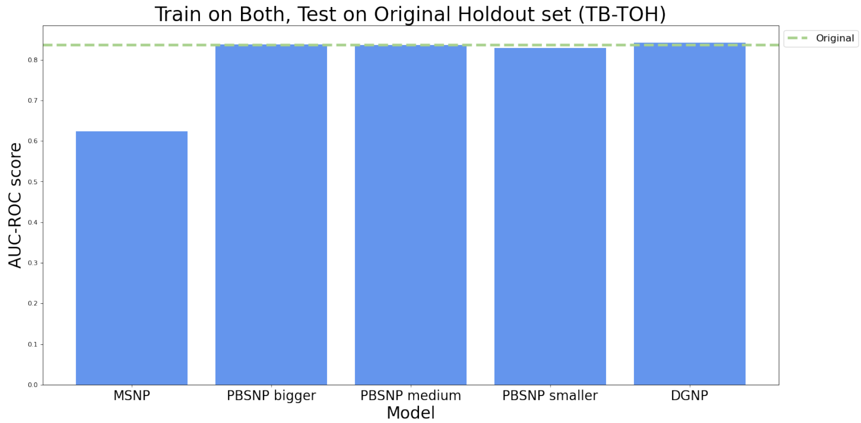


FIGURE 5.6: Comparison of predictive performance for 1-year survival prediction on the original holdout test set of original (dashed green line) and all non-private (NP) generative methods (blue bars).

network size are less restricted on parent set sizes of patient covariates, and thus are able to directly capture more correlations between these features. Interestingly, while the PCD scores for PBS-NP smaller network size increases considerably, the TB-TOH score decreases only to a very small extent (TB-TOH score of 0.991 for PBS-NP smaller network size). This implies that although some associations between features are less present in the synthetic data, the most important relations between features and our target variable (1-year survival) are retained. This finding supports the claim of Zhang et al. (2017) that lower-degree networks can sufficiently approximate the original data distribution, at least for the purpose of retaining predictive power in the synthetic data set.

For DG-NP, we observe a TB-TOH score of 1.007. This implies that a model trained on the synthetic data set generated by DG-NP performs slightly better on the original holdout set compared to a model trained on the original training set. The - potentially unexpected - better performance of a model trained on synthetic data compared to a model trained on the original data (i.e., TB-TOH > 1) also appears for some generative methods and data sets in other research (e.g., Mendelevitch & Lesh, 2021; Yoon et al., 2019). It can potentially be explained by the possibility that a generative method (especially a flexible method as a GAN) may learn an approximate, more general representation of the original data distribution and therefore generates a synthetic, less noisy version of the original data set that captures the overall patterns. The potential decrease of noise in the synthetic data set compared

Method	**AUC-ROC**	**Metric 6 (1)** **TB-TOH**
MS-NP	0.623	0.745
PBS-NP bigger	0.837	1.000
PBS-NP medium	0.837	1.000
PBS-NP smaller	0.829	0.991
DG-NP	0.843	1.007

TABLE 5.4: Values for quality metric 6 (TB-TOH using extended versions of the data sets V) for all non-private (NP) generative methods. The symbol on the right of the metric indicates: (1) the closer to one the better.

to the original training set may simplify the task and complexity of the survival prediction model. This may in turn lead to a prediction model that is less influenced by noise and therefore generalizes better to a holdout test set. However, the better performance of the model trained on the synthetic data set generated by DG-NP compared to the model trained on the original training set is only very subtle and no evidence of the hypothesized explanation can be provided.

In conclusion, the results on TB-TOH imply that the synthetic data sets generated by DG-NP and all three variants of PBS-NP can be used to train a survival prediction model with comparable performance on a holdout test set of real examples as the model trained on the original training set.

5.1.3 Temporal metric

Sequential pattern mining

As mentioned in Section, Metric 7 (Jaccard similarity sequential patterns) measures the extent to which temporal patterns within treatment sequences are kept in the synthetic data set. Table 5.5 shows the outputs of the sequential pattern mining algorithms on the synthetic and original data set as well as the Jaccard similarity (Metric 7) for the three minimum support thresholds defined in Section 4.3.3. Most importantly, DG-NP and all three variants of PBS-NP considerably outperform MS-NP for all values of minimum support, by obtaining a Jaccard similarity of around 0.8. While the measure of Jaccard similarity on extracted *sequential* patterns in synthetic data has not been used in other research, we can compare our results to the results of Baowaly et al. (2019) to get some initial insight. In their research, Baowaly et al. (2019) generate static, binary versions of diagnosis, medications, and procedure codes and evaluate synthetic data set quality based on precision and recall of extracted association rules, amongst others. Still, the results they provide allow us to calculate Jaccard similarities as well. Their highest-performing model reaches a Jaccard similarity between association rules of 0.67, being lower than the Jaccard similarity of around 0.8 for PBS-NP and DG-NP in our research. However, this comparison should be made with care for two reasons: (1) both data sets and tasks (static data generation and sequential data generation) differ, and (2) the research of Baowaly et al. (2019) contains data sets of higher dimensionality (i.e., between 942 and 1651 unique codes compared to 47 unique treatment codes in our research), potentially making the rule/pattern extraction task more challenging. Nonetheless, this comparison provides some additional insights into the quality of our results.

Additionally, we see some differences in the order of generative model performance for the different values of support. Therefore, we cannot draw any conclusion on which generative model (PBS-NP or DG-NP) performs best on this metric, and our main finding is that both perform reasonably well and considerably outperform the baseline (MS-NP).

As expected, the baseline method MS-NP performs poorly on Metric 7, obtaining a Jaccard similarity between extracted sequential patterns of 0.1563 for a minimum support value of 0.01. However, from a first perspective, it is striking that the synthetic data set generated by MS-NP does contain some frequent sequential patterns at all, of which some also exist in the frequent sequential pattern set extracted from the original data set. After an investigation of the extracted patterns, this phenomenon can be explained as follows. The extracted patterns that are also in the original data set (intersection extracted patterns MS-NP and original) are simply patterns of two frequently occurring treatments following each other. In other words,

Support	# extracted patterns original	Method	# extracted patterns synthetic	intersection size original synthetic	union size original synthetic	Metric 7 ↑ Jaccard similarity sequential patterns
0.01	51	MS-NP	24	10	65	0.1538
		PBS-NP bigger	48	45	54	0.8333
		PBS-NP medium	46	43	54	0.7963
		PBS-NP smaller	44	42	53	0.7925
		DG-NP	45	43	53	0.8113
0.02	26	MS-NP	4	2	28	0.0714
		PBS-NP bigger	25	23	28	0.8214
		PBS-NP medium	25	23	28	0.8214
		PBS-NP smaller	25	23	28	0.8214
		DG-NP	21	21	26	0.8077
0.03	13	MS-NP	2	1	14	0.0714
		PBS-NP bigger	13	11	15	0.7333
		PBS-NP medium	12	11	14	0.7857
		PBS-NP smaller	12	11	14	0.7857
		DG-NP	10	10	13	0.7692

TABLE 5.5: Sequential pattern mining results for all non-private (NP) generative methods. The last column shows metric 7: Jaccard similarity between sequential patterns extracted from the original and synthetic data set. The symbol on the right of the metric indicates: ↑ the higher the better.

if two treatments occur frequently, they might also be likely to follow each other frequently. The extracted patterns in the synthetic data set generated by MS-NP that are not in the original data set (difference extracted patterns MS-NP and original) are mainly two frequently occurring treatments following each other, that are unlikely to follow each other in practice (i.e., in the original data set). This entails for example two exactly the same frequently-occurring treatment codes following each other, or a surgery mainly executed for one specific topology followed by a surgery mainly executed for another topology. In conclusion, the only sequential patterns present in the data set generated by MS-NP are combinations of highly frequent treatments and its performance on this metric is - as expected - very low.

5.2 Quality evaluation - Experiment 2 (PBS-DP)

In this section, we present the results for Experiment 2, the differentially private version of PBS, including the additional pre- and post-processing steps we take to improve differentially private synthetic data set quality. Nonetheless, we included results for the naive way of implementing DP in PBS as well, in order to validate the expected increase in performance when including the additional pre- and post-processing steps in DP. In the following, we term the approach including the additional pre- and post-processing steps "Exp. 2", where the naive way of implementing DP in PBS is termed "Naive". For clarification, whenever we use the terms "PBS $\varepsilon = 1$" and "PBS $\varepsilon = 0.1$", we refer to Exp. 2. Whenever we elaborate upon the naive way of implementing DP, we explicitly mention this by using the terms "PBS $\varepsilon = 1$ (Naive)" and "PBS $\varepsilon = 0.1$ (Naive)". It is important to note that in Exp. 2 and Naive, the input data for the treatment data is different in dimensionality, where Naive covers the specific treatment codes (as in Experiment 1) and in Exp. 2 each treatment is mapped to its main group.

5.2.1 Univariate metrics

The results on metrics regarding univariate statistics (Metric 1, 2, 3, and 4) for the differentially private version of PBS for the two epsilon values are shown in Table 5.6. Indeed, we see that for three metrics (Metric 1, 2 and 4), PBS $\varepsilon = 1$ outperforms PBS

	Epsilon	**Metric 1 ↓** **JS distance** **covariates**	**Metric 2** **RMSE ↓ (CC* ↑)** **occurrence percentages**	**Metric 3 ↑** **Support coverage** **treatments**	**Metric 4 ↓** **JS distance** **sequence lengths**
Exp. 2	$\varepsilon = 1$	0.0284	0.0207 (0.9985)	1.0	0.0297
	$\varepsilon = 0.1$	0.0485	0.0724 (0.9848)	1.0	0.0642
Naive	$\varepsilon = 1$	0.0122	0.0222 (0.8682)	1.0	0.6019
	$\varepsilon = 0.1$	0.0398	0.0266 (0.4665)	1.0	0.7182

TABLE 5.6: Values for quality metrics related to univariate data statistics for PBS $\varepsilon = 1$ and PBS $\varepsilon = 0.1$. Exp. 2 is the DP approach including pre- and post-processing steps and Naive is the naive way of implementing DP. The symbols on the right of the metrics indicate: ↓ the lower the better, and ↑ the higher the better. * We also include the correlation coefficient (CC) for metric 2, as it is more applicable for varying treatment column dimensionality.

$\varepsilon = 0.1$ and that both underperform compared to PBS-NP. Only for Metric 3 (Support coverage treatments), results are the same as for PBS-NP. However, as mentioned before, this measure is a relatively basic measure of synthetic data set quality that only focuses on whether the generative model captures the diversity in treatment codes present in the original data set. It was mainly included as an indication of mode collapse in the GAN model and is, therefore, less relevant for this experiment.

