TU WIEN Informatics

Federated Generation of Synthetic Tabular Data

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Logic and Computation

eingereicht von

Daniela Martinez Duarte, BSc
Matrikelnummer 12045638

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Privatdoz. Mag.rer.soc.oec. Dipl.-Ing. Dr.techn. Edgar Weippl
Mitwirkung: Univ.Lektor Mag.rer.soc.oec. Dipl.-Ing. Rudolf Mayer

Wien, 2. September 2024

Daniela Martinez Duarte	Edgar Weippl

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

TU WIEN Informatics

Federated Generation of Synthetic Tabular Data

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Logic and Computation

by

Daniela Martinez Duarte, BSc
Registration Number 12045638

to the Faculty of Informatics

at the TU Wien

Advisor: Privatdoz. Mag.rer.soc.oec. Dipl.-Ing. Dr.techn. Edgar Weippl
Assistance: Univ.Lektor Mag.rer.soc.oec. Dipl.-Ing. Rudolf Mayer

Vienna, September 2, 2024

Daniela Martinez Duarte	Edgar Weippl

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Daniela Martinez Duarte, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 2. September 2024

Daniela Martinez Duarte

Acknowledgements

I would like to express my sincere gratitude to those who have supported and contributed to the completion of this master's thesis. This accomplishment would not have been possible without you.

First, I would like to express my deepest gratitude to my family for their unconditional love, support, and understanding throughout my academic journey. Their encouragement has been invaluable, inspiring me to pursue my goals with determination and resilience.

I would like to thank my supervisor, Edgar Weippl, for his support throughout this work. I would also like to thank my co-supervisor, Rudolf Mayer, for his patience, expertise, and encouragement throughout this process.

Lastly, I thank all my friends and co-workers for their constant support and encouragement. I also want to give a special thanks to my boyfriend for his unconditional support. Your words of motivation and understanding have made a significant difference in this journey.

Abstract

Machine learning (ML) models have been demonstrated to be beneficial in various domains. However, their application remains severely limited due to concerns about (1) using personal data for training ML models and (2) exchanging data between different organizations, like hospitals and banks. Both cases might lead to privacy breaches and disclosure of sensitive information.

In this work, we tackle both problems simultaneously by generating synthetic data in a federated learning manner. Previous work in this field primarily addresses image data generation, while we focus on tabular data, which is more relevant for sensitive data domains. In particular, we proposed adapting two centralized tabular data generation methods, Bayesian Networks and Variational Autoencoders, to the federated setting with a novel aggregation approach applied specifically to Bayesian Networks. We perform an exhaustive evaluation of the generated synthetic on three datasets in terms of fidelity, utility, and privacy. Further, we demonstrate how the data performance changes depending on data partition among clients participating in federated learning and how the number of clients impacts the results. Our results suggest that, in many cases, the proposed methods in federated settings perform similarly to those in centralized settings and outperform local data generation. However, the imbalance among clients significantly affects the synthetic data generated by Variational Autoencoders.

Contents

CHAPTER 1

Introduction

1.1 Motivation

The vast increase in the amount of data generated and collected in today's digital world has been an important aspect of the development of innovative machine learning (ML) applications in various domains. For example, ML-based solutions are leveraged in healthcare to enable personalized treatments, aid drug discovery, and improve disease diagnosis [BM20]. In other domains, such as finance, ML approaches help to improve customer satisfaction and enhance operational efficiency [PRALP+23]. Many of these approaches rely on having access to a sufficient amount of high-quality data. However, researchers and organizations encounter several obstacles when accessing and sharing data in practice.

A significant obstacle in this regard arises when working with data that may contain sensitive information related to an individual. This not only raises privacy concerns but is subject to strict data protection regulations like the EU's General Data Protection Regulation (GDPR). For example, health data used for ML applications may include personal patient information such as medical conditions, treatments, or insurance data. Unauthorized access to such sensitive information directly compromises individual privacy. Therefore, organizations must guarantee adequate measures to comply with these regulations [BDH18]. These measures often involve expensive and time-consuming processes, such as establishing data processing agreements, obtaining ethical approvals, and conducting privacy risk assessments, which makes the research process more complex. Another barrier for organizations and researchers is data silos. Data is sometimes spread across numerous geographical locations or multiple institutions, and centralized data access can be challenging or even impossible due to the legal barriers [FTPR+21]. Therefore, the amount and quality of data available might be insufficient for ML applications.

To address these challenges, several privacy-preserving approaches have been proposed, including anonymization methods (e.g., k-anonymity), synthetic data generation (SDG),

federated learning (FL), and differential privacy (DP). Among these, SDG is one approach that has gained popularity over the last few years [MAK+23, FV22]. Synthetic data is artificially generated data that mimics the properties of real-world data [FV22]. It can be generated by manually developed rules or in an automated way using a statistical model or algorithm [JSH+22]. Usually, the choice of the model depends on the data type, the domain of interest, and the specific task [FB23]. Synthetic data is particularly appealing for organizations and researchers because it offers a promising solution for privacy-preserving data release, serves as an alternative for testing and evaluating ML pipelines, and is also helpful for data augmentation when data is scarce [JSH+22]. However, synthetic data relies on real data, which is problematic with data silos because it cannot be easily centralized. In this case, synthetic data would need to be generated for each silo, and it may not fully capture the diversity and actual distribution of real-world data.

Federated learning (FL) is an alternative privacy-preserving approach that specifically addresses the problem of data silos [WZL+23]. The approach enables several clients, such as hospitals or banks, to jointly train an ML model without sharing their private data [KMA+21]. Instead, each client trains a model locally and shares the parameters with a central server. The server then aggregates the clients' parameters to obtain a global model. This approach has become popular in many applications because it can achieve results comparable to centralized ML [NSU+18, SVG18]. Nonetheless, compared to synthetic data, one disadvantage of FL is that developing and evaluating different ML models is more complex for data analysts [PA22]. In FL, an additional effort must be carried out to adapt the ML pipeline. This can be particularly challenging for some preprocessing steps and models, adding computational overhead to the training. However, one potential solution that can alleviate both the problem of data sharing and data silos is to use FL to generate synthetic data. The advantage of this approach is that the synthetic data generator model needs to be adapted only once. Afterward, synthetic data can be used to run multiple ML pipelines.

1.2 Problem Definition

Federated learning for generating synthetic data is an emerging research area in privacy-preserving data publishing (PPDP) [LEA23]. This approach enables organizations to generate more representative and diverse synthetic data without sharing their private data [LEA23]. The synthetic data can later be used for different purposes (e.g., privacy-preserving data release and data augmentation). The process works similarly to many traditional FL approaches. The main characteristic is that in this case, the model is a generative model. This means that the model is specifically trained to generate new data that resembles the statistical properties of the clients' data. Furthermore, instead of evaluating the model in, e.g., a predictive task, the synthetic data generated by the model is evaluated in three different dimensions (fidelity, utility, and privacy) to ensure that the data is useful for further analysis and tasks while not disclosing sensitive information. An overview of this approach is depicted in Figure 1.1.

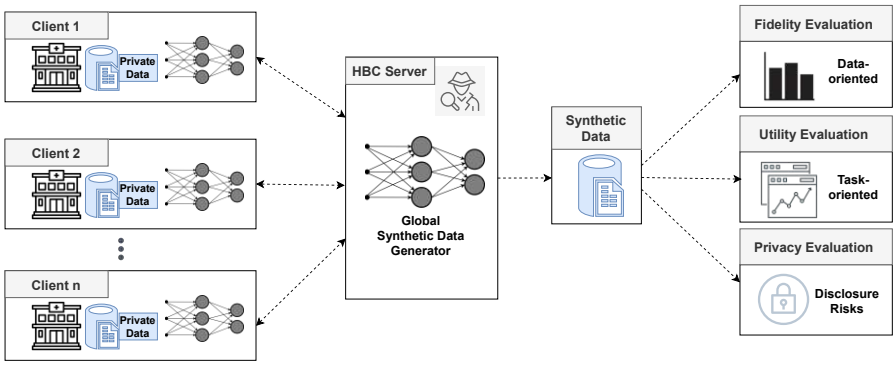


Figure 1.1: Overview of a synthetic data generation pipeline in a distributed setting: clients are organizations or companies (e.g., hospitals), and they collaborate together to learn a global generative model without sharing their private data but instead sharing parameters of a local model. The global model is used to generate synthetic records, which are then evaluated in terms of fidelity, utility, and privacy to ensure the reliability of the data for further tasks while mitigating disclosure risks.

While most of the research in this area pays attention to image data [BUS+22, CQZ+20, TF20], tabular data is the most common type of data in real-world applications [SZA22] and has not been extensively investigated for several scenarios in the federated setting (e.g., data that is not independent and identically distributed (i.i.d.), multi-modal distributions, mixed data types, multiple generative models). Unlike tabular data, image data is homogeneous in several characteristics (size, value ranges, ...) and is characterized by spatial relationships. These characteristics make images suitable for synthetic data generation with deep neural networks [BLS+22], which can be easily adapted to the federated setting. Furthermore, as the data is homogeneous, the preprocessing steps can be performed locally, reducing the process's complexity.

In contrast, tabular data typically contains a combination of discrete and continuous attributes. Continuous attributes might have multiple modes, and discrete attributes might be highly imbalanced [DLH+23]. Consequently, generating tabular data can be more challenging, especially in the federated setting. Adapting traditional generative models to the federated context may be necessary for certain scenarios, a process that is not always straightforward for several types of models. Additionally, since tabular data can contain sensitive information and FL can still be exposed to inference attacks [LXW22], additional privacy-preserving technologies might be required in the process.

In this thesis, we adapt two commonly employed methods for generating synthetic tabular data in centralized settings, Bayesian Networks (BNs) and Variational Autoencoders (VAEs), to the federated setting. We thoroughly evaluate the quality of the synthetic data generated across various data and client scenarios, including different numbers of clients and different client partitions (i.i.d and non-i.i.d distributions). To benchmark the performance of the generative models, we compare the results against two different baselines: centralized learning and local learning, i.e. each client generating their

own synthetic from their own original data. Furthermore, we consider three different dimensions for evaluating synthetic data (fidelity, utility, and privacy), analyze the trade-offs between the dimensions, and investigate additional privacy-preserving techniques, such as differential privacy. We also analyze the challenges in the adaptation from the centralized setting to the federated setting of the synthetic data generation methods in terms of preprocessing, training, hyperparameter tuning, and evaluation.

1.3 Research Questions

We address the following research questions in this thesis:

1. **To what extent is federated synthetic tabular data useful?**

   We investigate the quality of the synthetic data generated using BNs and VAEs in a federated setting in terms of fidelity and utility. Fidelity is evaluated with respect to the univariate, bi-variate, and multivariate distributions using the following metrics: Hellinger distance (HD), Pairwise correlation difference (PCD), Propensity score (PS), and Log-Cluster (LC). On the other hand, utility is evaluated using two well-known analyses: Train Real Test Real (TRTR) and Train Synthetic Test Real (TSTR).

   a) **How does federated synthetic tabular data compare with centralized synthetic data and local synthetic data from a single client?**

      This question aims to explore the quality of the synthetic data generated in the federated setting with respect to two baselines: the centralized setting and the local setting. The centralized setting refers to the scenario where real data is centralized, and a generative model is trained directly from this data. Meanwhile, the local setting refers to the scenario where each client generates synthetic data with their own local data. For the local setting, we average the results for all the clients for each of the metrics. The main goal is to observe whether federated results achieve similar performance to centralized results and outperform the average local results.

   b) **How do fidelity and utility of federated synthetic tabular data compare for the different synthetic data techniques and datasets?**

      This question aims to investigate which technique (BNs or VAEs) provides better performance in terms of fidelity and utility in the federated setting for different datasets. We use the aforementioned metrics for fidelity and utility for comparison, and additionally, we provide visual comparisons for the quality of the synthetic data. The relationship between fidelity and utility is also analyzed.

2. **To what extent is the federated generation of synthetic tabular data sensitive to hyperparameters and data distribution?**

This question analyzes how the quality of the synthetic data generated in the federated setting is affected under different parameters and data distribution scenarios. We use fidelity and utility metrics to quantify the impact.

a) **To what degree do the training hyperparameters used in the federated synthetic data generator affect the fidelity and utility?**
We consider hyperparameters affecting the aggregation or local models to determine the federated synthetic data generator's sensitivity. For the BN, we focus on parameters in the genetic algorithm, such as the source genes, the aggregation interval, and the gene pool size. Additionally, for the VAE, we concentrate on hyperparameters affecting the model's architecture and training, including the number of epochs, the optimizer, batch size, and local epochs.

b) **To what extent do the number of clients and the data heterogeneity affect the fidelity and utility of synthetic tabular data generated in a federated setting?**
To determine the impact of the data distribution among clients in the federated setting, we simulate different client scenarios by changing the number of clients and modeling i.i.d and non-i.i.d partitions. Then, we compare the fidelity and utility scores across various scenarios.

3. **To what extent does federated synthetic tabular data preserve privacy?**
Synthetic data is widely used to mitigate privacy risks. In particular, it mitigates re-identification risk because there is no direct mapping between a real and a synthetic record. However, SDG models can overfit the training data and generate records that closely resemble training data records, which, in the end, turns out to produce high utility but at the cost of privacy [AVBSvdS22]. Furthermore, some works have shown that synthetic data can still be vulnerable to inference attacks [SOT20]. We aim to explore the risk of disclosure of synthetic tabular data generated in a federated setting using distance-based metrics and attribute disclosure attacks.

a) **To what extent do records in federated synthetic tabular data resemble those in real data?**
This question investigates how closely the synthetic data generated in the federated setting resembles the training records used by the clients. To measure this, we compute the minimum distance from a synthetic record to the samples in the training data from each client. We check whether there are exact matches and then compare the distances to samples from a hold-out set following the approach proposed by Platzer and Reutterer [PR21]. The idea is to assess how well the SDG model generalizes insights from the real data without learning the specific details. The expected result is that the distance between real and synthetic records is similar regardless of whether the data was used or not as input for synthetic data generation.

b) **To what extent does federated synthetic tabular data prevent attribute disclosure?**

This question explores attribute disclosure risks in the synthetic data that is generated in a federated manner. For this purpose, we define a scenario for each dataset following the approach in [HME20], which selects a set of quasi-identifiers and a sensitive attribute as the target and formulates the attribute disclosure problem as a classification task. Then, we compare the results in terms of predicting the target attribute accuracy with the original and synthetic data. In this case, the original data is used as an upper bound to estimate the disclosure risk, and the dummy classifier is used as a lower bound.

1.4 Methodology

This section outlines the methodology used to accomplish this thesis's main objective and to address the research questions proposed.

1.4.1 Literature Review

The literature review follows a restricted version of Kitchenham's guidelines [KC07] and covers related works in the privacy-preserving data publishing (PPDP) field, mainly focusing on synthetic data and federated learning. This is relevant because it provides the foundation for developing this thesis. In particular, we conduct the following steps as part of the review:

1. Research the state-of-the-art methods for generating synthetic tabular data in a centralized setting and select commonly used methods.