When comparing the results of Experiment 2 with the results of the naive implementation of DP, we observe that their performance on Metric 1 is similar. This can be explained by the fact that Metric 1 covers patient covariates, where the θ-usefulness constraint for the static part of the network $\mathcal{N}_c$ in the PBS algorithm is met for both Exp. 2 and Naive. However, when looking at the metrics regarding sequential data statistics, especially Metric 2 and 4, we can see that Exp. 2 considerably outperforms the Naive experiment. For Metric 2 (RMSE occurrence percentages), we also included the correlation coefficient CC (Pearson's R). Recall that treatment data dimensionality differs between Exp 2. and Naive, where Naive has higher dimensionality. Therefore, occurrence percentages in Naive are smaller compared to Exp. 2, leading to lower values of RMSE. The CC is not impacted by a change in treatment data dimensionality, and therefore represents a more reliable quantitative evaluation of the similarity in treatment occurrence percentages between the original and synthetic data sets. CC for Exp. 2 are substantially higher than for Naive for both values of ε. Furthermore, for Metric 4 (JS distance sequence length), Exp. 2 considerably outperforms Naive for both values of ε, where the original sequence length distribution is largely lost in the synthetic data for the naive implementation of differential privacy. Based on the aforementioned results, we can conclude that our pre- and post-processing steps taken in Experiment 2 considerably improve the quality of the differentially private synthetic data with respect to sequential data statistics.

In order to define whether the synthetic data sets resemble the original data set closely when it comes to the metrics regarding univariate statistics, it is helpful to qualitatively assess the measures using visual representations. We present visual representations for Exp. 2 only, as the quantitative results provide substantial evidence that synthetic data set quality is higher for Exp. 2 than for the Naive experiment. Figure 5.7 represents a visual representation of Metric 1. For a visualization on a per-feature basis, we refer to Appendix C. Indeed, we recognize the better performance of PBS $\varepsilon = 1$ compared to PBS $\varepsilon = 0.1$ that was shown in the quantitative measure. Nonetheless, for both values of ε, the probability distributions of patient

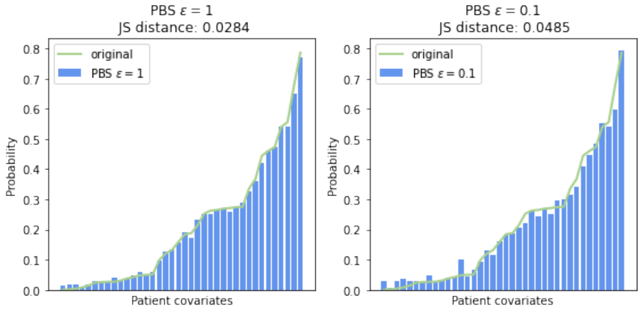


FIGURE 5.7: Visual representation of univariate feature distributions in the original (line) and synthetic (bar) data set for PBS $\varepsilon = 1$ and PBS $\varepsilon = 0.1$. Categories are ordered by their probability of occurrence in the original data set.

covariates in the synthetic data sets generally follow the probability distributions of patient covariates in the original data set.

A visual representation of Metric 2 is shown in Figure 5.8. Here, we see that the treatment occurrence percentages in the synthetic data set generated by PBS $\varepsilon = 1$ closely follow the black diagonal line indicating optimal performance, with a slight over-representation of infrequent treatment main groups and a slight under-representation of frequent treatment main groups. For PBS $\varepsilon = 0.1$ these over- and under-representations become more apparent, to an extent that the synthetic data set does not resemble the original data set anymore in this respect.

Lastly, Figure 5.9 presents a visual representation of Metric 4. Here, again the quantitative differences visually appear. Where PBS $\varepsilon = 1$ closely approximates the original sequence length distribution with a slight under-representation of sequence length 5, PBS $\varepsilon = 0.1$ shows a more apparent under-representation of sequence length 1, together with an over-representation of sequence lengths 4 and 5.

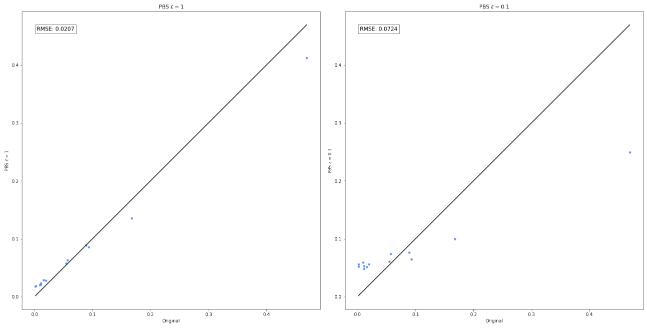


FIGURE 5.8: Scatterplots of treatment occurrence percentages in the original data set (x-axis) vs. synthetic counterpart (y-axis) for PBS $\varepsilon = 1$ and PBS $\varepsilon = 0.1$. Each dot represents one of the 47 treatment codes. The diagonal line indicates the ideal performance where the original and synthetic data set show equal occurrence percentages for each treatment.

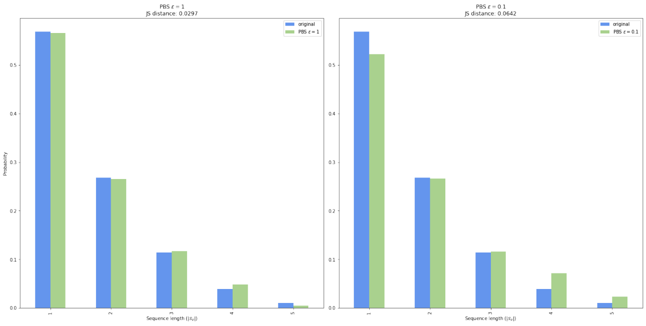


FIGURE 5.9: Bar plot of sequence length distribution in the original (blue) and synthetic (green) data set for PBS $\varepsilon = 1$ and PBS $\varepsilon = 0.1$.

5.2.2 Relational metrics

Associations

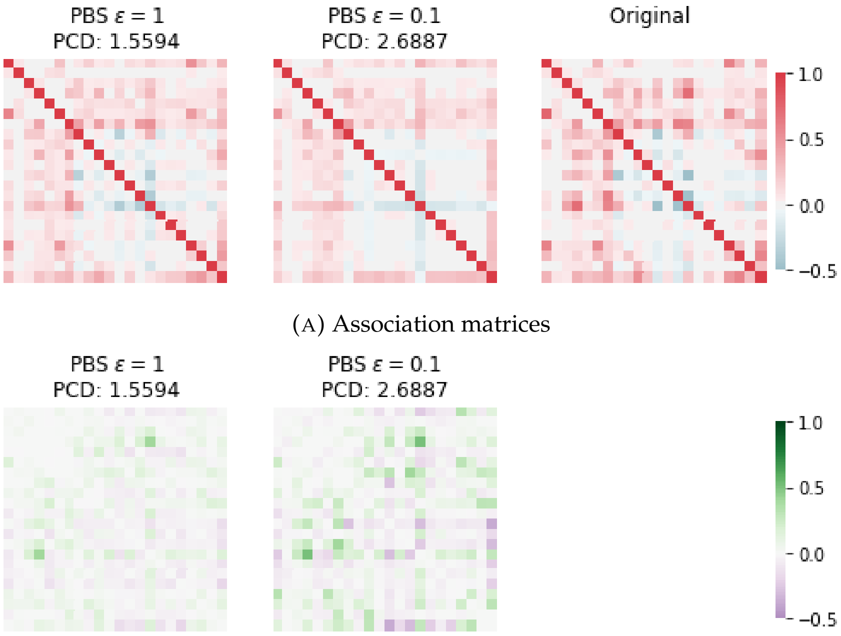
Table 5.7 presents the results for Metric 5. As for the metrics on univariate statistics, we observe that PBS $\varepsilon = 1$ outperforms PBS $\varepsilon = 0.1$ and that both underperform compared to PBS-NP. From the measures on a subset of features in the association matrix, we deduce that the main differences come from PCD treatments and to a lesser extent from PCD covariates-treatments. The differences in PCD covariates are relatively small. This can be explained by the fact that we satisfy the θ-usefulness constraint for the static part of the network $\mathcal{N}_c$ in the PBS algorithm, while not satisfying it for the user-defined sequential part of the network $\mathcal{N}_\mathcal{S}$.

The results of the naive implementation of DP show a higher PCD than for Experiment 2, for both ε values. As expected, as for both the Naive experiment and Exp. 2 the θ-usefulness constraint for the static part of the network $\mathcal{N}_c$ is satisfied, these increased PCD values for the naive implementation of DP compared to Experiment 2 mostly originate from PCD covariates-treatments and PCD treatments. Based on the results for Metric 5, we can conclude that our pre- and post-processing steps proposed in Experiment 2 considerably improve the quality of the differentially private synthetic data by better retaining relations between features.

	Epsilon	Metric 5 ↓ Pair-wise Correlation Distance (PCD)	PCD covariates	PCD covariates-treatments	PCD treatments
Exp. 2	$\varepsilon = 1$	1.5594	0.3269	0.8250	0.9817
	$\varepsilon = 0.1$	2.6887	0.3681	1.4553	1.6905
Naive	$\varepsilon = 1$	2.7539	0.1954	1.6419	1.4678
	$\varepsilon = 0.1$	3.4673	0.4402	2.0721	1.8004

TABLE 5.7: Values for quality metric 5 (PCD feature associations in the extended versions of the data sets V) for PBS $\varepsilon = 1$ and PBS $\varepsilon = 0.1$. Exp. 2 is the DP approach including pre- and post-processing steps and Naive is the naive way of implementing DP. The symbol on the right of the metric indicates: ↓ the lower the better.

Figure 5.10 shows the association matrices (Figure 5.10a) and heatmaps of the differences in synthetic and original association matrices (Figure 5.10b) for PBS $\varepsilon = 1$ and PBS $\varepsilon = 0.1$. We show the visual representations for Experiment 2 only, as the quantitative results already provide substantial evidence that synthetic data set quality is higher for Exp. 2 than for the Naive experiment. We examine that for PBS $\varepsilon = 1$ associations between features become weaker, but the general association patterns are preserved. Thus, the synthetic data set resembles the relations between features in the original data set up to a certain extent. For PBS $\varepsilon = 0.1$, (stronger) relations between features mostly disappear and do not represent the associations between features in the original data set. From a visual evaluation, it can be seen that this holds especially for relations between derived treatment features and relations between patient covariates and treatments, which was also noted from the quantitative evaluation.



(A) Association matrices

(B) Differences between synthetic and original association matrices

FIGURE 5.10: Association matrices of the extended versions V of the synthetic data set for PBS $\varepsilon = 1$ and PBS $\varepsilon = 0.1$ and the original data set.

TB-TOH

Table 5.8 presents the result for Metric 6 (TB-TOH). As for all the other metrics, we observe that PBS $\varepsilon = 1$ outperforms PBS $\varepsilon = 0.1$ and that both underperform compared to PBS-NP. Furthermore, Experiment 2 outperforms the Naive experiment for Metric 6. Nonetheless, the differences between Experiment 2 and the naive implementation of PrivBayes are not as evident as for the previous metrics.

Unexpectedly, however, especially when taking into account the values for Metric 5 (PCD), for both values of ε for both experiments, the results show that the generated synthetic data sets can be used to train a survival prediction model with performance on a holdout test set of real examples close to the model trained on the original training set. This phenomenon of limitations in synthetic data set quality not becoming evident from metrics regarding predictive performance is also observed in other research (Lin et al., 2020). For our specific use-case, this finding may be attributed to (one of) two different causes. First of all, from the visual association matrices (presented in a bigger size in Appendix D), we can observe that for our target variable ′ 1-year survival′, associations with other features are retained relatively well compared to associations between other features. As the TB-TOH metric solely focuses on 1-year survival prediction, and thus on associations between features and

	Epsilon	AUC-ROC	Metric 6 (1) TB-TOH
Exp. 2	$\varepsilon = 1$	0.822	0.982
	$\varepsilon = 0.1$	0.816	0.976
Naive	$\varepsilon = 1$	0.812	0.971
	$\varepsilon = 0.1$	0.781	0.933

TABLE 5.8: Values for quality metric 6 (TB-TOH using extended versions of the data sets V) for PBS $\varepsilon = 1$ and PBS $\varepsilon = 0.1$. Exp. 2 is the DP approach including pre- and post-processing steps and Naive is the naive way of implementing DP. The symbol on the right of the metric indicates: (1) the closer to one the better.

this target variable only, this may have caused the relatively high performance of the differentially private version of PBS on this metric. Another cause may be that the 1-year survival can be predicted with a reasonable performance from a subset of features in the data set. If these features include mainly static patient covariates (i.e., the subset of features for which the θ-usefulness constraint is satisfied and PCD differences for the differentially private synthetic data sets are lowest), this may cause the model trained on the synthetic data to be able to accurately rely on these potentially most important features for the 1-year survival prediction task in general. Therefore, the drop in prediction performance may not be as evident as the drop in synthetic data set quality when taking into account all associations between all features as in Metric 5. In order to provide an initial test on the second hypothesis, we obtained permutation feature importance scores for the classification model trained on the original data set. The permutation feature importance scores are shown in Figure 5.11. Indeed, it appears that the three most important features are the static patient covariates ′Stage′, ′Age at diagnosis′ and ′Nr positive lymph nodes′, with ′Stage′ being the most important feature that has a considerably larger impact on model predictions than any of the other features.

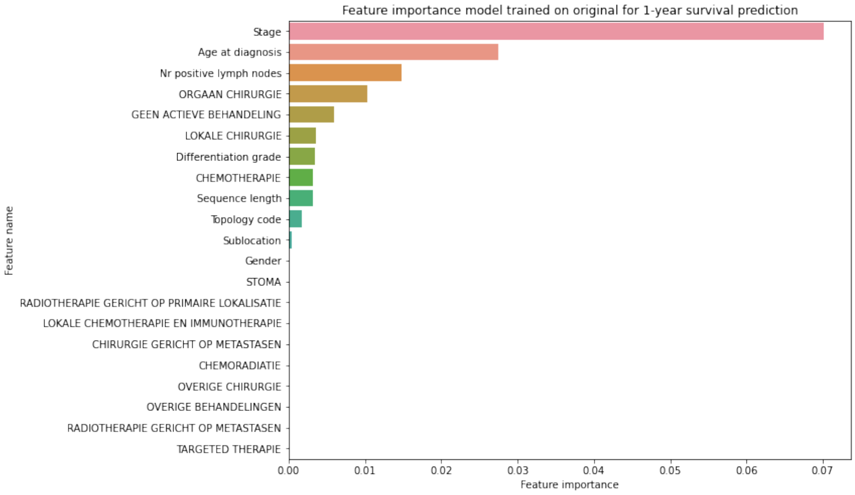


FIGURE 5.11: Bar plot of permutation feature importance scores for the model trained on the original data set.

5.2.3 Temporal metric

Sequential pattern mining

Metric 7 (Jaccard similarity sequential patterns) measures the resemblance of the synthetic treatment sequences to the original treatment sequences in terms of sequential patterns. Table 5.9 shows the results for Metric 7 for different values of minimum support. We confirm that PBS $\varepsilon = 1$ outperforms PBS $\varepsilon = 0.1$ and that both underperform compared to PBS-NP with regards to temporal patterns. Again, Experiment 2 outperforms the Naive experiment for Metric 7 for both values of ε. However, it is important to note that an exact comparison between Experiment 2 and the Naive experiment is impossible to make, as the values for Experiment 2 represent sequential patterns between treatment main groups, where the results for the Naive experiment represent sequential patterns between specific treatment codes. Nonetheless, we observe that most sequential patterns are lost entirely for the naive implementation of differential privacy. Also, we are able to accurately compare the results of the Naive experiment with the non-private baseline MS-NP, as both cover sequential patterns between specific treatment codes. We can see that the Naive experiment mostly underperforms compared to the non-private baseline MS-NP, with the only exception being for $\varepsilon = 1$ and Support $= 0.1$. Therefore, we can conclude that using the naive implementation of differential privacy, frequent sequential patterns within treatment sequences are not retained.

Interestingly, for PBS $\varepsilon = 1$, although some frequent sequential patterns disappear in the synthetic data set, also shown by the lower Jaccard similarity compared to PBS-NP, a reasonable amount of frequent sequential patterns is kept as the Jaccard similarity on extracted sequential patterns for PBS $\varepsilon = 1$ is considerably higher than the Jaccard similarity for the non-private baseline MS-NP for all values of minimum support. For PBS $\varepsilon = 0.1$, most temporal relations are lost, as values are close to the results obtained for the non-private baseline MS-NP that models no (temporal) relations between treatments. Hence, PBS $\varepsilon = 0.1$ performs poorly on this respect. Nonetheless, again, an exact comparison is impossible to make based on these results, as the values for PBS-DP represent sequential patterns between treatment main groups, where the results for MS-NP represent sequential patterns between treatment codes.

From these results, we can conclude that the pre- and post-processing steps taken in Experiment 2 increase the quality of the differentially private synthetic data with regards to the retention of sequential patterns. Moreover, we can generate synthetic data in which a reasonable amount of sequential patterns are kept for PBS $\varepsilon = 1$, using our proposed pre- and post-processing steps.

	Support	Epsilon	Metric 7 ↑ Jaccard similarity sequential patterns
Exp. 2	0.1	$\varepsilon = 1$	0.6042
		$\varepsilon = 0.1$	0.1720
	0.2	$\varepsilon = 1$	0.6552
		$\varepsilon = 0.1$	0.2857
	0.3	$\varepsilon = 1$	0.5714
		$\varepsilon = 0.1$	0.1818
Naive	0.1	$\varepsilon = 1$	0.3281
		$\varepsilon = 0.1$	0
	0.2	$\varepsilon = 1$	0.0385
		$\varepsilon = 0.1$	0
	0.3	$\varepsilon = 1$	0
		$\varepsilon = 0.1$	0

TABLE 5.9: Sequential pattern mining results for PBS $\varepsilon = 1$ and PBS $\varepsilon = 0.1$. Exp. 2 is the DP approach including pre- and post-processing steps and Naive is the naive way of implementing DP. The symbol on the right of the metric indicates: ↑ the higher the better.

Chapter 6

Discussion

6.1 Conclusion

The aim of this study was to generate synthetic health event data that resembles the statistical properties of the original data, in the application context of (colorectal) cancer patient data including both patient covariates and treatment sequences. Thus, the data has a static (patient covariates) as well as a sequential (treatment sequences) component. Generating this type of data is challenging in the sense that multiple types of relations are present in the original data, including relations within static patient covariates, relations between static patient covariates and treatment sequences (i.e., patient and tumour characteristics tend to influence treatment choices), and temporal relations within treatment sequences. The main research question is split into three parts, in which the focus lies mostly on the two former ones: (1) Identifying suitable existing generative methods and applying or adapting them to the use case of co-generating patient covariates and treatment sequences, (2) Examining and proposing relevant (existing) quality metrics that evaluate the extent to which the synthetic data resembles the original data (and thus evaluate synthetic data usefulness for data analytics purposes) in this context, and (3) Studying how to obtain *private* synthetic data.

Regarding the first sub-question, we identify and apply two generative methods for the generation of synthetic patient data including patient covariates and treatment sequences, and compare them against a naive baseline (termed MS-NP). The first method is an adaptation of the non-private PrivBayes (Zhang et al., 2017) algorithm designed for tabular data. We propose a data pivoting approach and the inclusion of temporal and domain knowledge to adapt and enable the method to co-generate patient covariates and treatment sequences and term this algorithm PBS-NP. Secondly, we apply an existing GAN model - DoppelGANger (Lin et al., 2020) - to our context and name it DG-NP. We show that both methods are capable of generating synthetic data that resembles the original data with respect to univariate feature distributions and (temporal) relations between and within patient covariates and treatment sequences, by considerably outperforming the naive baseline. Differences in the quality of the synthetic data from different perspectives generated by these methods appear, where the main limitation of DG-NP is that it suffers from partial mode collapse and generates only part of the treatment codes present in the original data set. PBS-NP does not have this limitation and is able to generate all treatment codes present in the original data set. On the other hand, DG-NP slightly outperforms PBS-NP when it comes to capturing relations between patient covariates and (static representations of) treatment data. Besides the differences in results, the generative methods differ in nature. DG-NP is a fully black-box method that does not require or allow for user input on hypothesized relationships, where PBS-NP enables interpretability of as well as user input to the generation process. The interpretability of PBS-NP allows for a more intuitive explanation of what the generated synthetic data is capable of and useful for, and the user input allows for active selection of which patterns (not) to preserve in the synthetic data.

For the third part of the research question, we shortly covered an experiment in which we use the differentially private version of PBS for two privacy budgets: $\varepsilon = 1$ and $\varepsilon = 0.1$. Several steps were taken to limit the negative effect of differential privacy on the quality of the synthetic data, including a generalization of the treatment data, slightly adapting the context columns, and post-processing the generated data. For both privacy budgets, the quality of the synthetic data was significantly harmed compared to the non-private setting, where this lead to a synthetic data set barely improving upon the non-private baseline for $\varepsilon = 0.1$. Nonetheless, for $\varepsilon = 1$, we were able to retain both univariate statistics as well as (temporal) relations between features to a certain extent by considerably outperforming the non-private baseline on all metrics. It is expected that this generalized type of treatment data is still useful for most data analytics purposes. However, it does not suffice whenever one aims to study the effects of specific treatment options (e.g., effects of a specific type of chemotherapy).

With respect to the second part of the research question, we propose seven quality metrics that cover various aspects of synthetic data set quality in the context of (colorectal) cancer patient data including both patient covariates and treatment sequences. Altogether, these metrics cover three important characteristics and criteria of this type of data and evaluate whether the synthetic data:

1. Retains univariate statistics of patient covariates and/or (static representations of) treatment sequences
2. Keeps existing relations between and within patient covariates and (static representations of) treatment sequences
3. Preserves temporal patterns within treatment sequences

The proposed metrics and their type and criterion they evaluate are presented in Table 6.1. We take the first steps in confirming the validity of the quality metrics by comparing their results on synthetic data generated by the aforementioned generative methods with their hypothesized behaviors. In general, the metrics perform

Metric(s)	**Type**	**Evaluation criterion**
Metric 1: JS distance patient covariates ↓ Metric 2: RMSE treatment occurrence percentages ↓ Metric 3: Support coverage treatment codes ↑ * Metric 4: JS distance sequence lengths ↓ *	Statistical	1. Univariate statistics
Metric 5: Associations (PCD) ↓	Statistical	2. Relations
Metric 6: TB-TOH (1)	Utility	2. Relations
Metric 7: Sequential pattern mining (Jaccard similarity) ↑	Utility	3. Temporal patterns

TABLE 6.1: Proposed metrics for evaluating synthetic data set quality in the context of (colorectal) cancer patient data including both patient covariates and treatment sequences. Symbols on the right of the metric indicate: ↓ the lower the better, ↑ the higher the better, and (1) the closer to one the better. Annotated asterisk (*) indicates the metric being a basic measure of synthetic data set quality and should mostly be used as a necessary condition.

as expected and give a correct indication of synthetic data set quality for our application purpose. Nonetheless, our results show that one of the most widely used evaluation metrics in the literature - Metric 6 (TB-TOH) - might give overly optimistic results. In particular, when the prediction model in TB-TOH heavily relies on one or a small subset of feature(s), a loss in association values may not become evident if the associations between the small subset of features and the target are kept. Extending TB-TOH by including multiple target variables (and hence multiple prediction models) is an interesting solution to overcome this limitation of TB-TOH with one target variable.