2. Analyze the evaluation metrics and experimental setup used to assess the quality of synthetic tabular data generated with those methods.

3. Research state-of-the-art papers using federated learning to generate synthetic data, along with relevant related papers on similar models for other applications to investigate different aggregation techniques.

4. Review state-of-the-art papers using additional techniques to enhance federated learning and synthetic data privacy.

1.4.2 Experiment design and implementation

For conducting the experiments and evaluating the results in this thesis, we use the Cross-industry standard process for data mining (CRISP-DM) [WH00] as a guideline and perform the following steps.

Data selection and SDG models selection

We identify datasets commonly used in the synthetic data literature and select those containing individual-level information from various sensitive domains. On the other hand, for the SDG model selection, we consider the following two criteria: (1) Select models that operate on fundamentally different principles and (2) Select models that are commonly used in the centralized setting for tabular data but have not been extensively studied in the federated setting.

Design and implementation of the synthetic data models in FL

The design of the synthetic data models in FL involves considering what the clients will share with the server, identifying whether this can generate a potential privacy issue, and determining a strategy to aggregate the clients' shared information. To decide on these aspects, we inspired our solution of the BN in two previous works [HME22, DFDCK+23] and for the VAE model, we follow the traditional approach in federated learning (FL) of aggregating neural network models using FedAvg [MMRyA16]. The implementation is then carried out using the framework Flower.

Evaluation and comparison of the results

For the two models adapted to the federated setting (BN and VAE), the quality of the generated synthetic data is evaluated in terms of fidelity, utility, and privacy, considering multiple scenarios (different numbers of clients, data partitions, and hyperparameters) as described in the research questions. The results are then compared against centralized and local baselines and between the models to determine if they meet the desired performance.

1.5 Structure of the work

The remainder of this thesis is structured as follows: Chapter 2 provides background knowledge on approaches for privacy-preserving data sharing that are relevant to the thesis and includes related work on privacy-preserving data synthesis in distributed settings. Chapter 3 and Chapter 4 cover the two synthetic data generation methods investigated in this work, namely Bayesian Networks and Variational Autoencoders, and present the approach proposed to adapt these methods to the distributed setting. Chapter 5 describes the methodology used for the experimental setup. Chapter 6 presents the experiments' results and analysis. Chapter 7 summarizes the main contributions of the thesis and presents possible future work options.

CHAPTER 2

Background and Related Work

This chapter covers background knowledge and related work relevant to this thesis. First, we provide an extensive overview of synthetic data, including different generation methods and evaluation metrics used in this work. Furthermore, we review additional privacy-preserving approaches, specifically federated learning, differential privacy, and secure-multiparty computation, which can enhance privacy for synthetic data generation in distributed settings. Finally, we discuss related work on privacy-preserving data synthesis of tabular data in distributed settings.

2.1 Synthetic Data

Synthetic data is artificial data that mimics the structure and properties of real-world data [EEMH20]. Synthetic data generation consists of training a model on real (also called original) data and then sampling from the model to generate new (artificial) data. Different models can be used in the generation, including statistical, probabilistic, and deep learning models. However, depending on the data type, the domain of interest, and the specific task, some models might be preferred over others [JSH+22]. Synthetic data mitigates privacy risks, which is key in applications where privacy or data protection regulations are a major concern.

Synthetic data can also be used for data augmentation. In the industry, for example, multiple applications have benefited from augmented synthetic data. Jain et al. [JSP+22] showed that synthetic augmented data improves the performance of deep learning models in classifying surface defects. Khan et al. [KHK21] demonstrated that synthetic data augmentation through virtual sensors provides a better generalization of original data and improves the performance of deep learning-based models for fault diagnosis of rotating machines. Other applications using synthetic data augmentation include sensory anomaly detection in industrial robots [LDQ+22], fiber layup inspection in the aerospace industry [MMSG21], and automated quality inspections of structural adhesive

applications for automotive parts [PGMB21]. Another interesting application of synthetic data is balancing target classes in imbalanced datasets. This reduces the bias in the ML models and improves their fairness and trustworthiness [JSH+22].

In this thesis, we focus on synthetic data as a replacement for real data, specifically for applications with sensitive tabular data. Therefore, the use of synthetic data for augmentation will not be explored as it is outside the scope of this work.

2.1.1 Types of Synthetic Data

Synthetic data can be classified into three main categories: fully synthetic, partially synthetic, and hybrid [SM17].

- **Fully Synthetic Data** is completely generated, meaning it does not include any original records. This concept was introduced in 1993 by Rubin [Rub93], who leveraged multiple imputation theory to generate fully synthetic data. Subsequent work proposed in this regard uses parametric statistical models and, more recently, leverage techniques from ML [Rei23].
- **Partially Synthetic Data** is generated by replacing sensitive attributes in a real dataset with synthetic values. This approach was introduced by Little [Lit93] as an imputation mechanism to protect the confidentiality of respondents in census and surveys, and was further developed by Reiter [Rei03]. The risk of re-identification in partially synthetic data is higher than in fully synthetic data because it contains real data. However, in terms of utility, partially synthetic data is, in most cases, superior to fully synthetic data since less information is lost from the real data [DBR07].
- **Hybrid Synthetic Data** is generated by combining real and synthetic records, similar to partially synthetic data. The difference in this case is that fully synthetic data is generated first, and then random records from the real data are chosen and merged with the closest record in the synthetic data. The benefit of this approach is that it provides a better trade-off between privacy and utility compared to the other categories [SM17]. However, it is more computationally expensive.

2.1.2 Synthetic Data Generation Methods

Literature proposes multiple approaches for generating synthetic data. Hernandez et al. [HEA+22] classified these approaches into three main categories: classical, deep learning, and others. The classical approaches encompass statistical and probabilistic models such as Bayesian Networks (BNs) or copulas and ML models such as support vector machines (SVMs) or classification and regression trees (CART). The deep learning approaches are based on neural networks, such as generative adversarial networks (GANs), variational autoencoders (VAEs), or diffusion models. Other approaches refer to generation methods that rely on various modules or steps like SynSys [DC19], an approach that combines hidden Markov models (HMMs) and regression algorithms in several steps to generate

synthetic health data. In this thesis, we mainly focus on the generation of tabular data using classical and deep learning methods. Below, we provide an overview of relevant methods:

- **Copulas** are probabilistic models that describe the dependence between random variables. Copulas are particularly useful because they separate the marginal distributions of variables from their dependence structure and output the joint probability distribution that best fits the structure [EEMH20]. Generating synthetic data using copulas from a dataset with n variables denoted as $Z_1, \ldots, Z_n$ comprises two main steps [MNH21]. In the first step, the corresponding marginal distributions $F_1, \ldots F_n$ are identified for each variable. A way to accomplish this is by estimating their empirical distribution function. The second step creates the model representing the joint probability distribution based on Sklar's Theorem [Skl59]. This theorem states that a joint probability distribution $F(Z_1, \ldots, Z_n)$ can be expressed through its univariate marginal distributions and a copula function C. Formally,

$$F(Z_1, \ldots, Z_n) = C(F_1(Z_1), \ldots, F_n(Z_n)) \tag{2.1}$$

  The copula function encodes the dependence between the variables. Synthetic data can be generated by sampling data from the copula. Various families of copulas exist to capture this dependence. One famous family of functions is the Gaussian Copula, which represents the joint probability distribution using a multivariate normal distribution and a correlation matrix. The work by Patki et al. [PWV16] provides an example of applying Gaussian copulas for synthetic data generation in relational databases.

- **Classification and Regression Trees (CART)** is an algorithm commonly used in machine learning for learning a tree-based predictive model. CART can also be used for generating synthetic data, as first suggested by Reiter in 2005 [Rei05]. The idea of CART is to construct a binary decision tree by recursively partitioning a dataset into smaller subsets until a stopping criterion is met (e.g., a minimum number of samples in a leaf node). To decide the partition points, CART uses impurity measures, such as the Gini Index, and identifies the variable and split criteria that minimize the impurity score from the set of input variables. Additionally, CART uses pruning techniques to reduce the tree's complexity. The leaf nodes of the tree determine the outcome of the target variable. CART models are used sequentially to generate fully synthetic data [NRD16]. This implies that variables are synthesized one at a time. The process works as follows: the first variable is generated using random sampling with replacement from its observed values as it has no predictors. The second variable is generated using the first variable as a predictor. The third variable uses the second and first variables as predictors. The following variables are generated similarly, incorporating previous variables as predictors. Previous work has shown that CART models perform well compared to other methods. For instance, Pathare et al. [PMS+23] compared synthetic

data generation techniques for tabular data using two well-known fidelity metrics, propensity and cluster log metric, and showed that the CART performed better for all the datasets.

- **Bayesian Networks** are probabilistic graphical models that use a directed acyclic graph (DAG) to represent the joint probability distribution between a set of random variables. The graph represents each random variable as a node and the conditional dependencies between variables as edges. After the graph is constructed using real data, it is utilized to draw new data samples, which compose a synthetic dataset [PSH17]. An advantage of Bayesian networks is that the resulting model is explainable and can incorporate expert knowledge.

- **Variational Autoencoders (VAEs)** consists of two main components: an encoder and a decoder. The encoder and decoder are deep neural networks trained simultaneously to minimize a loss function. The encoder maps samples from the real data to a latent space. Conversely, the decoder aims to reconstruct the input from the latent space. The loss function measures the reconstruction loss and uses, e.g., the Kullback–Leibler (KL) divergence [FV22].

- **Generative Adversarial Networks (GANs)** are a class of deep learning models capable of generating synthetic data. GANs are based on a zero-sum game between two neural networks, the generator and the discriminator. The generator aims to produce samples that follow the original data distribution. The discriminator aims to identify whether a sample comes from real data or from the generator. This game continues until the discriminator cannot distinguish the real data from the generated data [GPAM$^+$14]. A well-known architecture for generating tabular data is a conditional GAN called CTGAN [FV22], which introduces a mode-specific normalization, a conditional generator, and a sampling method to tackle the challenges of multi-modal distributions and imbalance of categorical columns [XSCIV19].

2.1.3 Evaluation Metrics for Synthetic Data

The evaluation of synthetic data is commonly performed in literature considering three main dimensions: *fidelity*, *utility* and *privacy* [HEA$^+$22, DII22]. *Fidelity* estimates the similarity of the synthetic data with respect to the original data. *Utility* metrics evaluate the usefulness of synthetic data for a specific application compared to the original data. *Privacy* metrics aim to quantify the risk posed by publishing synthetic data instead of original data.

Further dimensions that have been proposed in the literature are *diversity*, and *generalization* [AVBSvdS22]. *Diversity* indicates whether the synthetic data samples covered the variability of the original data. *Generalization* quantifies the extent to which synthetic data copies the original data (e.g., indicating if the model overfits the original data).

Different metrics have been proposed in each of these dimensions. However, there is no direct guideline for choosing specific metrics [DII22]. Some authors have investigated the possibility of defining an All-in-one metric that covers different dimensions to assess synthetic data [CTM+22, AVBSvdS22, DI21].

Chundawat et al. proposed a universal metric named TabSynDex, which results from averaging five different scores [CTM+22]. These scores include fidelity and utility metrics. Dankar et al. proposed reducing four popular fidelity metrics to a unique value using Principal Component Analysis (PCA) [DI22]. Alaa et al. [AVBSvdS22] proposed a 3-dimensional metric (α-Precision, β-Recall, Authenticity) that characterizes fidelity, diversity, and generalization and enables the evaluation of synthetic data at two levels: sample-level and population-level. Instead of providing a unique metric, other works classify the synthetic data as poor, good, and excellent using different dimensions [HEA+23].

Despite the efforts made to evaluate synthetic data in a standardized way, there is still a lack of consensus in the literature. In this thesis, we considered commonly used metrics to evaluate synthetic data in the three main dimensions: *fidelity*, *utility*, and *privacy*. Instead of looking for the best synthetic data generator in the federated setting, we investigate the trade-off between these metrics for two SDG methods using different datasets to provide guidelines for choosing the best method depending on the primary goals of the application.

Fidelity Metrics

Fidelity metrics can be classified into three categories: univariate fidelity, bivariate fidelity, and population fidelity [DII22]. Univariate fidelity metrics determine whether synthetic data preserves the statistical characteristics, structure, and marginal distributions of the attributes in the original data. Bivariate fidelity metrics capture the correlations between attribute pairs. Population fidelity metrics provide a global assessment of the similarity of original and synthetic data distributions.

In this work, we will evaluate the following metrics to assess fidelity, covering the different categories described above:

- **Hellinger Distance (HD)** is an univariate fidelity metric which uses a distance metric to quantify the similarity between two probability distributions. It ranges from 0 to 1, which makes it interpretable. A distance closer to 0 indicates that the distributions are similar, while a distance closer to 1 indicates that the distributions differ significantly. It can be applied to both continuous and categorical data. The average of this metric can be used to determine the similarity of univariate distributions across all variables in synthetic data and real data [DII22]. For two discrete probability distributions $P = (p_1, \ldots, p_k)$ and $Q = (q_1, \ldots, q_k)$ the metric can be calculated as follows:

$$HD(p,q) = \frac{1}{\sqrt{2}}(\sqrt{\sum_{i=1}^{k}(\sqrt{p_i} - \sqrt{q_i})^2} \tag{2.2}$$

In the case of numerical variables, we can use binning techniques to transform them into a discrete form.

- **Pairwise Correlation Difference (PCD)** is a metric that compares pairwise correlations in synthetic and real data by calculating the difference of correlation matrices in terms of Frobenius norm [DII22]. It falls into the category of bivariate fidelity and is defined as follows:

$$PCD(R,S) = ||Corr(R) - Corr(S)||_F \tag{2.3}$$

where R and S correspond to the real and synthetic datasets, respectively, and Corr$(\cdot)$ represents a matrix of correlation coefficients. These coefficients are calculated using different formulas depending on the variable's types. When both variables are numeric, the Pearson correlation coefficient is used. When both variables are categorical, the Cramér's V [Cra99] coefficient is used. On the other hand, the correlation ratio is used for numerical and categorical variables. In the implementation, we used the associations function from the Python dython package to estimate the correlation coefficient matrix.

- **Propensity Score** is a popular metric to assess the distinguishability between real and synthetic data through a classification model. This metric was first adapted for synthetic data in [SRN+18] and has since been widely applied for evaluating the fidelity of synthetic data in literature [EEMFEH22, DI21, EF23, PMS+23]. To compute this score, the real and synthetic data are combined into a new dataset with an additional label indicating the corresponding source dataset of each record. This dataset is used as input to train a classification model. The propensity score for each record is then calculated using the prediction from the classifier. Based on these scores, a metric that can be used to assess population fidelity is the propensity mean square error (pMSE), which can be computed as follows:

$$pMSE = \frac{1}{N}\sum_{i}(\hat{p}_i - c)^2 \tag{2.4}$$

where N is the size of the input dataset, $\hat{p}_i$ is the probability that the record i comes from the synthetic data, and c is the proportion of synthetic records to the number of real records. The pMSE ranges from 0 to 0.25. A value of 0 indicates no distinction between synthetic and real records; in this case, the probability of all records is 0.5. Conversely, a value close to 0.25 occurs when the classifier is confident that a record comes from the synthetic data. Common classification models used to calculate the propensity score are Logistic Regression and CART

models. The work in [DI21] found that calculating the propensity score based on CART models provided a better model for distinguishability. Therefore, we used a CART model in this thesis.

- **Log-Cluster Metric** [GRS+20] is a metric that was introduced to compare the similarity of real and synthetic data in terms of clustering. The metric is computed by merging the real and synthetic data to form a new dataset. The k-means algorithm with a specified number of clusters G is used in the new dataset to perform a cluster analysis. The intuition behind the analysis is that if real and synthetic data are similar, then the distribution of the records in the different clusters is even, and they cannot be distinguished. The metric is calculated as follows:

$$U_c(R,S) = \log\left(\frac{1}{G}\sum_{i=1}^{G}[\frac{n_j^R}{n_j} - c]\right) \tag{2.5}$$

where R and S correspond to the real and synthetic datasets, respectively, n_j^R is the number of real records in cluster j and n_j is the total number of records in cluster j and c is the proportion of real records to the total number of records in the combined dataset. To implement this metric, we first use principal analysis component (PCA) to reduce the dimensionality of the combined dataset to a number of components that explain 85% of the total variance. Then, the transformed dataset is used for cluster analysis. In this step, the k-means algorithm is run for different numbers of clusters ranging from 2 to 20 as performed in [PMS+23], and the one that gives the lower U_c value is selected as the final metric.

Utility Metrics

Utility metrics depend on the specific application where synthetic data will be used. A common way to evaluate utility is to train ML models using synthetic data and compare the results with those trained on the original data. Similar results indicate that synthetic data is useful for performing the same analyses done with original data. Popular strategies proposed in the literature include training on real and testing on real (TRTR) and training on synthetic and testing on real (TSTR) [HEM19, SRRW23, HEA+23]. Different classification metrics in machine learning are then used to evaluate the performance. Figure 2.1 shows a confusion matrix that can be used to derive several of these metrics.

- **Accuracy:** fraction of correct predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.6}$$

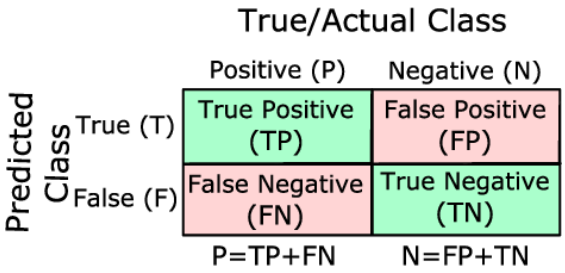


Figure 2.1: Example of Confusion Matrix for Binary Classification. Source: [Tha20]

- **Precision:** fraction of correctly classified positive instances out of the total number of instances predicted as positive.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.7}$$

- **Recall:** fraction of correctly classified positive instances out of the total number of actual positive instances.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.8}$$

- **False Positive Rate (FPR):** fraction of negative instances that were incorrectly classified as positive in a test.

$$\text{FPR} = \frac{FP}{FP + TN} \tag{2.9}$$

- **F1-Score:**: harmonic mean of precision and recall.

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.10}$$

- **Area Under the Receiver Operating Characteristic Curve (ROC AUC)**: quantifies the performance of binary classifiers by computing the area under the ROC curve, which plots the True Positive Rate (TPR) against the False Positive Rate (FPR) across thresholds [HL05].

In this work, we use three classification models, namely Random Forest, Naive Bayes, and k-Nearest Neighbors, to evaluate the machine learning utility. We report the ROC AUC score as a performance metric since it is commonly used for binary classification. Furthermore, we prepare the real and synthetic data using the following preprocessing before training the ML models: for the numeric values, we impute missing values with Standard Scaler, and for the categorical columns, we use label encoding.

Privacy Metrics

Privacy metrics quantify the leakage of information on the data subjects in the real data when sharing synthetic data or giving access to generative models. Several earlier studies have assumed that synthetic data is inherently private since there is no direct mapping between real and synthetic records. However, there is an increasing concern when synthetic data is used in specific application domains, particularly healthcare, without conducting a proper privacy evaluation [MAK+23]. To date, there has been little agreement on how to measure the privacy of synthetic data, and several different metrics have been used across the literature. These metrics can be divided into two main groups: distance and similarity metrics, and attack-based metrics. Distance and similarity metrics rely on the idea that leakage can occur when synthetic records are too close or similar to real records, especially to those representing minority classes, and that this could lead to re-identification. On the other hand, attack-based metrics rely on a threat model for designing potential attacks. This model defines the attacker's and defender's profiles and capabilities. One potential attack considered in this thesis is attribute disclosure, which occurs when the released synthetic dataset increases the knowledge about sensitive attributes.

In this work, we will employ the following metrics to assess privacy, covering the two categories mentioned above:

- **Distance to Closest Record (DCR)** measures the Euclidean distance between any record in the synthetic dataset and the nearest neighbor record in the original dataset. The greater the DCR value, the higher the privacy level. A DCR value of 0 indicates an exact match in synthetic data. Some works use all distances' mean and standard deviation as a reference for analyzing privacy [HEA+23]. A higher mean distance and lower standard deviation indicate better privacy.
- **Attribute Disclosure (AD)** occurs when an attacker can learn new information about an individual in the real dataset. For example, an attacker that has access to some quasi-identifier attributes (e.g., the age, the gender, the education) of a patient then infers the value of a sensitive attribute using the synthetic dataset. This can occur without necessarily linking the record to an individual.

2.2 Federated Learning

Federated Learning is a privacy-preserving approach that enables multiple clients to train a machine learning model collaboratively from distributed data sources [WZL+23]. Contrary to centralized learning, federated learning does not require data collection in a centralized place. Instead, each client trains a model using their local data, and then, a central server aggregates all the model parameters [KMA+21]. The main objective of this approach is to address the problem of data silos, where data is collected and stored distributed across various locations, making sharing difficult due to privacy risks

and data protection regulations [LDCH22]. Federated Learning has become popular in several application settings, as it can achieve results comparable to centralized learning while reducing privacy risks [NSU+18, SVG18].

A typical federated learning process consists of the following main steps:

1. **Initialization:** The central server initializes the model that will be federated, either randomly or using a pre-trained model.
2. **Client Selection:** The central server selects the clients participating in the federated round. In some cases, all the clients can participate in the round. However, the selection strategy varies depending on the client's characteristics and the specific learning task. For example, the server can select the clients based on specific criteria (e.g., computational power, sufficient data, internet connection).
3. **Broadcast:** The central server distributes the global model to the selected clients.
4. **Client Computation:** Each client trains the global model on their local data and sends the updated model parameters to the server.
5. **Aggregation:** The central server aggregates the parameters received from the clients. Several aggregation algorithms were proposed in the literature. A well-known example is *FedAvg* introduced by McMahan et al. [MMR+17], which computes the average of the client's parameters.
6. **Model Update:** The central server updates the global model with the aggregated parameters. Depending on the type of model used, the learning process may continue iteratively from step 2 until the model converges or a stopping criteria is fulfilled (e.g., maximum number of iterations).

2.2.1 Types of Federated Learning

Federated learning can be categorized into two main categories, depending on the types of clients participating:

- **Cross-silo Federated Learning:** This setting refers to scenarios where clients are typically organizations or companies (e.g., hospitals from various countries, research institutes from different universities) that want to train a model collectively. In such cases, clients are considered reliable and well-known. The number of clients is generally small, typically ranging from 2 to 100, with most clients participating in each federated round [KMA+21].
- **Cross-device Federated Learning:** This setting refers to scenarios where clients are edge devices. The number of clients is usually huge (e.g., up to 10 billion clients), and not all of them participate in each federated round [KMA+21]. Reliability issues are more common with these clients, as they might experience failure or

drop out. Furthermore, given the number of clients, communication is also a major hurdle in this case.

Alternatively, federated learning can also be categorized based on how data samples are partitioned among the clients as follows [ZXB+21]:

- **Horizontal Federated Learning:** This setting refers to scenarios where clients share the same feature space (i.e., the same columns), and the number of samples may or may not differ between clients. For example, different hospitals want to predict the treatment for a specific disease, and all hospitals collect the same patient demographic and clinical information. Still, the number of patients with this disease varies among the hospitals, and they usually do not overlap.
- **Vertical Federated Learning:** This setting refers to scenarios where different clients share the same samples, but the features are not the same, and they aim to learn a model collaboratively without privacy leakage. For instance, this might be the case when two companies (e.g., a bank and an internet company) try to predict information about customer behavior. One company has access to customers' financial information, while the other has access to their purchase data.

Within the scope of this work, we will focus on cross-silo horizontal federated learning. The intersection of these settings is particularly important when different trusted parties collect the same data from various individuals and wish to train a joint machine-learning model. However, data protection regulations and privacy risks prevent them from sharing their data. In particular, the model we aim to train is a generative model for publishing synthetic tabular data that can later be used for further research.

2.2.2 Frameworks for federated learning

Federated learning has gained increased attention in recent years because it was shown it has the ability to achieve comparable performance to centralized learning in several applications, such as prediction on keyboards, prediction of human trajectories, and prediction of mortality rates in heart disease patients [LFTL20].

There has also been an ongoing effort to develop different frameworks to perform federated learning at both research and industrial levels. These frameworks differ on several aspects, including the aggregation methods, the privacy and security mechanisms, the model, device, and machine learning support, as well as the use cases [BKKZ22]. An overview of some of the available open-source frameworks is provided below:

- **FATE**[1] is an open-source project initiated by the company WeBank and later hosted by the Linux Foundation. It provides a secure framework for federated learning at

[1] https://github.com/FederatedAI/FATE

the industry level. This framework is suitable for deployment in production-ready environments and supports standalone and cluster deployments. Additionally, it includes secure computation protocols based on homomorphic encryption and multi-party computation (MPC).

Regarding federated training, FATE supports various federated learning algorithms, including logistic regression, tree-based algorithms, and deep learning algorithms. It also offers federated modules for preprocessing and feature engineering, such as federated sampling, feature binning, and feature scale. The modules can be combined to form a pipeline.

- **PySyft**[2] is an open-source library enabling privacy-preserving machine learning developed as part of the OpenMined initiative. This library provides different techniques to enhance data privacy, including federated learning, differential Privacy, and encrypted computation based on homomorphic encryption and multi-party computation. In addition, it integrates with the major deep learning frameworks PyTorch and TensorFlow. However, the support of ML models is limited.

  To model different clients in the federated environment, PySyft employs virtual workers, which refers to separate processes running on the same machine [RTD+18]. These workers communicate through a standardized communication protocol to simulate FL.

- **Flower**[3] is an open-source framework for implementing federated learning systems at a large scale. It was initiated as a research project at the University of Oxford to provide a tool that is easily extendable, framework-agnostic, and flexible [BTM+22]. The current version of Flower includes implementations of multiple aggregation strategies for performing federated learning proposed in the literature and examples for different use cases.

  Flower's core architecture comprises three main components: the strategy, the client manager for edge clients or virtual clients, and the FL training pipeline as illustrated in Figure 2.2. Edge clients refer to real devices that communicate with a server over a remote execution protocol (specifically, remote procedure call, RPC). In contrast, virtual clients correspond to ephemeral clients managed by the Virtual Client Engine in a resource-aware manner, primarily used for simulation purposes [BTM+20].

  In addition to its architectural components, Flower also offers mechanisms to mitigate data privacy risks. In particular, Flower includes different secure aggregation protocols to prevent the server from inferring client information during model aggregation, such as SecAgg and SecAgg+ [LdGBL21]. Additionally, it offers a differential privacy wrapper, which is currently experimental.

  In this work, we use Flower to perform the federated learning experiments. We selected this framework because it offers comprehensive documentation, facilitates

[2] https://github.com/OpenMined/PySyft
[3] https://github.com/adap/flower

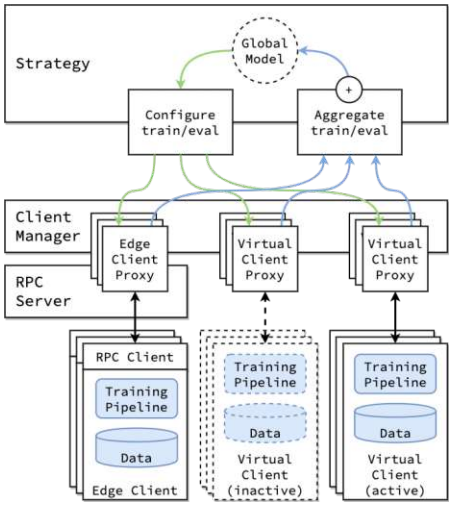


Figure 2.2: Flower's Framework Architecture, taken from [BTM+20]

extension and integration of new modules, and allows running experiments with multiple clients with minimal overhead with the virtual client engine [BTM+20].

2.2.3 Challenges in Federated Learning

Several challenges arise when implementing FL in real-world scenarios. One prominent challenge is data heterogeneity. Usually, the data collected from different clients is non-i.i.d (Non-Independent and Identically Distributed) or varies in size. Compared to i.i.d. settings, this degrades the performance of models trained using federated learning [LDCH22]. Another important challenge is security and privacy. Several works have shown that federated learning is vulnerable to inference attacks [PM20, MSDCS19, HAPC17]. For instance, the information exchanged with the central server can leak information about the clients' data. Additional privacy-preserving methods (e.g., differential privacy, secure aggregation, homomorphic encryption) are thus often needed to mitigate privacy and security risks in FL. Further challenges in FL include communication overhead and computation resources [WZL+23].

This work investigates two main challenges when training synthetic data generators in the federated setting: data heterogeneity and privacy. In particular, we consider a scenario with an honest but curious server. This means we have a server that follows the protocol, but still tries to infer some information from the clients.

2.3 Differential Privacy

Differential privacy (DP) is a widely used concept to define privacy in statistical analysis; it is based on rigorous mathematical guarantees. The concept, formally introduced in 2006 by Dwork et al. [DMNS06], emerges from the idea of learning statistical properties

about a population without compromising the privacy of a single individual. In particular, it ensures that any possible output of an algorithm is equally likely, regardless of whether an individual was part or not of the database(s) used in a study. This is achieved by adding controlled random noise. A formal definition of differential privacy is given below.