These quality metrics provide a unified indication of synthetic data set quality, but might not all be considered equally important. Metrics annotated with an asterisk (*) (Metric 3 and Metric 4) are rather basic metrics of data set quality and should mostly be used for confirming necessary conditions for useful synthetic data and may be of help when interpreting the other metrics. Then, Metric 1 and Metric 2 focus on univariate statistics only. Data that performs well on these metrics can be used for simple analytics purposes involving marginal feature distributions. However, in order to allow for more extensive analytics purposes, it is important that (temporal) relations within and between patient covariates and treatment sequences are accurately represented in the synthetic data. Thus, metrics involving (temporal) relations (Metric 5, 6, and 7) generally provide the most convincing evidence of synthetic data set quality for data analytics purposes and may be used to define sufficient conditions for synthetic data set quality. Exact conditions are highly dependent upon the context and purpose of the synthetic data and are outside the scope of this research, as they should be constructed in practice in relation to a specific purpose.

Additionally, some overlap between different metrics may exist. For example, a synthetic data set that performs poorly on Metric 5 (PCD Associations) - especially when it comes to relations between the target variable and other features - is expected to perform poorly on Metric 6 (TB-TOH), as prediction power is likely to be lost whenever associations between features disappear. Here, we can conclude that Metric 6 (TB-TOH) provides a more extensive quantitative indication of data set quality in terms of relations between features compared to Metric 5 (PCD Associations). However, as discussed, Metric 6 (TB-TOH), especially with only one target variable included, may give overly optimistic results. Therefore, it is valuable to include both metrics, as Metric 5 provides a more interpretable measure of relations between features, where one can visually compare association matrices to identify what associations exactly are lost or retained. A similar overlap exists between Metric 5 (PCD Associations, specifically within derived treatment features) and Metric 7 (Jaccard similarity sequential pattern mining), where temporal patterns are unlikely to be kept when static relations between derived treatment features are lost. Thus, here, Metric 7 provides a more extensive metric of relations within treatment sequences as it takes into account their temporal aspect.

The quality metrics may be presented in a table, from which the user can attach importance to each quality metric depending on the purpose of the synthetic data. Moreover, it is important to note that some quality metrics may require additional qualitative information through for example visual representations (e.g., especially in the case of Metric 5 PCD Associations) in order to convey a more meaningful indication of synthetic data set quality. Next to visual representations, quality metrics may include relevant sub-measures such as PCD on a subset of features in the association matrix (Figure 5.4) or Metric 7 (Jaccard similarity sequential pattern mining) for different values of minimum support.

To conclude, this thesis proposes a set of seven generic quality evaluation metrics for synthetic health data. Additionally, these metrics were specified to the context of synthetic health event data, with an application of (colorectal) cancer patient data including static patient covariates and treatment sequences. In our experiments, the quality metrics are validated using two (adapted) existing generative methods. Moreover, using the quality metrics, we show that both methods are able to generate synthetic colorectal cancer patient data - including static patient covariates and treatment sequences - that resembles the statistical properties of the original data in terms of univariate feature distributions and (temporal) relations between and within patient covariates and treatment sequences by considerably outperforming the baseline method. Lastly, by taking case-specific steps, we were able to generate a generalized differentially private synthetic data set with reasonable quality (i.e., outperforming the baseline) for privacy budget $\varepsilon = 1$.

With these results, this research contributes to the field of synthetic health event data generation, a topic largely unexplored in the literature that mainly focuses on either cross-sectional or time series data generation, and in practice. We believe that this research provides initial insights into both generative methods for health event data as well as ways to evaluate its quality and usefulness for data analytics purposes. Herewith, we aim to motivate other parties to consider the option of sharing synthetic patient data in order to safely increase the availability of patient data and accelerate advances in medical knowledge.

6.2 Limitations and future research

This section covers limitations related to the research and provides directions for future research. The limitations and suggestions for future research will be divided into three topics: (1) privacy, (2) quality metrics, and (3) data.

6.2.1 Privacy

This research focuses on the generation of synthetic data and evaluating its quality. While we covered one additional experiment with differential privacy guarantees, future work that focuses more on the privacy aspect of synthetic health event data is needed. In general, a generative method should aim to capture the general properties of the original data, without resembling it to an extent that unique patterns within the original data are transferred to the synthetic data. In case unique patterns are captured, the privacy of individuals in the synthetic data cannot be preserved anymore and this would undermine the motivation for releasing synthetic data. Preserving the privacy of individuals and maintaining the quality of the synthetic data might be two conflicting goals commonly referred to as the privacy-utility trade-off (Jordon et al., 2019).

Our results show that a naive implementation of differential privacy in PrivBayes substantially harms the quality of the generated health event data. In our differentially private experiment, we proposed some adaptations to improve the quality of the differentially private synthetic data. Using these additional steps, we are able to generate differentially private ($\varepsilon = 1$) synthetic health event data with reasonable quality, only at the cost of generalizing (i.e., decreasing the domain size of) the sequential data. However, these extensions are rather case-specific and future work is needed to extend (generic) methods for differentially private synthetic health event data generation.

On the other hand, our main experiments on non-private versions may have considerable implications on the privacy of the synthetic data. For example, in the case of the non-private PrivBayes method (PBS-NP), by directly sampling from (large) conditional probability distributions, any pattern that shows up in the synthetic data set between an attribute and its parents should have been present in the original data by definition of the algorithm. For unique patterns, this may lead to privacy leaks. Therefore, it is important to include randomness (i.e., noise) into the conditional probability distributions to create some sense of "plausible deniability". We encourage future research to investigate ways to implement randomness in the joint probability distributions, without including this much noise that the quality of the health event data deteriorates and becomes too poor for acceptable privacy budgets. Our initial suggestions would be to potentially add noise to the sequence as a whole or randomly applying noise to some time steps in the sequence, instead of randomizing every time step in the sequence. Moreover, one may focus on minimizing identifiability risk rather than implementing differential privacy, where unique patterns in the data need more noise (or distance from original data) in order to reduce the chance of identifying individuals from the synthetic data set (Yoon et al., 2020). In our use case, an example may be that a relatively large group of patients follows a standard treatment pattern (e.g., radiotherapy followed by rectal surgery for stage 3 rectal cancer (C20)). The treatment sequences for these patients do not cover any patient-specific, private information, as they are simply following the guidelines and apply for a relatively large group of patients. Data becomes more sensitive whenever treatment sequences derogate from standards or when a standard treatment sequence is given to a patient with unusual covariates for this sequence (e.g., local surgery for metastatic (Stage 4) cancer). While the aforementioned suggestions for noise addition may not lead to the generative methods formally satisfying differential privacy guarantees, they may be combined with post-generation privacy tests that make a strong case for plausible deniability for any individual as in Platzer and Reutterer (2021).

For the other generative method employed in this thesis, DoppelGANger, its black-box technique prevents an attacker from directly claiming that any pattern in the synthetic data should have been present in the original data. However, the adversarial training procedure together with the high model complexity of deep GAN networks encourages a distribution that is concentrated around training samples (Xie et al., 2018). Therefore, the model and synthetic data is susceptible to an inference attack that identifies sensitive information about individuals (Hitaj et al., 2017). Due to time and computational limitations, we were unable to implement differential privacy standards or apply post-generation privacy tests in this thesis. Future work may focus on either a post-generation evaluation of privacy of the synthetic data using metrics that make a strong case for plausible deniability mentioned in Section 2.4.2, or evaluate the quality of the synthetic data generated with a differentially private version of DoppelGANger. Lin et al. (2020) have already evaluated the efficacy of DoppelGANger with DP-GAN using Tensorflow Privacy[1], which lead to poor-quality synthetic data in their use case of networked time series data. However, it is interesting to evaluate whether this (extreme) reduction of quality also holds for the use case of health event data. In addition, future work may implement other DP-GAN frameworks that give minimal cost to utility within DoppelGANger, as for example PATE-GAN, an approach using differentially private student-teacher learning (Jordon et al., 2019).

[1] https://github.com/tensorflow/privacy

6.2.2 Quality metrics

While our quality metrics provide an initial framework to evaluate synthetic health event data, consisting of both static patient covariates as well as sequential (treatment) data, the metrics are not all-encompassing. Several interesting and needed extensions exist.

First of all, extensions of Metric 6 (TB-TOH) regarding predictive performance are interesting to examine. This research uses AUC-ROC as the evaluation score for the imbalanced 1-year survival prediction model. Future research may incorporate multiple, appropriate evaluation scores for the prediction model with regards to the case at hand (e.g., precision, recall, F1 score) for TB-TOH, together giving more evidence of the synthetic data quality. Furthermore, in this research, we apply the Random Forest model as a suitable discriminative model. Yet, other prediction algorithms with different characteristics (e.g., Support Vector Machine (SVM), Logistic Regression, Artificial Neural Network (ANN)) may be applied to obtain a more complete overview. Lastly, our TB-TOH evaluation is centered around one target variable (1-year survival) considered relevant for the data set at hand. However, this limits the utility evaluation to relations between features and this specific target variable only. In future research, multiple relevant features may be selected as the target once, and TB-TOH can be repeated for each of them. Also, multi-class or multi-label classification tasks specifically relevant for health event data may be considered, for example, a step-wise prediction of treatment sequences from static patient covariates. However, such a prediction task is complicated and non-trivial given the limited feature set included in this research (e.g., excluding treatment outcomes and side-information). In addition, in the most extreme case, every feature in the extended static version of the data set, including both patient covariates as well as derived treatment features, may be used as a target once and the TB-TOH metric can be averaged over all models. The use of multiple prediction tasks for TB-TOH may naturally help overcome its aforementioned limitation of being an overly optimistic measure of synthetic data set quality when using only one target variable if this one prediction model heavily relies on one or a small subset of features only. Nonetheless, it is important that each prediction model reaches a better than random performance on the original data set in order to provide a meaningful TB-TOH measure. Therefore, we suggest including only those prediction tasks with target variables that reach a better than random performance (e.g., AUC-ROC score of at least 0.7 for binary classification) on the original data set.

Second, techniques from a predictive perspective other than TB-TOH can be included to give a more complete indication of synthetic data set quality. For example, "**T**rain on **R**eal, **T**est on **S**ynthetic" (TRTS) is an interesting approach that evaluates whether relations with the target variable in the original data set are retained in the synthetic data set, as compared to TB-TOH that mainly tests whether predictive power (and thus, some relations with the target) is retained in the synthetic data set, without evaluating whether the synthetic model captures the exact same relations as in the original model. Another extension that determines the extent to which the classifiers trained on real versus synthetic data rely on the same features for their predictions is the difference or correlation in feature importance scores. The Random Forest algorithm applied in this research enables the user to obtain feature importance scores directly, and increasingly more methods are available for deriving feature importance scores for (black-box) prediction models (e.g., Lundberg & Lee, 2017). Feature importance scores may be relevant for multiple evaluation purposes. For example, in our research, using the feature importance scores, we can test our hypothesis that the models trained on the differentially private synthetic data sets rely on different (or, a subset of) features to make their predictions. In this case, while presenting a high score for TB-TOH, the score for a feature importance metric is expected to be lower than for the non-private methods.