Definition 1 *(Differential Privacy [DR+14]) A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|x|}$ satisfies (ϵ, δ)-differential privacy if for all $\mathcal{S} \subseteq Range(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|x|}$ such that $\|x - y\|_1 \leq 1$:*

$$\Pr[M(x) \in S] \leq e^{\epsilon} \Pr[M(y) \in S] + \delta$$

where the probability space is over the coin flips of the mechanism $\mathcal{M}$. If $\delta = 0$, we say that $\mathcal{M}$ is ϵ-differentially private.

The definition contains two privacy parameters, ϵ and δ. The parameter ϵ, denoted as the *privacy budget*, quantifies the privacy loss of applying the mechanism $\mathcal{M}$. The smaller the value of ϵ, the greater the privacy protection, and vice versa. This parameter must be fine-tuned to control the trade-off between privacy and utility. On the other hand, the parameter δ is a probability expressing the likelihood of a privacy breach. Therefore, it is desirable to have small values of δ [DR+14], close to zero, to ensure stronger privacy guarantees.

When applying differential privacy, it is crucial to determine the amount of noise required. Typically, this amount is based on the sensitivity of the statistical function f applied to the data (i.e., the "query"). The sensitivity refers to the largest change in the output of f when a single entry from the input changes [LC11]. For example, the sensitivity of a counting query, such as "How many individuals in the database satisfy condition C?" is one since adding or removing an individual from the database alters the counts at most by one.

Different mechanisms can be used to achieve differential privacy. One of the simplest mechanisms is the *randomized response* [DR+14], which was proposed for surveys collecting responses on a sensitive topic by introducing random noise based on a chance mechanism (e.g., flipping a coin). Another approach is the *Laplace mechanism* [DMNS06], which adds noise drawn from the Laplace distribution to the query response. For categorical outputs, a well-known mechanism is the *exponential mechanism* [MT07], which assigns a higher probability to the output elements that maximize a previously chosen utility function.

Apart from the mechanisms, a key property for the design of differentially private algorithms is composition. This property enables the combination of multiple differentially private mechanisms while preserving differential privacy guarantees. In particular, sequential and parallel composition are relevant properties that we consider in our work, defined as follows [ZLZP17]:

- *Sequential composition*: the composition of a set of differential private mechanisms $(\mathcal{M}_1 \dots \mathcal{M}_n)$, where each mechanism $\mathcal{M}_i$ satisfies ϵ_i-differential privacy, is $\sum_i \epsilon_i$-differentially private.
- *Parallel composition*: the composition of a set of differential private mechanisms $(\mathcal{M}_1 \dots \mathcal{M}_n)$ applied on disjoint subsets of data, where each mechanism $\mathcal{M}_i$ satisfies ϵ_i-differential privacy, is $\max(\epsilon_i)$-differentially private.

Another critical property of differential privacy is that it is immune to post-processing [DR+14]. In other words, privacy guarantees hold even if the output of a differentially private mechanism undergoes arbitrary transformations or auxiliary information is available.

Because of its properties and the fact that DP can mitigate various privacy risks like linkage, inference, and reconstruction attacks [DR+14, YSZ+22], many researchers have been interested in combining differential privacy with other privacy-preserving techniques, such as synthetic data generation and federated learning [ZCP+17, DLH+23, STC+16].

Various differentially private mechanisms have been investigated to strengthen privacy in synthetic data generation. For instance, Zhang et al. [ZCP+17] used the exponential and Laplace mechanisms in their work called PrivBayes. Duan et al. [DLH+23] introduced Gaussian noise to the parameters of the discriminator of the CTGAN model to enhance privacy. Similarly, Fang et al. [FDK22] achieved differential privacy for the CTGAN model by clipping gradients and adding calibrated noise.

Additionally, differential privacy can prevent data leakage from the models shared in federated learning. Differential privacy mechanisms can be applied on the server, client, or both sides. Specifically, three categories can be distinguished when using DP in federated learning: central, local, and distributed differential privacy. The first applies noise on the server side and protects the global model updates, the second adds noise on the client side before sharing the parameters with the server, and the third combines secure aggregation with differential privacy. A full description of these categories is out of the scope of our work. For a detailed overview, we refer the reader to the survey by Zhang et al. in 2023 [ZLL23].

2.4 Secure Multi-Party Computation

Secure Multi-Party Computation (SMPC) [CD+15] is a privacy-preserving technique that enables multiple distributed parties holding a secret input to compute a joint function without disclosing any information beyond the output. It relies on cryptographic protocols that ensure two main properties: privacy of the parties' input and correctness of the function output if the number of dishonest parties [ZZZ+19] does not exceed the threshold of the scheme (in many settings, this means not more than half of the participant can be dishonest). The concept of SMPC has its roots in the famous Yao's Millionaires Problem [Yao82], first introduced in 1982, which involves two millionaires aiming to

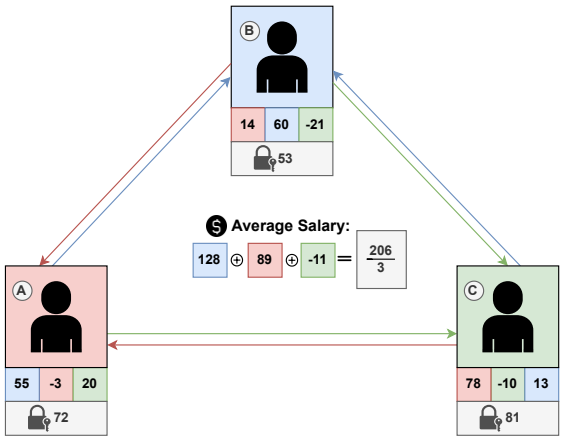


Figure 2.3: Example of average salary computation among three employees (A, B, and C) using additive secret sharing. The grey boxes indicate the private salary of each employee, whereas the colors (red, blue, and green) indicate the respective shares exchanged in the computation. The values used to compute the average salary correspond to the sum of the shares disclosed by each employee.

find out who is richer without disclosing their wealth. Several solutions to this problem have been proposed since its introduction, serving as building blocks to resolve more complex problems. Although the research was mainly theoretical in the early stages, SMPC has evolved in recent years into a more practical tool relevant to many applications (e.g., privacy-preserving machine learning and data mining) [ZZZ+19]. Some limitations remain, such as the communication overhead and computational cost. Nevertheless, there is an ongoing effort in the literature to find more efficient and practical protocols.

Secret Sharing is a well-known cryptographic primitive used in practical implementations of SMPC [KVH+21]. This technique splits a secret into shares and distributes them among multiple parties so no single party can learn the secret. However, when authorized parties combine their shares, they can reconstruct the original secret and perform computations on it. To showcase this, we consider the simple example in Figure 2.3, where three employees, represented as A, B, and C, want to determine their average salary without revealing more information. Each employee partitions their salary into three random shares such that the sum of the shares equals their salary. Then, they exchange one share with each other participant; in our example of three participants, this means that each of the other two participants receives a total of two other shares. Once the process is completed, each participant sums up all shares they possess and discloses the result to other participants to compute the average salary. Notice that the average salary obtained is the same as if we compute the average using their secret inputs. This is a simple example of a secret sharing scheme, but operations are usually performed over finite fields, and different schemes can be used.

Researchers have investigated the possibility of using SMPC in federated learning to

enhance the privacy of the local models during aggregation. Secret sharing-based protocols have gained popularity in this context. The first proposed protocol to address this problem was SecAgg [BIK+17], which uses pairwise random masks to hide the client's updates. These masks are later canceled at the server during aggregation. Several protocols have been developed to improve SecAgg, such as SecAgg(+) [BBG+20], LightSecAgg [SHY+22], and FastSegAgg [KRKR20]. However, they vary in the communication overhead, the ability to handle participant dropouts, and privacy guarantees. Despite ongoing research, there are still open challenges when implementing secure aggregation in FL across different scenarios.

2.5 Homomorphic Encryption

Homomorphic Encryption (HE) is an encryption mechanism that enables computations in ciphertexts while ensuring the same results as operations on plaintexts. There are different types of homomorphic schemes. Usually, these schemes can be classified as partially and fully HE schemes [KFE17]. Partially homomorphic encryption (PHE) only allows for one mathematical function on the ciphertexts (e.g., addition). Meanwhile, fully homomorphic encryption (FHE) enables multiplication and addition operations on ciphertexts [AAUC18]. The past fifteen years have seen significant advances in the field of HE with several schemes proposed and optimized.

2.6 Privacy-preserving data synthesis in distributed settings

Much research has been conducted on data synthesis in centralized settings. However, in some situations, having a representative amount of high-quality data in a centralized setting is not feasible due to legal constraints and privacy concerns. This directly impacts the quality of the synthetic data as it is closely related to real data quality. Therefore, there has been an increasing interest in using federated learning to enable data synthesis for distributed settings. The main goal is to produce more representative synthetic data by leveraging information from multiple clients without sharing the data. Little et al. [LEA23] provided an overview of this field's current state of research. Their findings demonstrate that much of the research concentrates on image data. Out of 21 papers, only six considered tabular data, and in most cases, they used GANs as the generation method but with different configurations on the server and clients. Moreover, from the papers analyzed, nine used differential privacy. The outcomes of this work suggest that federated data synthesis requires more exploration, especially for tabular data.

Duan et al. [DLH+23] proposed a federated generative model for decentralized tabular data synthesis consisting of three main parts: a federated variational Bayesian Gaussian mixture model for learning multi-modal distributions, a federated conditional one-hot encoding for categorical variables, and a privacy-consumption-based federated GAN for generating the synthetic data. The authors used five datasets to evaluate the final

synthesizer and split the data into three clients. Then, they trained a model for different ML tasks and compared the results with the state-of-the-art federated generative model for images and with the original data. The results showed that regarding utility, their approach outperforms the model for images and sometimes can even get better results than with the original data. Furthermore, they used a membership inference attack to verify the privacy level of the proposed model. This work is similar to the work in this thesis. Still, we consider more data partition scenarios, evaluate scenarios with different numbers of clients, extend the metrics for analyzing synthetic data, including various dimensions, and use other generation methods.

Similarly, Zhao et al. [ZBKC21] proposed a federated GAN for tabular data based on the well-known approach CTGAN. Their main contributions are (i) a novel feature encoding scheme that can reconstruct the entire column distribution via bootstrapping each client's partial information and (ii) a weighting scheme to effectively merge local models considering the quantity and distribution dissimilarity for every column across the clients. To evaluate the synthetic data, the authors used four datasets, generated different data distributions, and compared them against different baselines regarding two statistical metrics. In particular, for categorical features, they used the Average Jensen-Shannon divergence (Avg-JSD), while for continuous features, they used the Average Wasserstein distance (Avg-WD). The results demonstrated that both in the IID and non-IID cases the synthetic tabular data preserves the statistical properties of real data. This work has only examined the fidelity of synthetic data but not privacy and utility.

Other studies have also considered GANS for tabular data in the federated setting. Fang et al. [FDK22] proposed a federated version of CTGAN using DP on the client side and evaluated utility for binary classification tasks. However, they did not compare different scenarios or further dimensions like fidelity and privacy. Moreover, Weldon et. al. [WWB21] proposed a federated GAN to generate synthetic electronic health records (EHR), which are evaluated (i) using a statistical comparison between the real and the synthetic data and (ii) subjectively by medical experts through a survey that contained real and synthetic patients that were rated to see how realistic their characteristics are. Results demonstrated no significant difference between real and synthetic patients based on the experts. However, their work suffers from the mode collapse problem.

Moreover, some works have investigated other methods, such as Bayesian networks and VAEs, to generate synthetic tabular data in distributed settings. Su et al. [STC+16] proposed a differentially private sequential update of Bayesian networks, denoted as DP-SUBN, which addresses the problem of high dimensional data publishing in distributed settings. Their approach comprises different phases. First, they propose a search frontier that allows a semi-trusted curator(server) and multiple parties to build a Bayesian Network structure jointly. Then, they use the multi-party Laplace mechanism [PRR10] to learn the network parameters. Finally, they sample tuples from the approximate distribution defined by the learned Bayesian network. The search frontier contains possible candidate attribute-parent (AP) pairs and their marginal distributions. A strategy using

correlations between attributes is proposed to reduce the number of possible candidates. Since different steps in their approach required the exchange of information that might lead to a privacy breach, such as the correlations of attribute pairs, the authors used differential privacy mechanisms such as the Distributed Laplace Perturbation Algorithm (DLPA) [RN10] and exponential mechanism to provide privacy guarantees.

Cheng et al. [CTS+19] proposed an extended version of DP-SUBN, and performed a more exhaustive evaluation. Specifically, they introduced two different methods for the search frontier: an exact method based on backtracking and a heuristic method that adds edges greedily, ensuring the resulting structure is a valid directed acyclic graph. Furthermore, the authors changed the DP mechanism to aggregate results for the distributed Laplace permutation algorithm (DLPA) proposed in [RN10]. They used four tabular datasets and partitioned them randomly and equally among the parties for their evaluation. The results were evaluated with respect to the accuracy of α-way marginals, the misclassification rate of an SVM classifier, and the number of parties against the ϵ parameter used to achieve differential privacy. Also, the communication cost was reported. Both [CTS+19] and [STC+16] address a similar problem to this thesis. However, we propose a new methodology to build the Bayesian Network in the distributed setting and consider different partitions, including non-IID and unbalanced distributions among clients.

Margaritis [Mar21] proposed a federated variational auto-encoder with differential privacy for generating synthetic data in distributed settings in his diploma work. The approach federates the decoder and keeps the encoder private to provide guarantees in terms of privacy. The evaluation is performed on an image and tabular dataset and compared with PrivBayes [ZCP+17] and DPFedGANs [AMR+19], which have been adapted for the same purpose. The results were favorable regarding data utility and provided good privacy guarantees when the number of clients is larger than 1,0000. The author proposed several directions for future work, such as using a non-private encoder, alternative notions of local differential privacy (LDP), extending to other datasets and data partitions, and using further evaluation metrics. In this thesis, we considered all these aspects and studied VAEs in the federated setting specifically for synthetic data generation of tabular data under multiple scenarios.

CHAPTER 3

Federated Bayesian Networks

In this chapter, we provide the necessary background knowledge to understand how Bayesian network learning works, explain how it is used in a centralized setting to generate synthetic data, discuss the challenges of adapting this method to the federated setting for the same purpose in a privacy-preserving manner and present the approaches investigated in this work to solve these challenges.

3.1 Bayesian Networks Learning

A Bayesian network (BN) [KF09] is a probabilistic graphical model used to encode the joint probability distribution P over some set of random variables $\mathcal{X} = \{X_1, \ldots, X_d\}$. Formally, it can be defined as a pair $\mathcal{B} = (G, \theta)$, where G is the network representation and θ are the parameters associated with the network. More precisely, $G = (V, E)$ is a directed acyclic graph (DAG) characterized by a set of nodes V and a set of edges E. Each node $X_j \in V$ represents a random variable. An edge $(X_j, X_i) \in E$ indicates a direct dependence between X_j and X_i, where X_j is defined as a *parent* of X_i and X_i as a *child* of X_j. The set of all *parents* of X_i is denoted as Π_i. On the other hand, the degree of the network, denoted as k, is given by the size of the largest parent set Π_i. Figure 3.1 shows a simple example of a Bayesian Network with four variables $\{A, B, C, D\}$ with *degree* 1.

The notion of d-separation in Bayesian networks relates the separation of nodes in the DAG with the concept of independence [KF09]. This is crucial for obtaining a compact representation of the joint probability distribution P, which can be expressed as follows:

$$P(X_1, \ldots, X_d) = \prod_{i=1}^{d} P(X_i \mid \Pi_i) \tag{3.1}$$

where $P(X_i \mid \Pi_i)$ is the conditional probability distribution (CPD) of a variable X_i given its parents Π_i, the parameters θ correspond to the CPDs of all attributes.

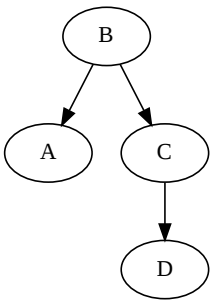


Figure 3.1: Simple Example of a Bayesian Network with four variables

The learning process comprises two main steps: *structure learning* and *parameter learning*. The *structure learning* step involves finding the DAG structure that best encodes the causal relationships between variables in the data. The *parameter learning* step estimates the CPDs of the data based on the DAG structure.

The *structure learning* step can be performed manually, through automatic learning from data, or by using a hybrid approach. The manual process when dealing with large amounts of variables can be time-consuming and may not be accurate [FZM+23]. On the other hand, learning the structure of the data is a very challenging task. Namely, it is an *NP-hard* problem, as shown in [CGH94, CHM04], because the number of possible DAGs grows exponentially with the number of variables. Therefore, researchers have investigated approaches based on heuristics and approximate algorithms to tackle this problem. Kitson et al. [KCG+23] provided an overview of these approaches and classified them mainly into three categories: constraint-based, score-based, and hybrid. The constraint-based approaches employ conditional independence (CI) tests to construct DAGs that accurately represent the independence of variables in the data. The score-based approaches employ optimization strategies to find a DAG that maximizes an objective scoring function. The hybrid approach commonly reduces the space of possible DAG structures using the constraint-based approach, and then it selects the best structure using the score-based approach.

Popular algorithms used as search strategies include hill-climbing, genetic algorithms, and greedy algorithms [KCG+23]. Meanwhile, scoring functions can be divided into Bayesian scoring functions and Information-theoretic scoring functions. The Bayesian scoring functions incorporate prior knowledge to find the structure. Examples include the K2 [CH92] and the Bayesian Dirichlet equivalent uniform (BDeu) [HGC95] scores. Information-theoretic scoring functions seek to balance the fit of the network to the data against the model complexity. Examples include the Bayesian Information Criteria (BIC) [Sch78] and the Mutual Information Test (MIT) [DCF06]. In the case of synthetic data generation, there is no prior information about the structure or dependencies. Hence,

information-theoretic scores are preferred [KCG+23].

The *parameter learning* step depends on whether the data is complete or incomplete. The data is complete if all variables are fully observed and there are no missing data. Otherwise, it is incomplete. Maximum likelihood estimation (MLE) is a well-known parameter learning method for complete data. The method maximizes a likelihood function by finding the parameters that better fit the observed data. On the other hand, expectation maximization (EM) is an algorithm that can be used when data is incomplete, which relies on inference methods to deal with hidden variables [JXM15].

This work concentrates on score-based approaches for the structure learning step, characterized by a search strategy and a scoring function, and considers complete data for the parameter learning step. However, if missing data is present, we discuss how to handle this case.

Several works use BNs to generate synthetic data in the centralized setting [YGP09, ZCP+17, KSP+21]. We adapt this to the federated setting in this thesis.

3.2 Bayesian Networks: Centralized Setting

In this section, we present specific approaches for generating synthetic data using Bayesian networks in a centralized setting and describe the main building blocks considered in our work as a reference for the federated setting.

Zhang et al. [ZCP+17] undertook relevant work in this line. In particular, they proposed PrivBayes, an approach for publishing private high-dimensional data in a centralized setting using low-degree Bayesian Networks and differential privacy. Their approach comprises three main steps. The first step is structure learning, which uses a greedy algorithm to construct a k-degree Bayesian Network. This step is modeled as an optimization problem that aims to reduce the Kullback–Leibler (KL) divergence between the original data distribution and the approximate version that comes from the network. To achieve this, the greedy algorithm selects a parent set for each attribute in such a way that the mutual information is maximized and ensures that the process satisfies differential privacy via the exponential mechanism. The second step is parameter learning, which injects Laplacian noise into the conditional distributions for each attribute in the Bayesian Network. The final step involves generating synthetic data by approximating the original data distribution using the constructed structure and noisy conditional distributions and then sampling from it.

Several works have extended PrivBayes. For instance, Ping et al. [PSH17] developed a Python-based implementation named DataSynthesizer, which generates synthetic data, leveraging the algorithms and privacy mechanisms of PrivBayes. This tool is composed of three main components: the DataDescriber, the DataGenerator, and the ModelInspector. The DataDescriber learns the characteristics of the input data, including the data types, domains, and missing rates. Then, depending on the mode of operation selected by the user (random, independent, or correlated), different mechanisms are used to describe

the input data. In particular, the correlated attribute mode constructs a k-degree BN with DP guarantees following the approach of Zhang et al. [ZCP+17]. The user can control the amount of noise injected in this mode by setting a value to the parameter ϵ. If $\epsilon = 0$, no noise is injected. The DataGenerator uses the description provided by the first component to sample new (synthetic) data. Finally, the ModelInspector generates a statistical report on the similarity between the input and synthetic data generated.

Furthermore, Hittmeir et al. [HME22] proposed a variation of the structure learning step in PrivBayes based on a genetic algorithm (GA) and introduced a novel approach for mitigating disclosure risks in BNs by decreasing specific correlations for sensitive attributes. Overall, their results demonstrated a significant improvement in efficiency with respect to the original greedy algorithm proposed by Zhang et al. [ZCP+17] and similar performance in terms of fidelity and utility on three tabular datasets. Furthermore, their experiments showed that the approach for mitigating disclosure risks also performs well against attribute disclosure attacks without compromising the utility of the synthetic data.

Considering the advantages in terms of efficiency of the approach by Hittmeir et al. compared to the greedy algorithm by Hittmeir et al. [ZCP+17], we investigate in detail this approach for the structure learning step. Meanwhile, we utilise the parameter learning and data generation step of PrivBayes [ZCP+17]. Furthermore, we use the DataSynthesizer tool for the implementation phase with the functionalities provided by Hittmeier et al. [HME22] and extend it to support our approach.

3.2.1 Structure Learning

The genetic algorithm proposed by Hittmeir et al. [HME22] represents possible BNs as individuals. Each individual is characterized by two independent chromosomes: the *ordering chromosome* and the *connectivity chromosome*. The *ordering chromosome* is a list representing the ordering in which the nodes (attributes) are added to the DAG structure. The *connectivity chromosome* is a list of sets where each set has exactly k attributes representing the possible parents of a node. Recall that k is the degree of the BN. It is important to bear in mind that the first k attributes in the ordering have fewer parents. This occurs because the number of preceding attributes in their case is less than k. Therefore, when transforming an individual into a valid DAG, attributes in the parent set of a given node appearing later in the ordering are ignored.

Figure 3.2 provides an example of an individual obtained when running the genetic algorithm in a simplified version of the Adult dataset described in Section 5.1. The upper box in the figure shows the individual's chromosomes, whereas the bottom box depicts the valid DAG structure after transforming the individual. Note that the ordering chromosome is a permutation of the attributes in the dataset, which, in the example, are encoded as numbers. Meanwhile, the i-th set in the connectivity chromosome always corresponds to the possible parents of attribute i. For instance, attribute 3 (Relationship) is the second attribute in the ordering chromosome, and its possible parents based on the

connectivity chromosome (3-rd set) are 5 and 2. However, attribute 2 (Marital-status) does not precede attribute 3 in the ordering chromosome. Therefore, it is ignored, and the only valid parent for this node in the DAG structure is attribute 5 (Income), as shown in the network. The same principle applies to all other attributes.

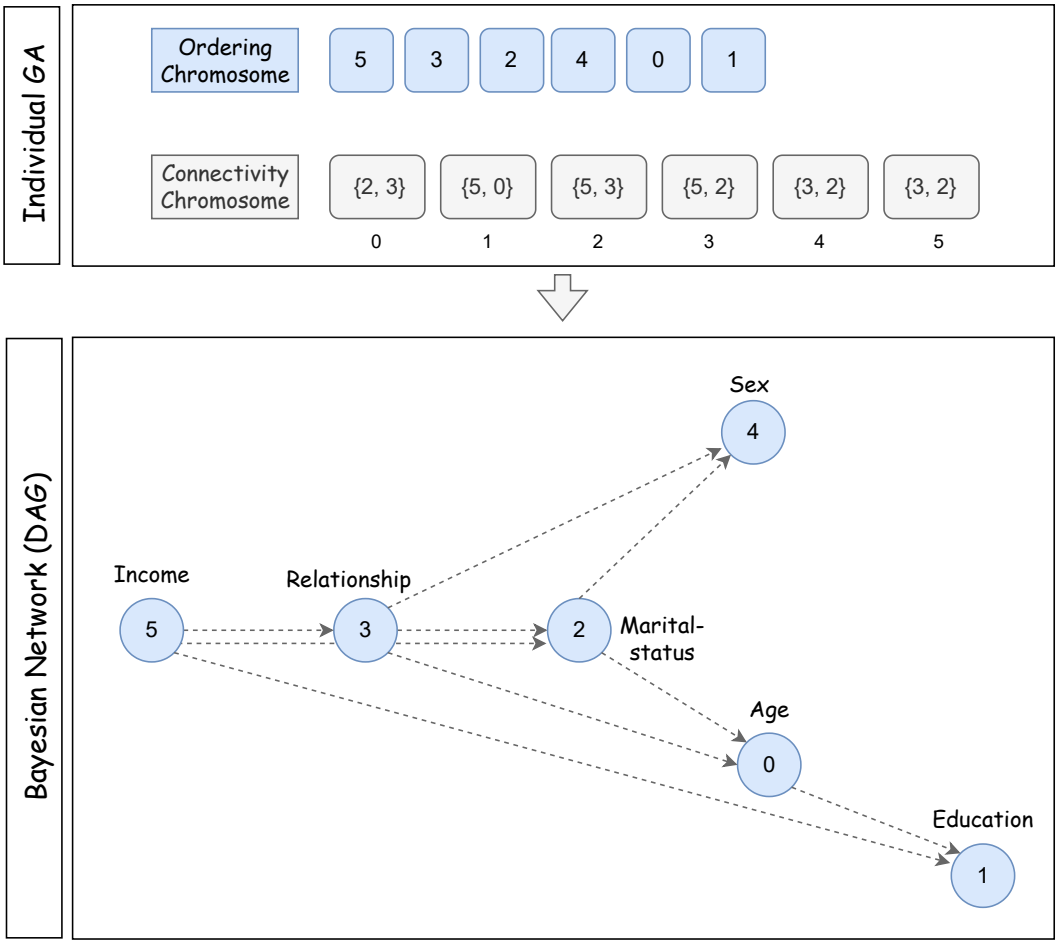


Figure 3.2: Example illustrating an individual in the genetic algorithm of a 2-degree Bayesian Network with its respective ordering and connectivity chromosomes for a simplified version of the Adult dataset.

Algorithm 3.1 presents the pseudo-code of the genetic algorithm proposed by Hittmeir et al. [HME22]. The process starts with the initialization step, where a population of N individuals is randomly generated. Then, these individuals are evaluated with respect to a fitness function, and then the S fittest individuals are selected. In this case, the fitness function is the sum of the *pairwise mutual information* of each attribute-parent pair. These values are pre-computed and stored at the beginning of the algorithm, so access to the input dataset is required only once. The S fittest individuals undergo several operations: crossover and mutations until a new generation of N individuals

is obtained. The **Crossover** operation selects two individuals and combines their connectivity chromosomes to generate a new individual. Meanwhile, for mutation, two operators are employed: (i) the **Order Flip** operation alters the ordering of some attributes in the ordering chromosome, and (ii) the **Swap** operation performs random swaps on the connectivity chromosome and repairs invalid structures obtained in the last operation. For more details on the operations, we refer the reader to the original work of Hittmeir et al. [HME22]. The same process from the evaluation step is repeated e times for individuals in the new generations. In the last generation, the fittest individual is selected and transformed into a valid DAG. The resulting DAG G is the solution of the structure learning step. Algorithm 3.2 describes how the parameters from the Bayesian network are derived in a differentially private manner in PrivBayes. For a detailed explanation, refer to [ZCP+17].

Algorithm 3.1: Genetic Algorithm (Adapted from [HME22])

$t \leftarrow 0$;
Initialize first generation P_0 with N random individuals;
Evaluate P_0 based on fitness function;
for $t \leftarrow 1$ **to** e **do**
    Let $\mathcal{E}$ be the set of the S fittest individuals from the previous generation P_{t-1};
    $\mathcal{G} \leftarrow \mathcal{E}$;
    while $|\mathcal{G}| < N$ **do**
        Choose a random individual i in $\mathcal{E}$;
        Generate a random number x in $[0, 1]$;
        if $x < r$ **then**
            Choose a random individual i' in $\mathcal{E}$;
            $i \leftarrow$ **Crossover**(i, i') ;
        end
        $i \leftarrow$ **Order Flip**(i);
        $i \leftarrow$ **Swap**(i);
        Add i to $\mathcal{G}$;
    end
    $P_t = \mathcal{G}$;
    Evaluate P_t based on fitness function;
end
Select the best individual in the last generation $P_{t=e}$;
Let G be the valid DAG obtained from the best individual;
return G;

3.2.2 Parameter Learning

The parameter learning step proposed by Zhang et al. [ZCP+17] assumes that the network structure of a Bayesian network $\mathcal{B}$ is fixed and that the input dataset D is

complete. It then uses a *frequentist approach* to estimate the CPDs of all attributes.

The *frequentist approach* uses maximum likelihood estimation (MLE) to estimate the parameters θ. The MLE of a Bayesian network can be computed by maximizing each local likelihood function independently and combining the solutions [KF09]. The local likelihood determines how well a variable X_i can be approximated given its parents Π_i. This is maximized using the local relative frequencies from the data. Therefore, the parameters to compute are reduced to:

$$\hat{\theta}_{ijk} = \frac{m_{ijk}}{\sum_j m_{ijk}} \tag{3.2}$$

where m_{ijk} corresponds to the number of frequency of occurrences where the variable $X_i = j$ given that the parent variables Π_i takes values corresponding to k.