Finally, for evaluating the extent to which the synthetic data set resembles the original data set in terms of relations between patient covariates and treatment sequences, this research mainly relies on PCD scores. Including alternative metrics can give more convincing evidence of synthetic data set quality in this respect. However, as mentioned before, treatment sequence prediction tasks from patient covariates are difficult given the limited (treatment) features at hand and the influence of patient preference on treatment sequences. Nonetheless, other interesting, potentially domain-specific metrics may be used. Examples include evaluating the extent to which treatment guidelines - that can be represented through rules (e.g., IF Stage equals 3 AND Sublocation equals 9, THEN Treatment sequence equals ('External radiotherapy', 'Low anterior resection')) - are followed in the original data versus in the synthetic data (for IKNL, these rules can be derived from Oncoguide for several cancer types[2]), or simply calculating the distance between conditional probability tables (CPTs) of patient covariates and presence of a treatment code in the treatment sequence for a patient (i.e., a binary variable for each treatment code). We encourage future research to investigate how these or alternative metrics regarding relations between patient covariates and treatment sequences can be incorporated into the quality evaluation framework. In addition, combining the quality metrics into a compact representation of synthetic data set quality is an interesting direction for future work. Nevertheless, we recommend allowing the user to detail such compact representation into the distinct specific metrics, as the different metrics may give a deeper and distinct understanding of different aspects of synthetic data set quality.

6.2.3 Data

This research is focused on a subset of features in the colorectal cancer data set provided by IKNL. More specifically, we include 8 features regarding patient covariates, and only the treatment code was included in the treatment sequence for a patient. In addition, numerical patient covariates were discretized using relatively wide bins. Both the selection of a subset of features and the binning procedure reduces the sampling space, making the generative methods more likely to closely resemble the training data and, consequently, reveal more sensitive patient information (Goncalves et al., 2020). Whether this phenomenon occurred - especially for DoppelGANger - may be evaluated in future research using post-generation privacy metrics. Moreover, future research may expand the data set in terms of the number or dimensionality of static patient covariates included as well as additional treatment-related information such as treatment outcome or drug dosage. Especially the inclusion of additional treatment-related information significantly increases the complexity of the problem at hand, making it unlikely that an interpretable, statistical method as PrivBayes can be used to generate this type of data. The use of DoppelGANger, which allows for the inclusion of multiple features per time step, or other deep generative methods using representation learning mentioned in Section 2.1.2, is probably more appropriate taking into account the complexity of this extension of the data set. Finally, we encourage future work to evaluate whether the (adaptations of) proposed generative methods and the quality metrics are applicable to broader use cases and data sets.

[2] https://oncoguide.nl/#!/projects

Bibliography

Agrawal, R., Srikant, R. et al. Fast algorithms for mining association rules. In: *Proc. 20th int. conf. very large data bases, vldb. 1215*. Citeseer. 1994, 487–499.

Arjovsky, M., Chintala, S., & Bottou, L. Wasserstein generative adversarial networks. In: *International conference on machine learning*. PMLR. 2017, 214–223.

Arnold, C., & Neunhoeffer, M. (2020). Really useful synthetic data–a framework to evaluate the quality of differentially private synthetic data. *arXiv preprint arXiv:2004.07740*.

Azur, M. J., Stuart, E. A., Frangakis, C., & Leaf, P. J. (2011). Multiple imputation by chained equations: What is it and how does it work? *International journal of methods in psychiatric research*, *20*(1), 40–49.

Baowaly, M. K., Lin, C. C., Liu, C. L., & Chen, K. T. (2019). Synthesizing electronic health records using improved generative adversarial networks. *Journal of the American Medical Informatics Association*, *26*(3), 228–241. https://doi.org/10.1093/jamia/ocy142

Beaulieu-Jones, B. K., Wu, Z. S., Williams, C., Lee, R., Bhavnani, S. P., Byrd, J. B., & Greene, C. S. (2019). Privacy-preserving generative deep neural networks support clinical data sharing. *Circulation: Cardiovascular Quality and Outcomes*, *12*(7), 1–10. https://doi.org/10.1161/CIRCOUTCOMES.118.005122

Camino, R., Hammerschmidt, C., & State, R. (2018). Generating Multi-Categorical Samples with Generative Adversarial Networks. http://arxiv.org/abs/1807.01202

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Che, Z., Cheng, Y., Zhai, S., Sun, Z., & Liu, Y. (2017). Boosting deep learning risk prediction with generative adversarial networks for electronic health records. *Proceedings - IEEE International Conference on Data Mining, ICDM*, *2017-November*, 787–792. https://doi.org/10.1109/ICDM.2017.93

Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W. F., & Sun, J. (2017). Generating Multi-label Discrete Patient Records using Generative Adversarial Networks. *68*, 1–20. http://arxiv.org/abs/1703.06490

Dahmen, J., & Cook, D. (2019). Synsys: A synthetic data generation system for healthcare applications. *Sensors*, *19*(5), 1181.

Dankar, F. K., & El Emam, K. (2013). Practicing differential privacy in health care: A review. *Trans. Data Priv.*, *6*(1), 35–67.

Dash, S., Dutta, R., Guyon, I., Pavao, A., Yale, A., & Bennett, K. P. (2019). Synthetic Event Time Series Health Data Generation. http://arxiv.org/abs/1911.06411

Dash, S., Yale, A., Guyon, I., & Bennett, K. P. Medical time-series data generation using generative adversarial networks. In: *International conference on artificial intelligence in medicine*. Springer. 2020, 382–391.

Dwork, C., McSherry, F., Nissim, K., & Smith, A. Calibrating noise to sensitivity in private data analysis. In: *Theory of cryptography conference*. Springer. 2006, 265–284.

Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*. https://doi.org/10.1561/0400000042

El Emam, K., Jonker, E., Arbuckle, L., & Malin, B. (2011). A systematic review of re-identification attacks on health data. https://doi.org/10.1371/journal.pone.0028071

Esteban, C., Hyland, S. L., & Rätsch, G. (2017). Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs. http://arxiv.org/abs/1706.02633

Fisher, R. A. Statistical methods for research workers. In: *Breakthroughs in statistics*. Springer, 1992, pp. 66–70.

Georges-Filteau, J., & Cirillo, E. (2020). Synthetic Observational Health Data with GANs: from slow adoption to a boom in medical research and ultimately digital twins?, 1–54. https://doi.org/10.22541/au.158921777.79483839/v2

Goncalves, A., Ray, P., Soper, B., Stevens, J., Coyle, L., & Sales, A. P. (2020). Generation and evaluation of synthetic patient data. *BMC Medical Research Methodology*, *20*(1), 1–40. https://doi.org/10.1186/s12874-020-00977-1

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial networks. *arXiv preprint arXiv:1406.2661*.

Hardt, M., Ligett, K., & Mcsherry, F. A simple and practical algorithm for differentially private data release (F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger, Eds.). In: In *Advances in neural information processing systems* (F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger, Eds.). Ed. by Pereira, F., Burges, C. J. C., Bottou, L., & Weinberger, K. Q. *25*. Curran Associates, Inc., 2012. https://proceedings.neurips.cc/paper/2012/file/208e43f0e45c4c78cafadb83d2888cb6-Paper.pdf

Hitaj, B., Ateniese, G., & Perez-Cruz, F. Deep models under the gan: Information leakage from collaborative deep learning. In: *Proceedings of the 2017 acm sigsac conference on computer and communications security*. 2017, 603–618.

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression* (Vol. 398). John Wiley & Sons.

Jordon, J., Yoon, J., & Van Der Schaar, M. (2019). PATE-GaN: Generating synthetic data with differential privacy guarantees. *7th International Conference on Learning Representations, ICLR 2019*, 1–21.

Kaur, D., Sobiesk, M., Patil, S., Liu, J., Bhagat, P., Gupta, A., & Markuzon, N. (2020). Application of Bayesian networks to generate synthetic health data. *Journal of the American Medical Informatics Association*, *00*(0), 1–11. https://doi.org/10.1093/jamia/ocaa303

Knoors, D. (2018). *Utility of differentially private synthetic data generation for high-dimensional databases* (Master's thesis). KTH Royal Institute of Technology. Stockholm, Sweden.

Lee, D., Yu, H., Jiang, X., Rogith, D., Gudala, M., Tejani, M., Zhang, Q., & Xiong, L. (2020). Generating sequential electronic health records using dual adversarial autoencoder. *Journal of the American Medical Informatics Association*, *27*(9), 1411–1419. https://doi.org/10.1093/jamia/ocaa119

Lin, Z., Jain, A., Wang, C., Fanti, G., & Sekar, V. (2020). Using GANs for Sharing Networked Time Series Data, 464–483. https://doi.org/10.1145/3419394.3423643

Lin, Z., Khetan, A., Fanti, G., & Oh, S. (2018). Pacgan: The power of two samples in generative adversarial networks. *Advances in neural information processing systems*.

Lundberg, S., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*.

McSherry, F., & Talwar, K. Mechanism design via differential privacy. In: *48th annual ieee symposium on foundations of computer science (focs'07)*. IEEE. 2007, 94–103.

Mendelevitch, O., & Lesh, M. D. (2021). Fidelity and privacy of synthetic medical data. *arXiv preprint arXiv:2101.08658*.

Nass, S. J., Levit, L. A., & Gostin, L. O. (2009). *Beyond the HIPAA Privacy Rule: Enhancing Privacy, Improving Health Through Research*. https://doi.org/10.17226/12458

Patki, N., Wedge, R., & Veeramachaneni, K. The synthetic data vault. In: *2016 ieee international conference on data science and advanced analytics (dsaa)*. IEEE. 2016, 399–410.

Perkonoja, K. (2020). *Generating synthetic longitudinal patient data with the privbayes method* (Master's thesis). University of Turku. Finland.

Platzer, M., & Reutterer, T. (2021). Holdout-based fidelity and privacy assessment of mixed-type synthetic data. *arXiv preprint arXiv:2104.00635*.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, *27*(3), 379–423.

Shokri, R., Stronati, M., Song, C., & Shmatikov, V. Membership inference attacks against machine learning models. In: *2017 ieee symposium on security and privacy (sp)*. IEEE. 2017, 3–18.

Sweeney, L. (2002a). Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *10*(05), 571–588.

Sweeney, L. (2002b). K-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *10*(05), 557–570.

Wang, Y., Wang, D., Ye, X., Wang, Y., Yin, Y., & Jin, Y. (2019). A tree ensemble-based two-stage model for advanced-stage colorectal cancer survival prediction. *Information Sciences*, *474*, 106–124.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., et al. (2016). The fair guiding principles for scientific data management and stewardship. *Scientific data*, *3*(1), 1–9.

Xie, L., Lin, K., Wang, S., Wang, F., & Zhou, J. (2018). Differentially Private Generative Adversarial Network. http://arxiv.org/abs/1802.06739

Xu, L., Skoularidou, M., Cuesta-Infante, A., & Veeramachaneni, K. (2019). Modeling tabular data using conditional GAN. *Advances in Neural Information Processing Systems*, *32*(NeurIPS).

Xu, L., & Veeramachaneni, K. (2018). Synthesizing Tabular Data using Generative Adversarial Networks. http://arxiv.org/abs/1811.11264

Yadav, P., Steinbach, M., Kumar, V., & Simon, G. (2018). Mining electronic health records (ehrs) a survey. *ACM Computing Surveys (CSUR)*, *50*(6), 1–40.

Yale, A., Dash, S., Dutta, R., Guyon, I., Pavao, A., & Bennett, K. P. (2020). Generation and evaluation of privacy preserving synthetic health data. *Neurocomputing*, (40). https://doi.org/10.1016/j.neucom.2019.12.136

Yan, C., Zhang, Z., Nyemba, S., & Malin, B. A. (2020). Generating electronic health records with multiple data types and constraints. *arXiv*.

Yoon, J., Drumright, L. N., & Van Der Schaar, M. (2020). Anonymization through data synthesis using generative adversarial networks (ADS-GAN). *IEEE Journal of Biomedical and Health Informatics*, *24*(8), 2378–2388. https://doi.org/10.1109/JBHI.2020.2980262

Yoon, J., Jarrett, D., & van der Schaar, M. (2019). Time-series generative adversarial networks. *Advances in Neural Information Processing Systems*, *32*(NeurIPS), 1–11.

Zaki, M. J. (2001). Spade: An efficient algorithm for mining frequent sequences. *Machine learning*, *42*(1), 31–60.

Zhang, J., Cormode, G., Procopiuc, C. M., Srivastava, D., & Xiao, X. (2017). Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, *42*(4), 1–41.

Zhang, Z., Yan, C., Lasko, T. A., Sun, J., & Malin, B. A. (2020a). SynTEG: a framework for temporal structured electronic health data simulation. *Journal of the American Medical Informatics Association*, *00*(0), 1–9. https://doi.org/10.1093/jamia/ocaa262

Zhang, Z., Yan, C., Mesa, D. A., Sun, J., & Malin, B. A. (2020b). Ensuring electronic medical record simulation through better training, modeling, and evaluation. *Journal of the American Medical Informatics Association*, *27*(1), 99–108. https://doi.org/10.1093/jamia/ocz161

Appendix A

Individual feature distributions (visual) (NP)

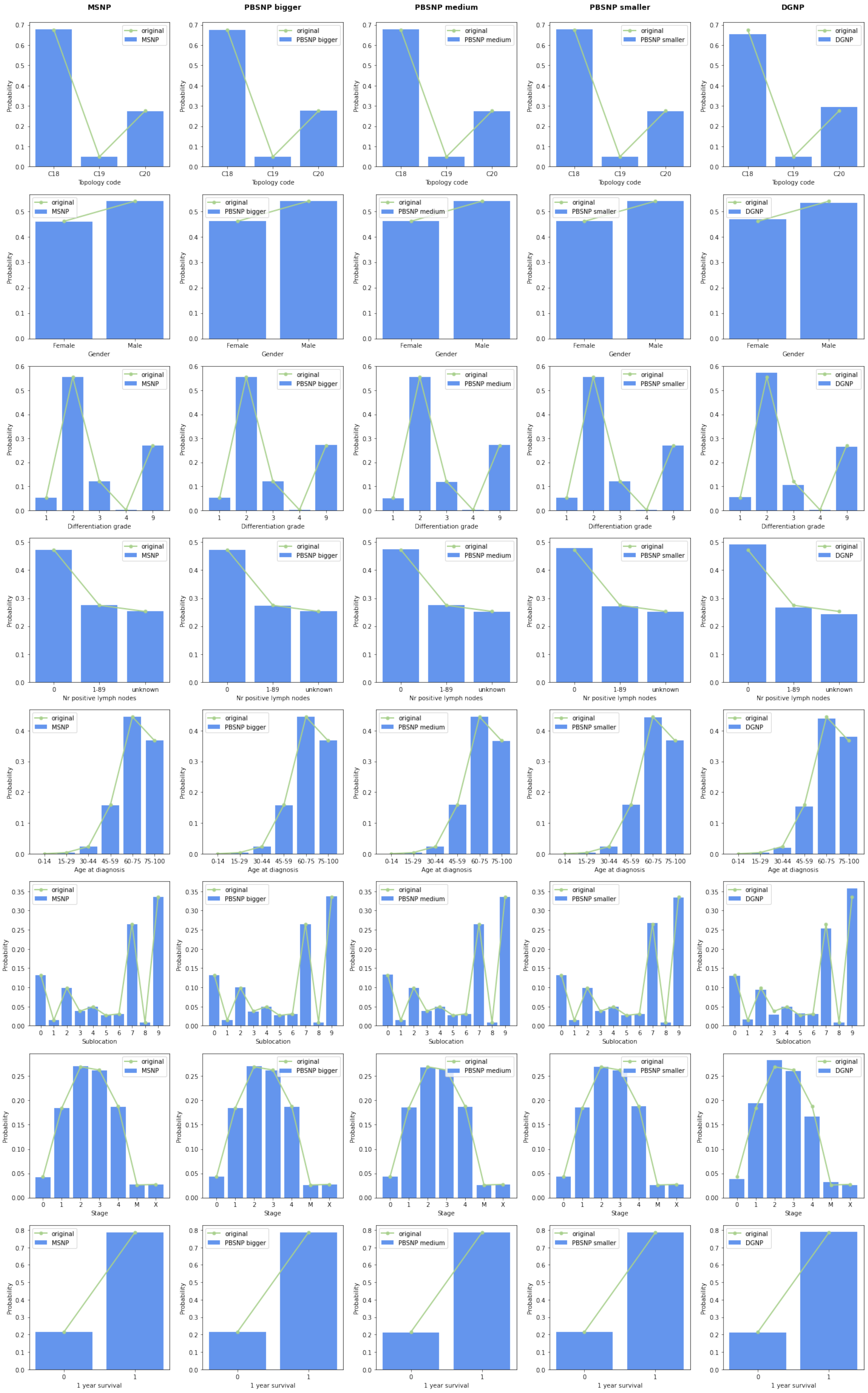
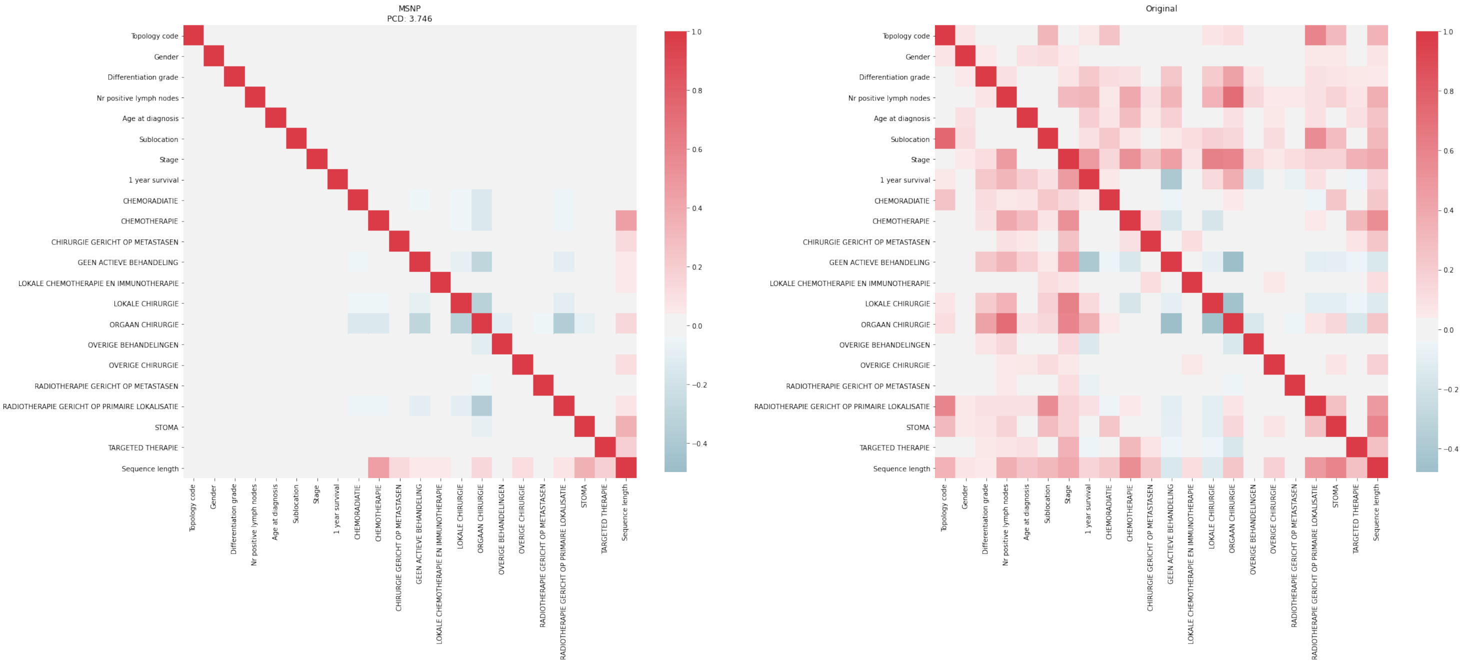


FIGURE A.1: Visual representation of individual feature distributions in the original (line) and synthetic (bar) data set for all non-private (NP) generative methods. Each row represents one feature, each column represents one generative method.

Appendix B

Association matrices synthetic data sets (NP)

B.1 MS-NP

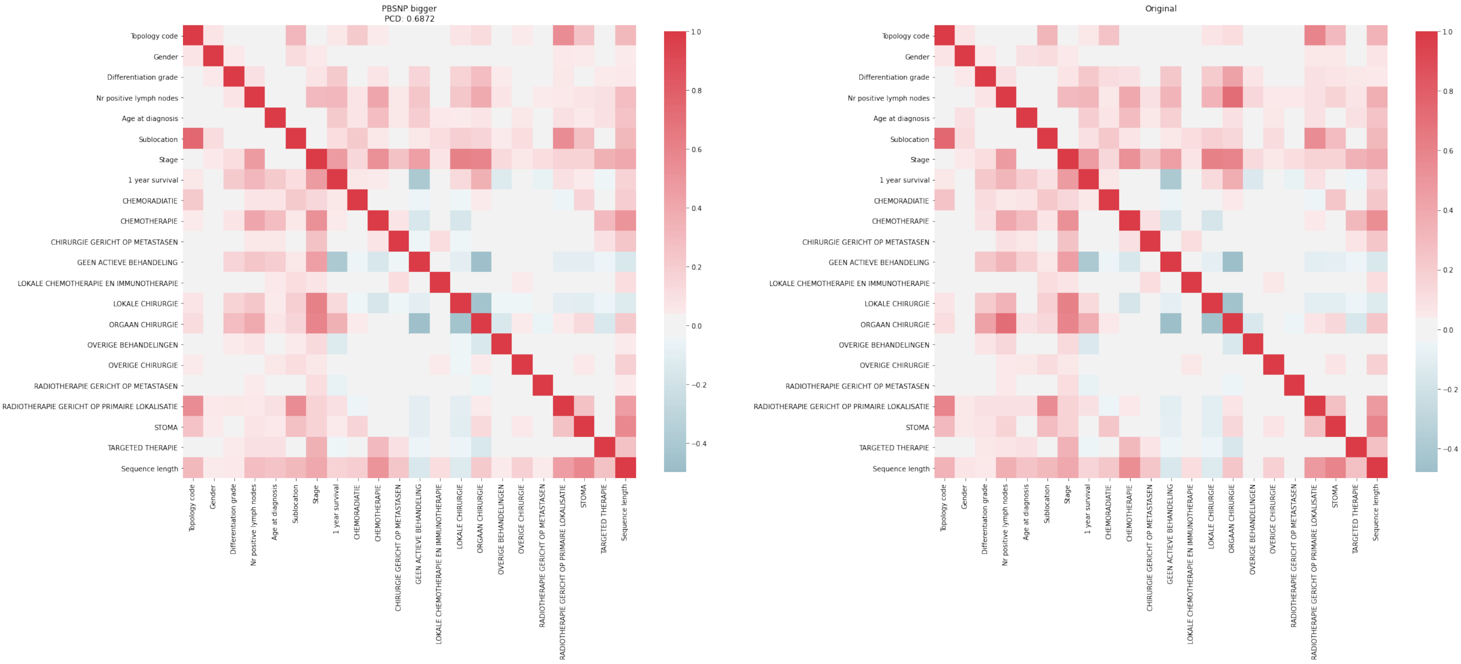


(A) MS-NP

(B) Original

FIGURE B.1: Association matrix of the extended version V of the synthetic data set generated by MS-NP. Exact association values are omitted for confidentiality.