Algorithm 3.2: NoisyConditionals [ZCP+17])

```
Initialize P* = ∅
for i ← k + 1 to d do
    Materialize the joint probability distribution P(X_i, Π_i)
    Generate differentially private P*(X_i, Π_i) by adding Laplace noise
      Lap(2·(d−k) / n·ε)
    Set negative values in P*(X_i, Π_i) to 0 and normalize; Derive P(X_i | Π_i) from
      P*(X_i, Π_i)
    add it to P*
end
for i ← 1 to k do
    Derive P(X_i | Π_i) from P*(X_{k+1}, Π_{k+1}); add it to P*
end
return P*
```

3.2.3 Data Generation

The approach proposed in PrivBayes for the data generation step uses the structure and parameters of the Bayesian network $\mathcal{B}$ to produce a synthetic dataset D^* with an arbitrary number of tuples. In particular, the structure G is used to sample attribute values efficiently. The process starts by sampling from an unconditional probability distribution of the root attribute X_1 and then follows by sampling from the conditional probabilities $P(X_j \mid \Pi_j)$ the remaining attributes $X_j (j \in [2, d])$ in the order they were inserted to the network. As shown in Equation (3.1), the conditional probabilities for each attribute X_i only depend on $\{X_i\} \cup \Pi_i$. Since the parents Π_i of an attribute X_i always appear before in the ordering, by the time when we sample X_i, the parents have already been sampled, and we are able to sample from the conditional probabilities $P(X_i \mid \Pi_i)$. Hence, we don't require the full approximate joint distribution to perform

the sampling process [ZCP+17], only the network and the parameters to sample attribute values in the appropriate order [PSH17].

3.3 Bayesian Networks: Federated Setting

The findings in the last section further motivate the idea of considering BNs as a potential method for generating high-quality synthetic tabular data in distributed settings. This section analyzes the challenges of adapting this method to the federated setting.

The problem we are considering is the following: A set of clients $C_1 \dots C_n$ aim to learn a Bayesian network $\mathcal{B}$ collaboratively for privacy-preserving data synthesis. Each client C_i owns a tabular dataset D_i containing sensitive information that cannot be disclosed to other clients due to legal constraints and privacy concerns. The private datasets $D_1 \dots D_n$ have the same feature space, but their sample space differs (i.e., the dataset is horizontally partitioned). The feature space can contain mixed data types (discrete and continuous variables). The model is constructed under the coordination of an honest-but-curious server. Therefore, each step for learning $\mathcal{B}$ must be carried out in a privacy-preserving manner to prevent the server from inferring client information. Furthermore, we assume the synthetic dataset generated using $\mathcal{B}$ will be shared with third parties. This means that the synthetic dataset should not compromise the privacy of the individuals in the private datasets but is expected to have high fidelity and utility to serve for further analysis or research purposes.

Before addressing the problem of adapting BNs for federated learning, we investigate related work in this direction. Table 3.1 provides an overview of approaches that address related problems. In particular, we classify these works based on the distributed setting (horizontal or vertical), the learning phase (structure learning, parameter learning), the algorithm used in the structure learning phase (if it applies), and the privacy-preserving techniques applied. We also include our work in the table to indicate the main differences with other works.

Ma and Sivakumar [MS06] proposed an approach based on post-randomization for structure and parameter learning. Meng et al. [MSK04] proposed a random projection-based method for securely learning the parameters of a given Bayesian network in a vertically distributed setting and showed that for binary value datasets, conditional probabilities can be expressed as a set of linear equations with inner products. Zhang and Wright [YW06] proposed an approach using SMPC and the K2 algorithm for structure and parameter learning from vertically partitioned data but extended it to non-binary data. Furthermore, Samet and Miri [SM09] proposed a privacy-preserving construction of Bayesian networks structure for horizontally partitioned data using the K2 algorithm and three secure protocols: Secure Exponentiation, Secure Multi-party Factorial, and Secure Product Comparison. It is worth noting that the purpose of most works is not synthetic data generation.

We now describe our approach to adapting BNs to the federated setting. The design

comprises four main parts: preprocessing, structure learning, parameter learning, and data generation. Here, we discuss the implications of each phase, the ways we can handle this privately, and the design we propose. For simplicity, in upcoming sections, we will refer to the proposed approach as FedBN.

Approaches		[WY04]	[MSK04]	[MS06]	[SM09]	[STC+16]	[Dig18]	[NZ22]	[VDIDB22]	Our work
Distributed Setting	Horizontal				✓	✓	✓	✓		✓
	Vertical	✓	✓	✓					✓	
Learning Phase	Structure	✓		✓	✓	✓	✓	✓	✓	✓
	Parameter		✓	✓		✓	✓		✓	✓
Algorithm (Structure)	K2	✓		✓	✓				✓	
	Greedy					✓	✓			
	GA									✓
	Other							✓		
Privacy Techniques	Post Randomization			✓						
	Random Projection		✓							
	DP					✓	✓			✓
	SMPC	✓			✓	✓			✓	
	HE					✓				

Table 3.1: Approaches for learning a BN in distributed settings

3.3.1 Preprocessing

Continuous variables are generally not suited for Bayesian Networks. Therefore, one step usually performed before learning BNs is discretizing continuous variables. This is not an issue in the centralized setting where the variable's range is known. However, in the distributed setting, the ranges might differ among clients, and sharing statistics as minimum and maximum values can sometimes raise privacy concerns, mainly if outliers exist in the data. As this is required for the federated setting, an option to prevent data leakage is to use SMPC protocols [ZCZ15] to perform the minimum and maximum computation privately. However, it is worth noting that while this prevents the server from learning each client's specific min max values, the server will still know the global min and max values.

On the other hand, not all the values of categorical attributes might be present in all the clients in the federated setting [DLH+23]. Therefore, information about the existing categories should also be shared with the server to align the features before learning the BN model. While this might not necessarily be a privacy issue, as some categories can be publicly known (e.g., gender, civil status), the server could potentially exploit this information if the categories are rare or too specific and belong to a sensitive attribute. In such cases, one alternative is to leverage an SMPC protocol, such as Private Set Union (PSU) [KS05], which ensures that the server only learns the union of each categorical variable without revealing which categories belong to specific clients. The disadvantage of such protocols is that they are computationally expensive. Another alternative for a production-ready setting is for clients to agree on a standard data model.

In our implementation, we assume we have access to an SMPC protocol to compute clients' minimum and maximum values. Meanwhile, for categorical attributes that are not publicly known, we assume the existence of one of the alternatives discussed beforehand.

3.3.2 Structure Learning

In the structure learning step, we need to decide what to share with the server to ensure that the Bayesian Network structure accurately approximates the distribution of the private datasets but does not leak private information. Intuitively, the structure encodes information about the correlations between the variables in the dataset [ZCP+17]. Therefore, we must ensure that this does not disclose information about the sensitive attributes in private datasets.

Considering the genetic algorithm described in section 3.2.1, we propose a distributed approach for learning BN structure inspired by the work proposed in [DFDCK+23]. In particular, [DFDCK+23] studied a Grammatical Evolution algorithm for glucose prediction in the federated setting. Their approach leverages the idea of migrating individuals over populations in each client over a certain frequency with a server in such a way that the clients can combine their individuals with those that other clients have best ranked, and the diversity can, in the end, provide a population that combines information from all the clients.

We leverage the same migration idea in [DFDCK+23] but apply it in the context of Bayesian network learning. Algorithm 3.3 describes our approach. Each client generates a random population of N individuals in the first round and runs the genetic algorithm for some local epochs. Then, individuals in the last generation are stored, and the best individual is sent to the server. The server collects the best individuals from all the clients. This set is denoted as I_t. The server proceeds to send I_t to the clients. Each client determines the set $\mathcal{E}$ of the fittest individuals from their previous generation and then adds the individuals in the set I_t from other clients to their set $\mathcal{E}$ in the first local iteration. After that, the client generates new generations for the corresponding local epochs and repeats the same process. The process stops after the server completes T rounds. At this point, the server again sends the set I_t to the clients. But this time, the clients evaluate all the individuals received in their data and send the scores to the server. The server calculates the average scores and selects the individual with the highest average score as the global solution for the structure learning step.

3.3.3 Parameter Learning

After the structure learning step, the global network structure G is known by the clients participating in the federated training. Therefore, the remaining task is to estimate the parameters θ of the Bayesian network.

In the centralized setting, we can compute the parameters immediately from the dataset. Unfortunately, the records are spread across multiple clients in the federated setting. This implies that each client needs to share *sufficient statistics* to compute the global CPDs (i.e., the counts for all possible combinations of nodes and their parents). This can leak information about their private datasets to the server. Note that the fewer the elements in the combinations counts, the higher the risk of re-identification. Therefore, we must use privacy-preserving techniques to protect the computation during the aggregation.

Algorithm 3.3: FedGA

Server runs:
Initialize the individuals set $I_0 = \emptyset$
foreach round $t = 0 \dots T$ **do**
  foreach client $k \in M$ **in parallel do**
    x_{t+1}^k= ClientUpdate(k,I_t)
  end
  $I_t = \bigcup_k \{x_t^k\}$
end
foreach client $k \in M$ **in parallel do**
  f_t^k= ClientEvaluate(k,I_t)
end
$f_t = \sum_{k=0}^{M} f_t^k$
Select the individual $i \in I_t$ with the highest fitness score on average in f_t
Let G be the valid DAG obtained from the best individual
Send G to each of the clients
Function `ClientUpdate` (k,I_t)**:**
  f = Evaluate generation P_{t-1} based on fitness function
  foreach round $j = 1 \dots$ local_epochs **do**
    Let $\mathcal{E}$ be the set of fittest individuals in f
    if $j == 1$ **then**
      $\mathcal{E} = \mathcal{E} \cup I_t$
    end
    P_j = nextGeneration($\mathcal{E}$)
    f = Evaluate P_j based on fitness function
  end
  $P_t = P_j$
  Store current generation P_t
  Let x_t^k be the best individual in the current generation
  return x_t^k to server
Function `ClientEvaluate` (k,I_t)**:**
  f_t^k = Evaluate individuals I_t based on fitness function
  return f_t^k to server

Note that, in general, when publishing synthetic data, the parameters can also leak information about the data; thus, considering the protection of the output, even if not specific to the federated setting, can influence the choice of protection of the computation during aggregation.

Zhang et al. [ZCP+17] proposed to use differential privacy when computing the parameters of the BN via the Laplace mechanism in the centralized setting to protect the output, as explained in section 3.2.2.

One approach that could be leveraged in the distributed setting is to let each client generate the noisy counts locally via the Laplace mechanism, ensuring differential privacy and sending the noisy counts to the server. The parallel composition theory implies that the aggregated result also satisfies differential privacy, so the leakage in the aggregation step is reduced. However, as noted in previous works [CTS+19, GX15], the total amount of noise in the aggregated result is too large, making the model ineffective. Therefore, one way to mitigate this problem is to use distributed privacy mechanisms.

A distributed privacy mechanism that has been used in this setting is the Distributed Laplace Perturbation Algorithm (DLPA) [CTS+19]. This mechanism exploits the Laplace distribution's property of *infinite divisibility*, which entails that the same distribution can be approximated by summing up n random variables. Different distributions can be used to draw the partial noise, including Gamma, Gauss, and Laplace [GX15]. The main difference between them is the number of random variables that need to be generated by the clients and the operations that should be performed under SMPC.

The Laplace distribution $\mathcal{L} \sim \mathcal{L}(0, s)$ can be simulated using four random variables drawn from the normal distribution as follows:

$$\mathcal{L}(0, s) = \mathcal{N}_1{}^2 + \mathcal{N}_2{}^2 - \mathcal{N}_3{}^2 - \mathcal{N}_4{}^2 \tag{3.3}$$

where $N_i \sim Gauss(0, \frac{s}{2})(i \in \{1, 2, 3, 4\})$.

In our implementation, we ensure differential privacy during the parameter learning step by injecting partial noise into each client's counts. This noise guarantees that the global parameters satisfy differential privacy when the server aggregates the counts. We assume access to the Distributed Laplace Perturbation Algorithm (DLPA) for this aggregation process.

CHAPTER 4

Federated Variational AutoEncoders

In this chapter, we provide the necessary background knowledge to understand how Variational Autoencoders work, explain how they are used in a centralized setting to generate synthetic tabular data, and then discuss how we adapted this method to the federated setting for the same purpose in a privacy-preserving manner.

4.1 Variational Autoencoders

Variational Autoencoders (VAEs) [KW13] are generative models composed of two neural networks, an encoder and a decoder, which are connected through an intermediary layer known as the latent space. Since their introduction in [KW13], they have been used for multiple applications, including synthetic data generation [XSCIV19, MTT+20], anomaly detection [AC15, KKH18], and representation learning [RV20]. From an architectural point of view, VAEs are similar to traditional autoencoders (AEs). However, the mathematical foundations of these models differ significantly [Doe16]. VAEs incorporate concepts of probability theory and statistics to achieve robust generative capabilities. We thus first outline the differences between traditional AEs and VAEs, focusing on their application for synthetic data generation.

In traditional AEs (see Figure 4.1), the encoder transforms the input data into an intermediate representation known as the latent space, usually with a lower dimension. This representation takes the form of fixed-size vectors z and is used by the decoder as input to reconstruct the original input. The main goal of AEs is to minimize the *reconstruction error* between the input data x and the reconstructed data $\hat{x}$. To achieve this, the encoder and decoder are trained simultaneously using backpropagation to find the parameters that reduce this error. Usually, the error is calculated using the Mean Squared Error (MSE) or the Binary Cross-Entropy (BCE) [Mic22].

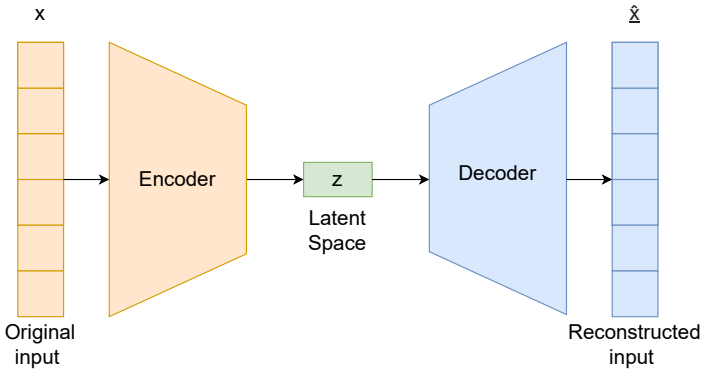


Figure 4.1: Traditional Autoencoders Architecture

Once trained, AEs can be leveraged to generate new data by sampling data points from the latent space and then passing them as input to the decoder [EW22]. However, since the latent space is not regularized, data points lying far from the training data might arise in the sampling process, leading to representations that the decoder cannot handle and thus, generate inconsistent outputs.