B.2 PBS-NP bigger network size


PBSNP bigger
PCD: 0.6872

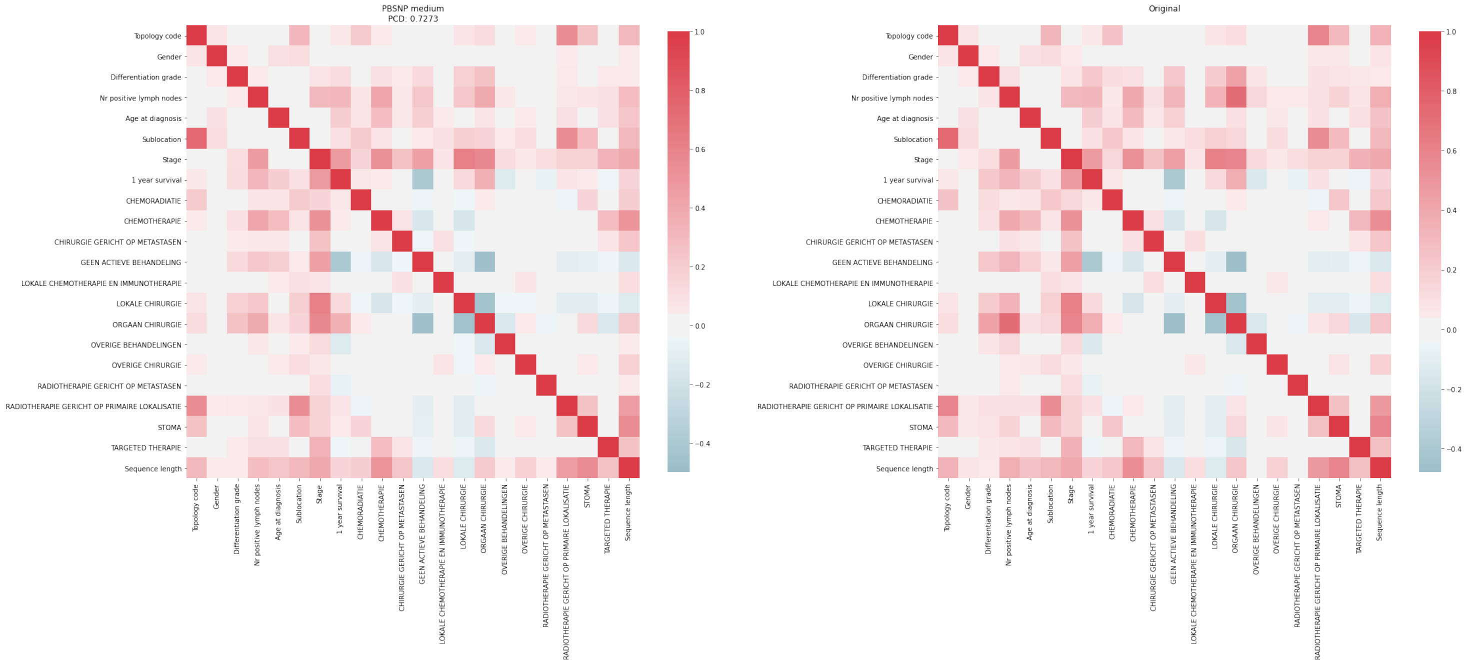
Original

(A) PBS-NP bigger network size

(B) Original

FIGURE B.2: Association matrix of the extended version V of the synthetic data set generated by PBS-NP bigger network size. Exact association values are omitted for confidentiality.

B.3 PBS-NP medium network size

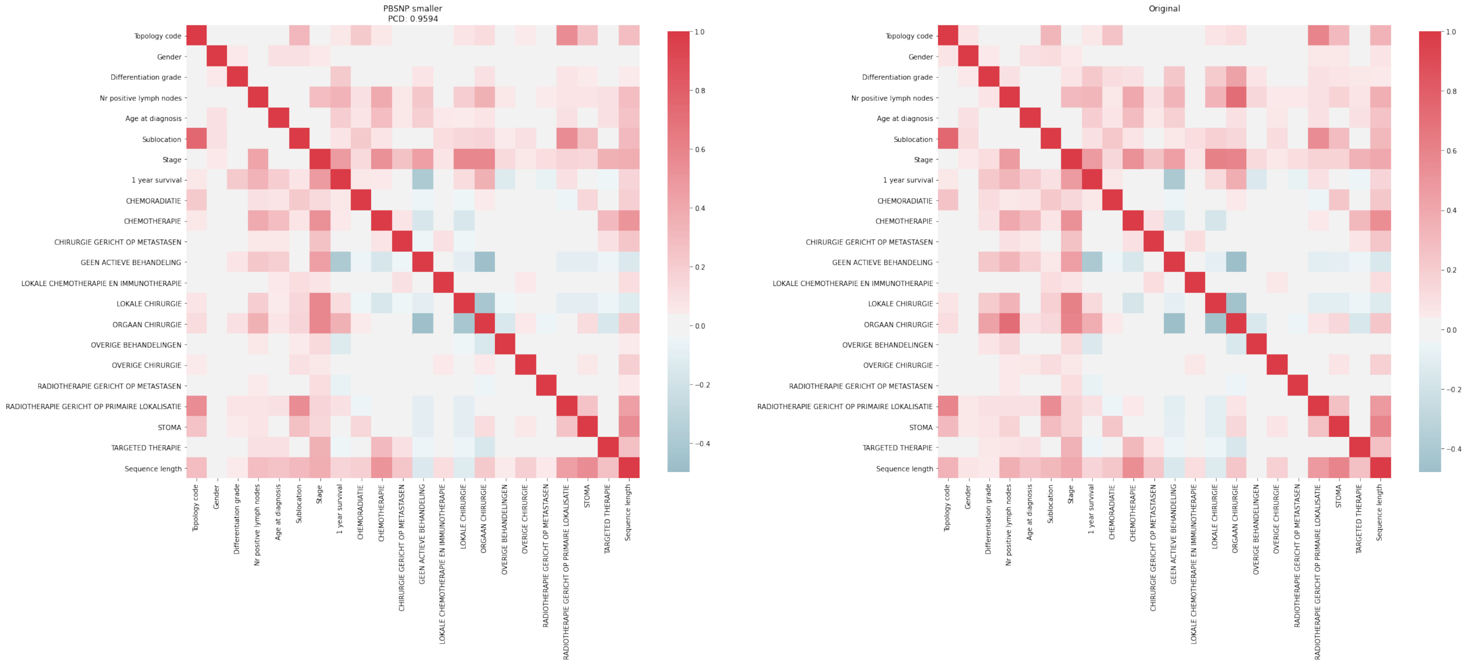


(A) PBS-NP medium network size

(B) Original

FIGURE B.3: Association matrix of the extended version V of the synthetic data set generated by PBS-NP medium network size. Exact association values are omitted for confidentiality.

B.4 PBS-NP smaller network size

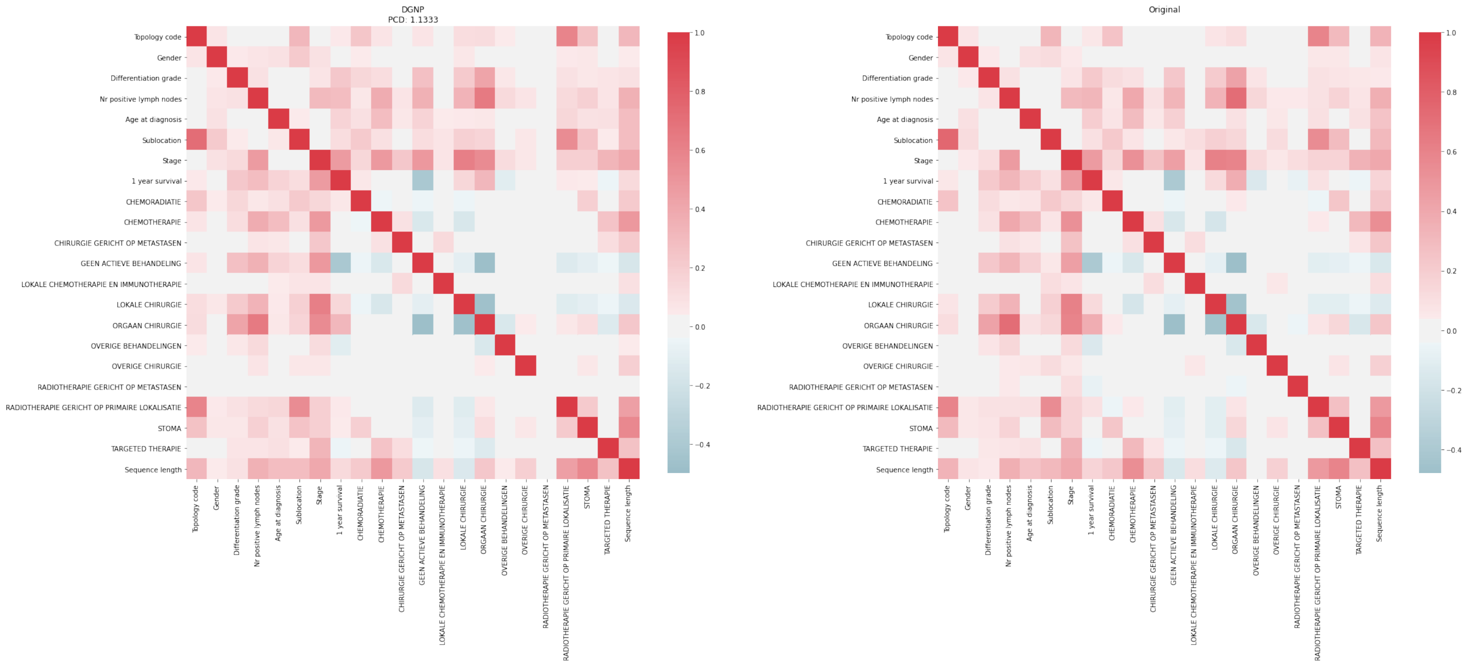


(A) PBS-NP smaller network size

(B) Original

FIGURE B.4: Association matrix of the extended version V of the synthetic data set generated by PBS-NP smaller network size. Exact association values are omitted for confidentiality.

B.5 DG-NP



(A) DG-NP

(B) Original

FIGURE B.5: Association matrix of the extended version V of the synthetic data set generated by DG-NP. Exact association values are omitted for confidentiality.

Appendix C

Individual feature distributions (visual) (DP)

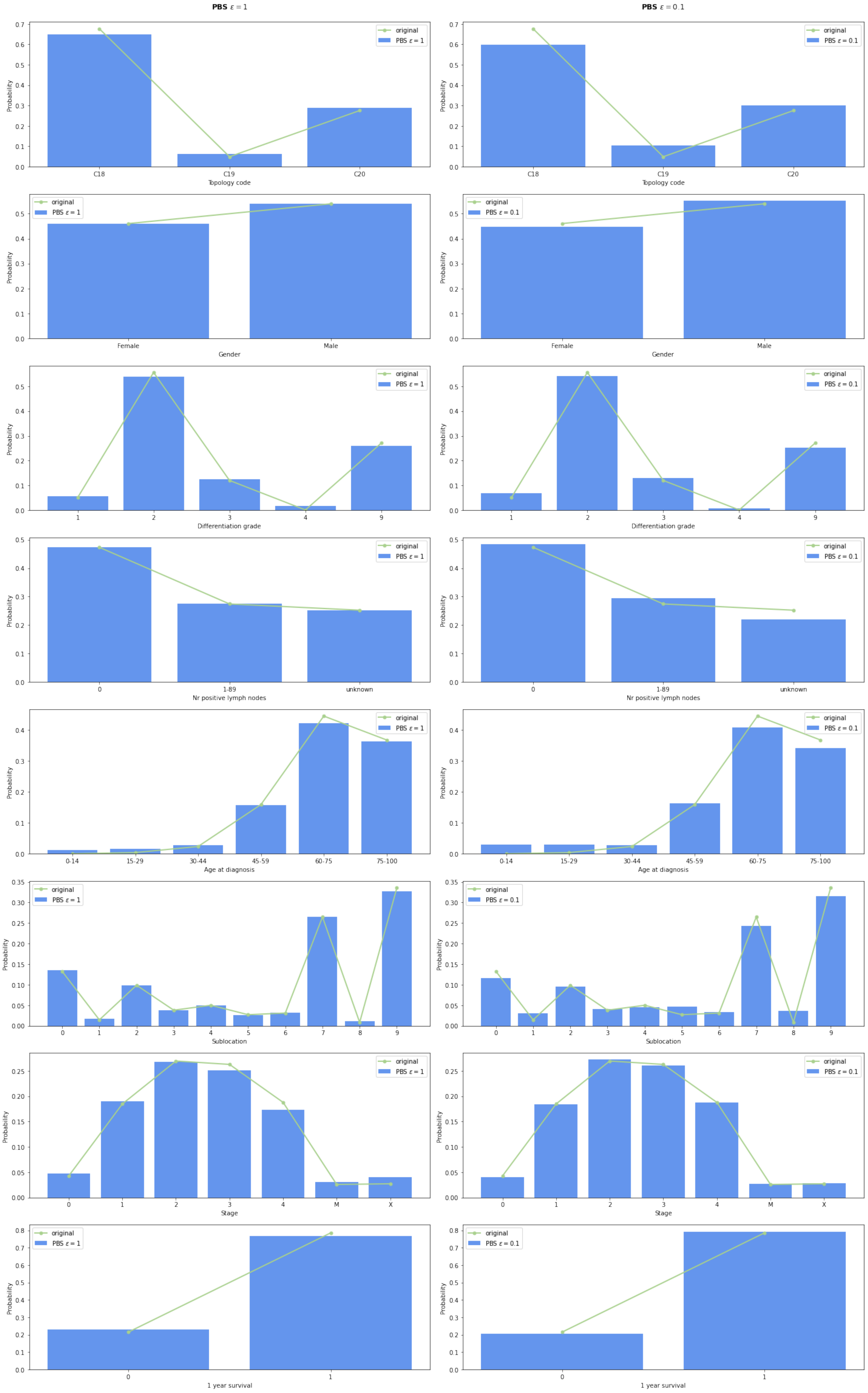
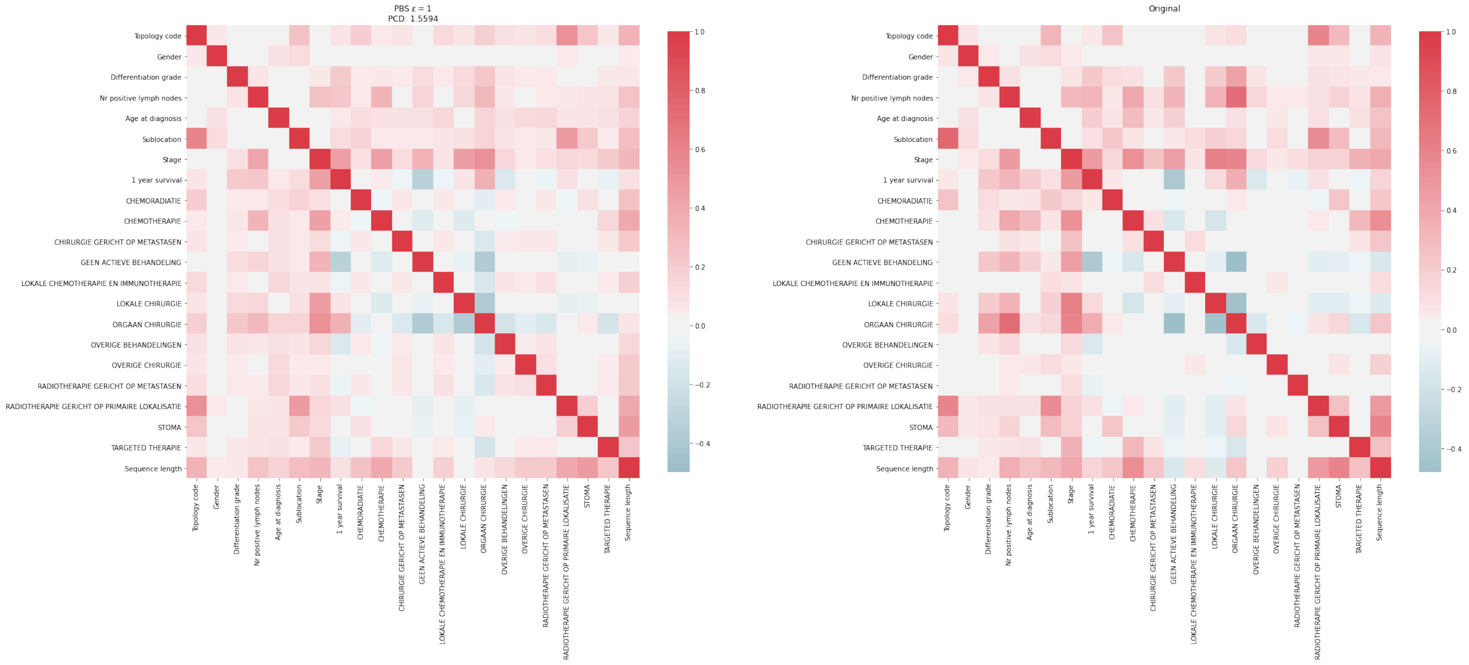


FIGURE C.1: Visual representation of individual feature distributions in the original (line) and synthetic (bar) data set for PBS $\varepsilon = 1$ and PBS $\varepsilon = 0.1$. Each row represents one feature, each column represents one privacy level.

Appendix D

Association matrices synthetic data sets (DP)

D.1 PBS $\varepsilon = 1$

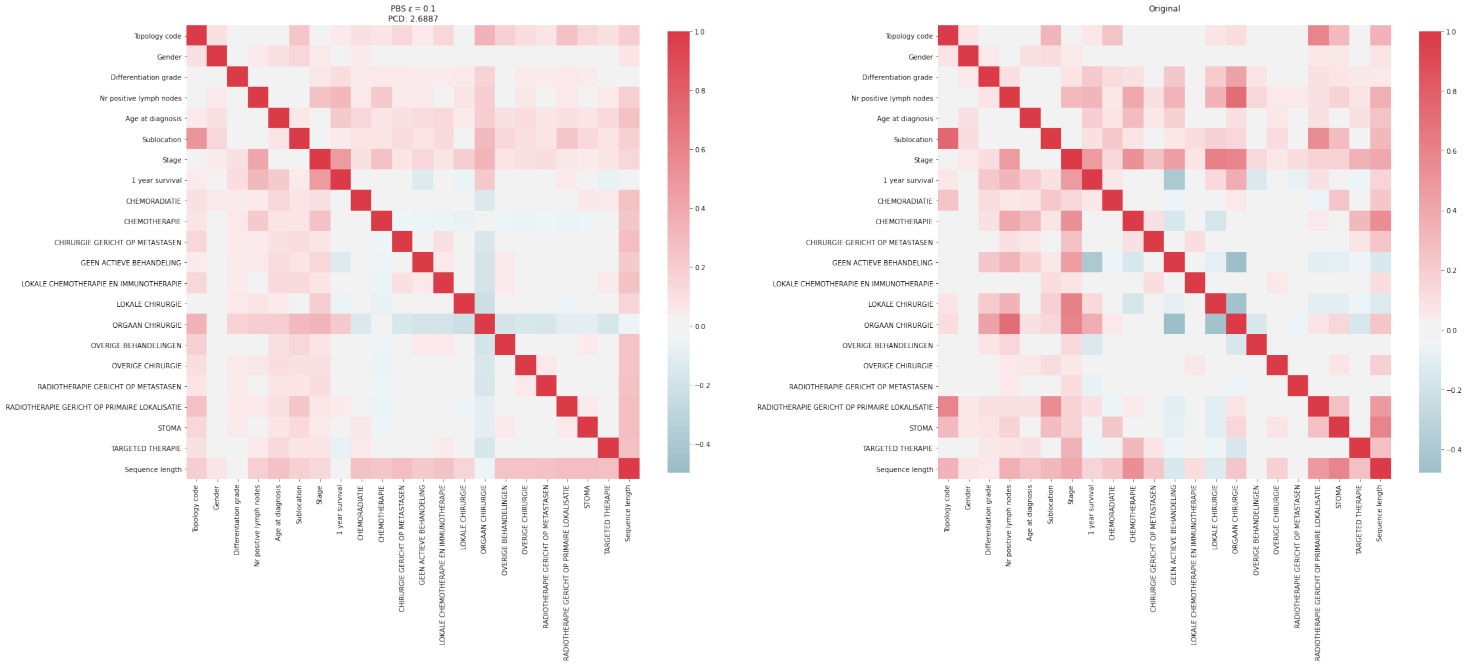


(A) PBS $\varepsilon = 1$

(B) Original

FIGURE D.1: Association matrix of the extended version V of the synthetic data set generated by PBS $\varepsilon = 1$. Exact association values are omitted for confidentiality.

D.2 PBS $\varepsilon = 0.1$



(A) PBS $\varepsilon = 0.1$

(B) Original

FIGURE D.2: Association matrix of the extended version V of the synthetic data set generated by PBS $\varepsilon = 0.1$. Exact association values are omitted for confidentiality.