VAEs address this limitation in traditional AEs representing the latent space as a probability distribution, typically Gaussian [Doe16], characterized by a mean and variance vector (see Figure 4.2). More precisely, the encoder is formulated as an inference model, $q_\phi(z|x)$, and the decoder as a generative mode, $p_\theta(x|z)$, using neural networks in each case [KW+19]. The inference model approximates the intractable posterior distribution $p_\theta(z|x)$ given the input data x using a method known as *variational inference* [GG14]. This method turns the approximation task into an optimization problem, which assumes the existence of a tractable distribution and tries to find the parameters ϕ that closely approximate the intractable one by minimizing the Kullback-Leibler (KL) divergence. This is still not trivial since the posterior distribution $p_\theta(z|x)$ is intractable.

However, minimizing the KL Divergence is equivalent to maximizing the Evidence Lower Bound (ELBO), denoted as $L_{\theta,\phi}(x)$ (see Equation (4.2)). This equivalence is crucial for VAEs because it achieves two key objectives [KW+19]. On the one hand, maximizing the ELBO also maximizes the marginal likelihood $p_\theta(x)$, improving the quality of the reconstructed data. On the other hand, it minimizes the KL divergence between the estimated posterior distribution and the true posterior distribution. This implies that the ELBO enables the optimization of both the decoder and encoder parameters, making it a suitable loss function for training VAEs.

$$L_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x,z) - \log q_\phi(z|x)] \tag{4.1}$$
$$= \log p_\theta(x) - D_{\mathrm{KL}}(q_\phi(z|x)\|p_\theta(z|x)) \tag{4.2}$$

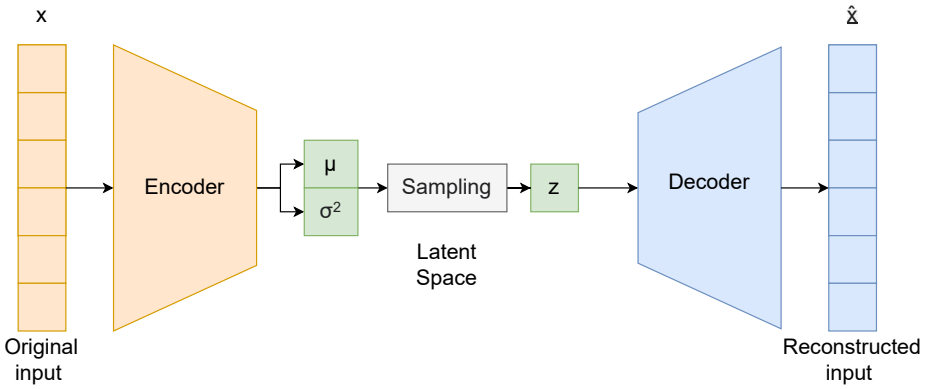


Figure 4.2: Variational Autoencoders Architecture, adapted from [WPL+23])

As mentioned, for traditional AEs, the encoder and decoder are trained together to minimize the loss function. Nonetheless, the sampling process for VAEs is not differentiable, so backpropagation is not possible [Doe16]. To address this, the *reparametrization trick* [KW13] is employed, which expresses the sample variable $z \sim q_\phi(z|x)$ in a differentiable form as follows:

$$\mathbf{z} = \mu + \sigma\epsilon \tag{4.3}$$

where ϵ is a random variable sampled from a standard normal distribution ($\epsilon \sim \mathcal{N}(0, 1)$).

Once the VAE model is trained, we can generate new data by sampling latent variables $z \sim \mathcal{N}(0, I)$ and then passing them as input to the decoder [Doe16].

4.2 Variational Autoencoders: Centralized Setting

This section describes TVAE, a well-known method for generating synthetic tabular data in the centralized setting introduced by Xu et al. [XSCIV19].

TVAE is a generative model that adapts the loss function of VAEs to handle mixed attributes (continuous and discrete) commonly found in tabular data. The model's design incorporates neural networks producing a joint probability distribution with $2N_c + N_d$ variables, where N_c is the number of continuous variables and N_d is the number of discrete variables.

4.3 Variational Autoencoders: Federated Setting

This section explains how TVAE can be adapted to the federated setting and outlines the challenges associated with privacy.

Similar to the problem described in section 3.3, the objective here is to collaboratively learn a TVAE model among n clients with horizontally partitioned tabular data, enabling privacy-preserving data synthesis under the coordination of an honest-but-curious server.

The design of federated VAEs comprises three parts: preprocessing, training, and post-processing, which are described in the following sections. For simplicity, in this work, we will refer to the proposed approach as FedVAE.

4.3.1 Preprocessing

Continuous variables usually need to be transformed before being used as input to neural networks. In the centralized setting of TVAE, this is achieved through an approach called *mode-specific* normalization, which normalizes each variable independently using a variational Gaussian mixture model (VGM). This approach is essential for the performance of TVAE because it allows learning multimodal distributions in continuous variables, helping to prevent mode collapse.

However, as pointed out in previous work [DLH+23], training a VGM model in federated learning is not trivial since the global distribution of continuous variables is not available, and sharing statistics can leak information about the clients. Duan et al. [DLH+23] proposed a novel method called the Federated Variational Bayesian Gaussian Mixture Model (Fed-VB-GMM) to solve this issue. This method adapts the Expectation Maximization (EM) algorithm, which is inherently not privacy-preserving, to the federated setting and employs homomorphic encryption to secure the information shared with the server. Their results demonstrated that Fed-VB-GMM performs comparable to VGM in the centralized setting. However, we cannot reuse their implementation in our work due to incompatibilities in programming environments, and re-implementing their approach from scratch is beyond the scope of this thesis. Despite this, their method is independent of the specific implementation details, and our work focuses on effectiveness rather than the efficiency of the implementation. Therefore, we assume that the approach proposed by [DLH+23] could be effectively adapted for a production-ready setting, but in the experiments conducted in this thesis, we perform the normalization before simulating the federated setting.

On the other hand, categorical variables also need to be adapted to train the TVAE model. This is usually done using one-hot encoding. In the federated setting, clients must share their categories with the server to achieve this. As discussed for the BN model in section 3.3.1, different alternatives can be leveraged if categories leak information, such as using an SMPC protocol or deciding on a standard data model before the federated setting. We assume that one of the alternatives is possible in this work.

4.3.2 Training

The aggregation step in VAEs is less complex than the one in BNs because VAEs are composed of neural networks, and aggregating parameters (weights and biases) of these types of models across different clients is more straightforward. However, one decision that has to be made is whether to share the parameters of the encoder, the decoder, or both. Previous work has explored a federated scheme in which the encoder is kept private, and only the decoder is synchronized [Mar21]. In our approach, we federate

both the encoder and the decoder and aggregate their parameters using the well-known *FedAvg* strategy, which samples clients in each federated round and then computes the average of the parameters on the server side. The pseudocode of this strategy is given in Algorithm 4.1. In our implementation, clients participate in all federated rounds.

Algorithm 4.1: FedAvg (Federated Averaging) [MMRyA16]

Input: Clients' data $\{D_k\}_{k=1}^{K}$, global rounds T, client fraction C, local epochs E, batch size B, learning rate η

Server executes:
Initialize w_0
foreach *round* $t = 1, 2, \dots T$ **do**
  $S_t \leftarrow$ (random set of $(\max(C \cdot K, 1))$ clients)
  foreach client $k \in S_t$ **in parallel do**
    $w_k^{t+1} \leftarrow$ **ClientUpdate**(k, w_t)
  end
  $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_k^{t+1}$
end
ClientUpdate(k, w_t):
; **for** *each local epoch i from 1 to E* **do**
  batches $\leftarrow$ split D_k into batches of size B
  for *batch b in batches* **do**
    $w \leftarrow w - \eta \nabla \ell(w; b)$
  end
end
return *w to server*

Sharing parameters can still leak information in federated learning [PM20]. Therefore, we also explored using additional privacy-preserving techniques to enhance privacy in the proposed approach. In particular, since we assume the server coordinating the federated training is honest but curious, we consider local DP, which provides strong guarantees for clients, as the noise addition is performed locally before sharing the parameters with the server. In particular, we consider the approach proposed by Naseri et al. [NHDC22]. which uses differentially private stochastic gradient descent (DP-SGD) to train the models locally. DP-SGD [ACG+16] is a popular approach used to train ML models with DP guarantees. The approach involves several steps. First, it samples a random batch from the training data. Then, it computes the gradients for each data point and clips them to a maximum norm. After this, Gaussian noise is added to the sum of the gradients. In our implementation, we used the Python library Opacus, which implements the DP-SGD algorithm.

4.3.3 Postprocessing

Since we transformed the data before training the VAE model, we also need to reverse this transformation in order to obtain the same structure as the input data. Here, is crucial to understand that each transformation has to be reversible to obtain consistent synthetic data.

CHAPTER 5

Experiment Design

This chapter describes the methodology followed to perform the experiments for the proposed synthetic data generation approaches. In particular, we describe the datasets used for the experiments and analyze their complexity. We also present the baselines considered to compare the quality of the synthetic data in the federated setting and the partitions used to distribute the data among clients. Furthermore, we explain how we ensure the consistency of the results in the generation process, select the hyperparameters used in the models, and describe the methodology for synthetic data evaluation.

5.1 Datasets

Synthetic data generation is beneficial in fields dealing with sensitive information, as it protects the privacy of individuals in datasets. Therefore, we considered the following criteria when selecting datasets for our experiments. First, we choose datasets containing individual-level information from various sensitive domains, such as healthcare and finance. Second, we prioritize datasets that are commonly used in the literature to benchmark synthetic data generation methods and are publicly available in machine learning repositories such as UCI and Kaggle. Third, we selected datasets with mixed numerical and categorical features, which are particularly challenging for synthetic tabular data generation.

To fulfill these criteria, we studied related papers published within the last 6 years and chose three datasets with the mentioned characteristics. Table 5.1 summarizes the datasets used in this work, including the number of samples and attributes, the target variable, the amount of numerical and categorical features, and the specific machine learning task for which they were originally collected. It is worth noting that we include datasets from the fields of demographics, finance, and healthcare. Also, the size and complexity of the datasets and the imbalance of the target variables differ.

Table 5.1: Summary statistics of the datasets used in this work

Dataset	Number of Instances	# Categorical Attributes	# Numerical Attributes	Target Attribute	Distribution Target Attribute	Task Type
Adult	32,561	8	6	income	75:25	Binary Class.
Cardio	70,000	4	6	cardio	50:50	Binary Class.
Bank	45,211	7	10	subscribed	88:12	Binary Class.

5.1.1 Adult Dataset

The Adult dataset [BK96] is derived from the U.S. Census Bureau data collected in 1994. It contains demographic information about individuals, including their age, occupation, education, race, and relationship. The target attribute is *income*, which is a binary variable indicating whether an individual earns more than 50K per year. Figure 5.1 shows the distribution of each attribute in this dataset. Note that numerical features such as *capital-gain* and *capital-loss* are highly left-skewed. Meanwhile, some of the categorical variables are highly imbalanced and have numerous categories (more than six, specifically). For example, *native-country* has 41 categories, but the majority of samples are from the United States; *education-num* has 16 categories, with the majority of samples coming from the 9th category, which corresponds to high school graduates. Also, the target variable is highly imbalanced, with the majority of samples corresponding to individuals earning less than 50K.

We determine the dataset's complexity with respect to its categorical attributes by calculating the maximum number of possible combinations, which is 4,762,800. This high value indicates that the likelihood of generating synthetic entries that exactly replicate the original records is low.

5.1.2 Bank Dataset

The Bank dataset [MRC12] contains information collected through phone calls from Portuguese bank clients during a marketing campaign. The variables include general information about clients, such as their age, gender, marital status, and occupation; information about their financial situation, such as their average balance, loans, and credits; and information related to the bank's marketing campaigns, such as the number of times a person was contacted, the duration of the phone calls, and the communication type, among others. The target variable is *subscribed*, indicating whether a client has subscribed or not to a term deposit. Figure 5.2 shows the distribution of each attribute in this dataset. Several observations can be made with respect to the distributions. Similar to the Adult dataset, several numeric variables are highly left-skewed, including *balance*, *day*, *duration*, *campaign*, *pdays*, and *previous*. Furthermore, the numeric variable *day* exhibits multiple modes, posing a challenge for synthetic data generation methods. On the other hand, categorical attributes have a maximum of 12 categories, and some of them are highly imbalanced, such as *default*, which indicates whether a client has a credit in default, or *subscribed*, which is the target variable.

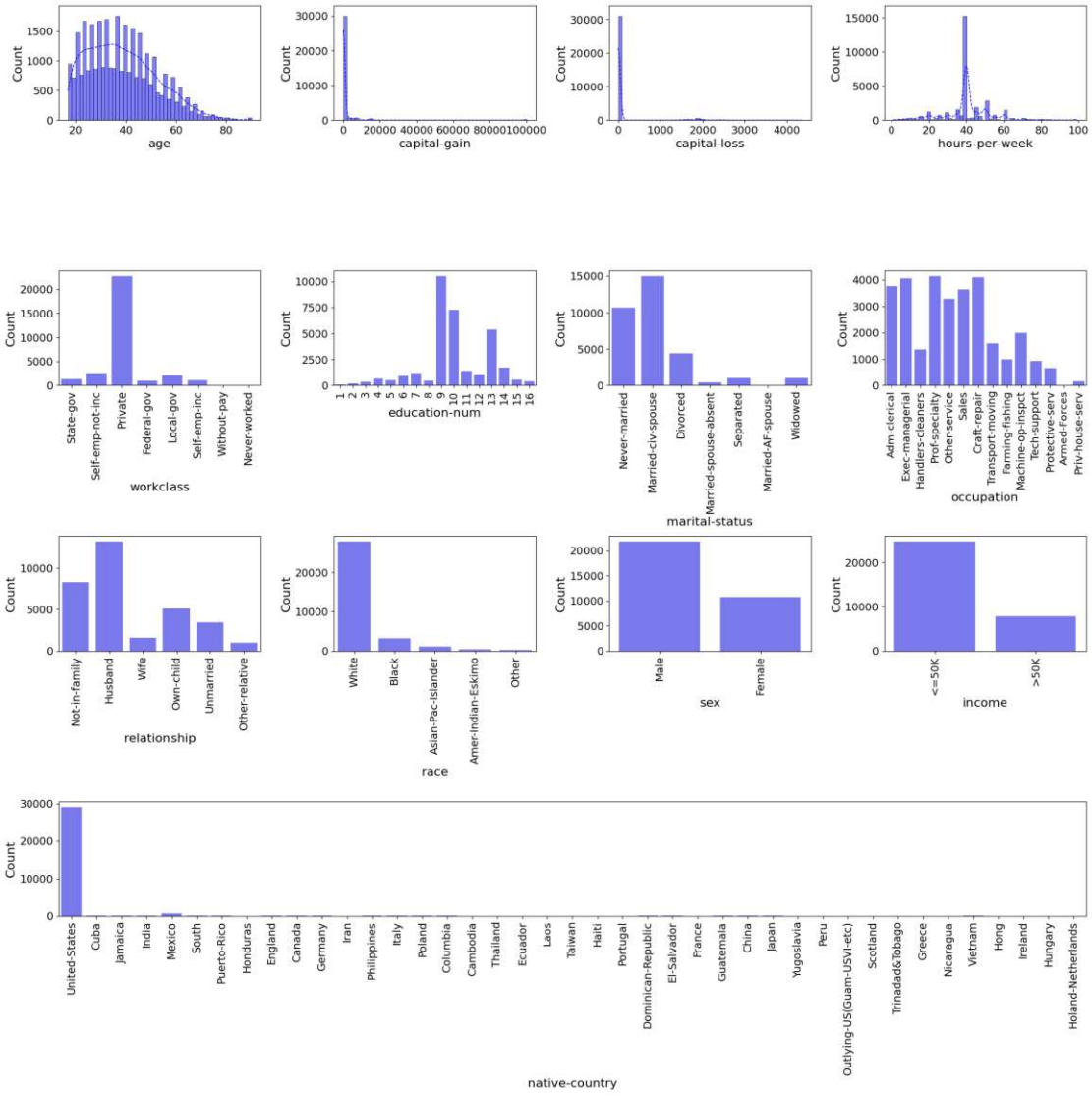


Figure 5.1: Distributions of the features in the Adult dataset

The complexity of the Bank dataset with respect to its categorical attributes is 63,552. Compared to the Adult dataset, this dataset is less complex, meaning there is a higher probability of generating synthetic records with similar entries to the original data.

5.1.3 Cardio Dataset

The Cardiovascular Disease dataset [Kag18] (referred to as the cardio dataset in this work) consists of a collection of 70,000 patient records with information related to cardiovascular diseases. Specifically, it includes demographic information such as age and gender, medical measurements collected when the patient was examined, including blood

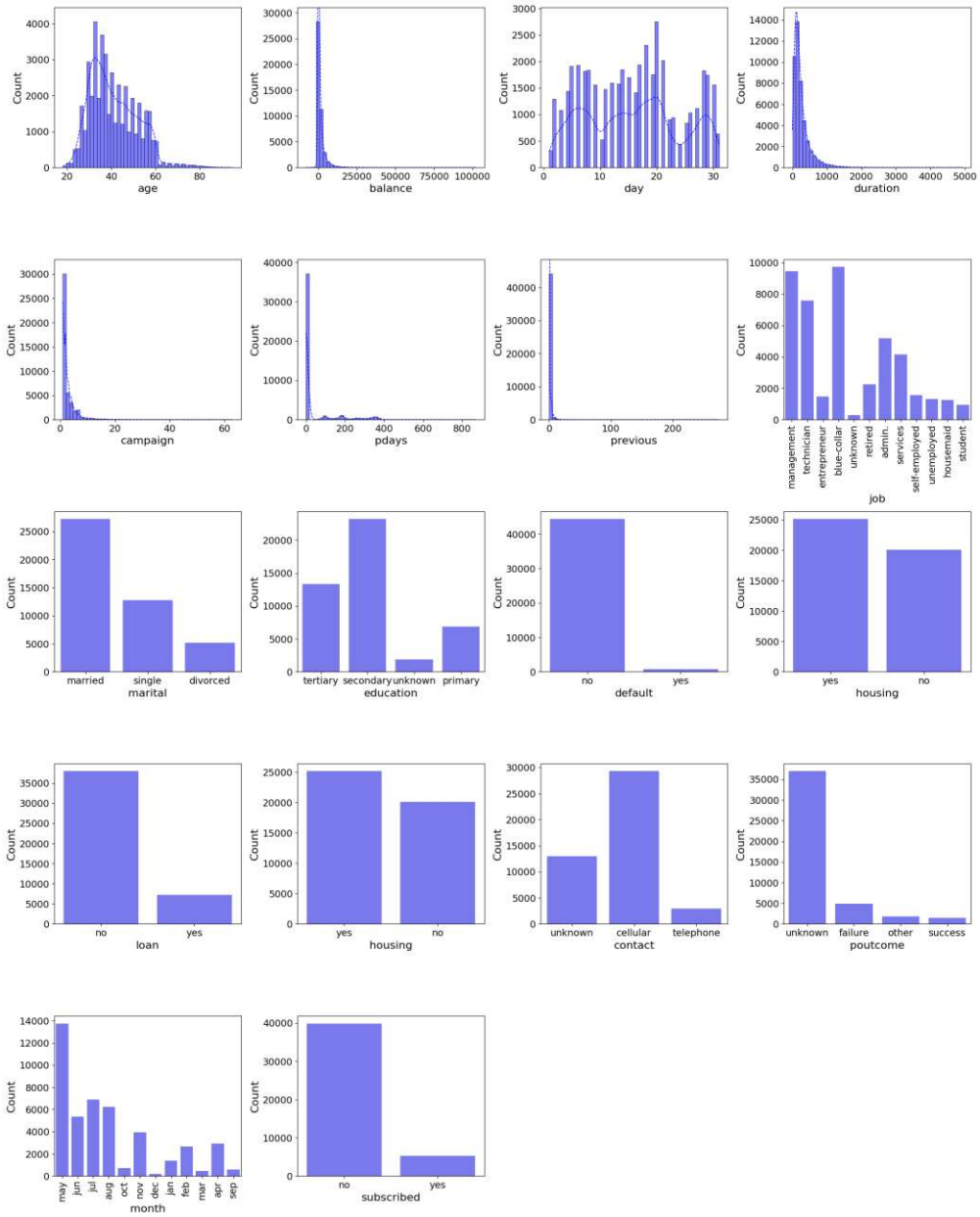


Figure 5.2: Distributions of the features in the Bank dataset

pressure, cholesterol, and glucose, and information provided by the patient with respect to their physical activity and specific habits like smoking and alcohol consumption. The target variable in the dataset is *cardio*, which is a binary variable indicating whether the patient has a cardiovascular disease. Figure 5.2 shows the distribution of each attribute in this dataset. Like in the previous two datasets, we also have numeric variables with highly left-skewed distributions. Namely, *ap_hi* and *ap_low* correspond to a patient's

systolic and diastolic blood pressure readings at the time of the medical examination. The distributions also reveal highly imbalanced categorical attributes such as *smoke*, *alcohol*, and *active*. However, in this dataset, the target variable is balanced, with an equal number of patients with and without cardiovascular disease.

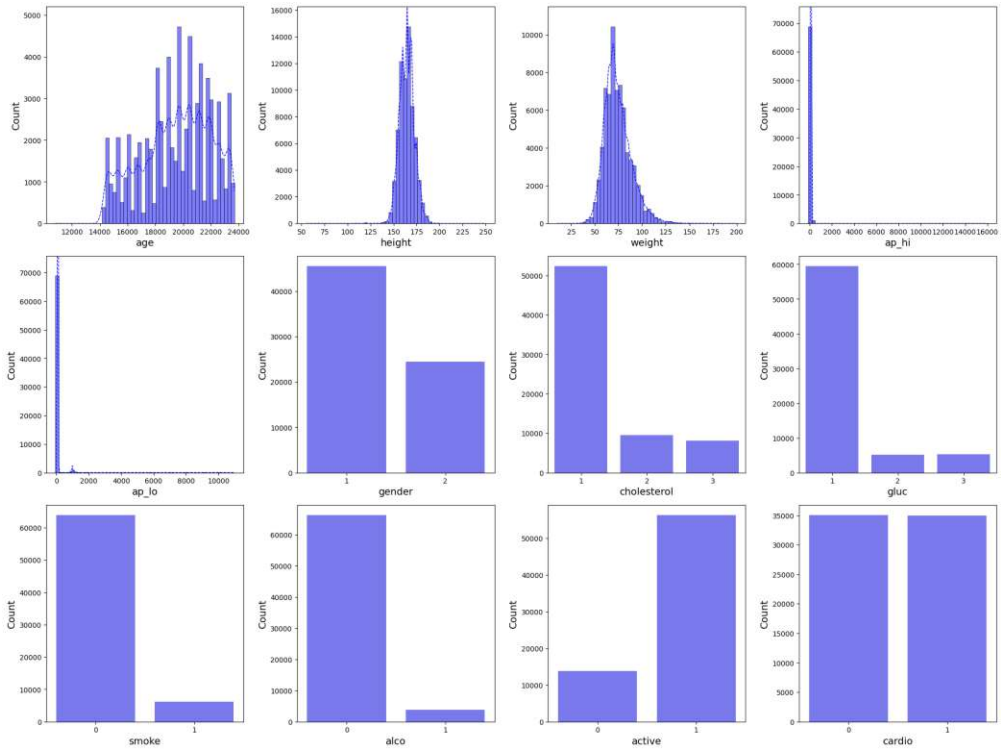


Figure 5.3: Distributions of the features in the Cardio dataset

Now, when considering the complexity of the Cardio dataset, we observe that the categorical attributes have a maximum of three categories. Therefore, this dataset is the least complex among those considered. In this case, the maximum number of possible combinations is 288, which means that it becomes very likely that we generate exact matches or close matches to the original records in the synthetic data.

5.1.4 Preprocessing

Preprocessing is usually a crucial step performed when using a dataset for machine learning purposes. It involves cleaning the raw data and transforming it for the specific task at hand. In the case of synthetic data generation, the goal is to preserve the original data's structure and statistical properties as closely as possible. Previous work suggests that there is no benefit from pre-processing real data prior to synthesizing it [DI21]. In our work, we delete redundant columns or entries with unique values and change the names of some attributes for convenience. In particular, we make the following modifications. For the adult dataset, we deleted the column *fnwl* because it contains only unique values;

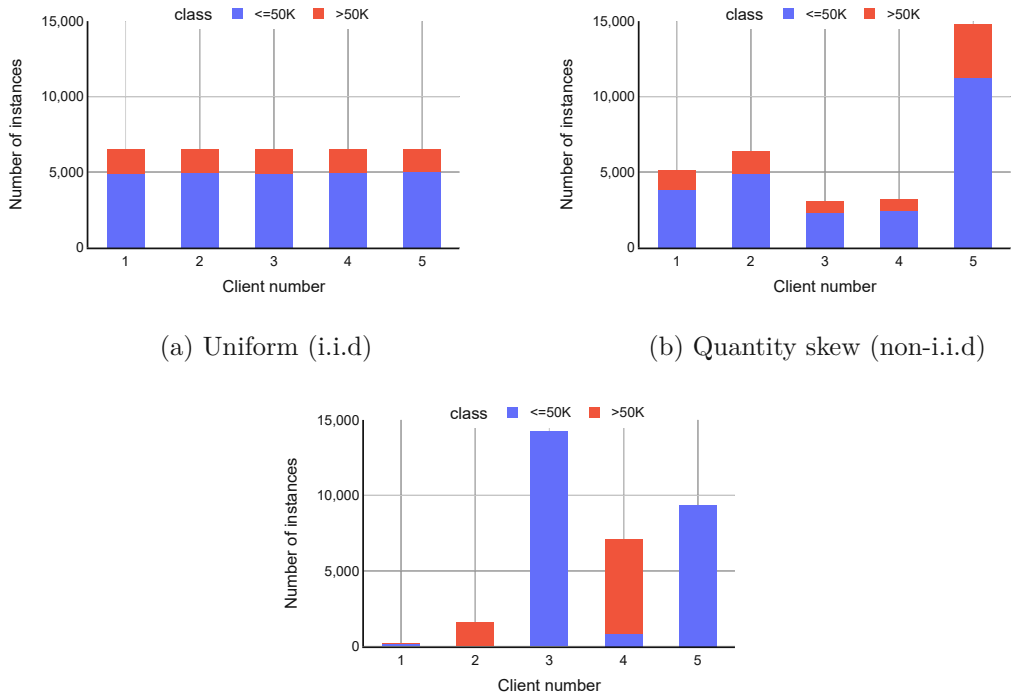
further, the column *education* because it is redundant with *education-num*. In the cardio dataset, we delete the *id* column since it only contains unique entries. Finally, for the bank dataset, no specific preprocessing was required. Furthermore, depending on the synthetic data generation models, we transformed the real data to ensure the input is suitable for training. Specifically, in the BN method, we binned the numeric attributes and label-encoded the categorical attributes. For the VAE, we used Gaussian Mixture Models (GMM) to encode numerical attributes and one-hot encoding for categorical attributes as previously proposed in the literature [XSCIV19].

5.2 Partition of data in different clients

One of the main challenges in federated learning is the heterogeneity of data distributions across different clients. In many applications, real-world data is often not balanced and not independent and identically distributed (non-i.i.d). For example, the distribution of patients from hospitals in different regions may vary significantly, as well as the number of patients with a specific disease [PSA21]. These differences impact the model performance and convergence [CCL23]. Our experiments investigate the effect of training synthetic data generators under i.i.d and non-i.i.d settings. In particular, we analyze two distribution scenarios for the non-i.i.d setting: quantity skew and label distribution skew. These scenarios have been proposed previously to analyze models trained with federated learning [LDCH22]. For simulating the scenarios, we used built-in functions for data partitioning provided by the framework FedLab [ZLH+23]. In the following, we describe in detail how each partition was obtained:

- **Uniform (i.i.d):** In this scenario, the data is partitioned evenly and distributed uniformly among the clients participating in the federation.
- **Quantity skew (non-i.i.d):** In this scenario, the data is partitioned unevenly across the different clients. In particular, this is achieved by sampling the indices of samples assigned to each client from a Dirichlet distribution. The Dirichlet distribution is commonly used as a prior distribution in Bayesian statistics because it is beneficial for modeling proportional data [Hua05]. This distribution is parameterized by a scalar parameter $\alpha > 0$, which controls the concentration of the Dirichlet distribution. The smaller the value of this parameter, the larger the unbalance [LDCH22]. In our experiments, α is set to 0.5 in all scenarios with this partition.
- **Label distribution skew (non-i.i.d):** In this scenario, each client is assigned a subset of the samples, decided by the value of the target variable, using the Dirichlet distribution. In some cases, clients may have samples from only one target variable class. We use $\alpha = 0.5$ in all scenarios with this partition. Additionally, we have another parameter denoted as the *minimum required size*, which controls the minimum number of samples that should be assigned to a client. In our experiments, the *minimum required size* is set to 200.

Figure 5.4 provides an example of the different data distribution settings described in this section for the Adult dataset. Specifically, it shows the distribution of samples with respect to the target variable *income* for the scenario with five clients in the federated setting.



(a) Uniform (i.i.d)

(b) Quantity skew (non-i.i.d)

(c) Label distribution skew (non-i.i.d)

Figure 5.4: Example of the different data distribution settings used in the Adult dataset for five clients with respect to the class label.

5.3 Data Generation in the federated setting

The following procedure was used to prepare the datasets for synthetic data generation in the federated setting and, subsequently, for evaluation:

1. We performed a hold-out method iteratively to ensure consistency of the results. This method generates three random dataset partitions into training and hold-out data with a split ratio of 80:20. Figure 5.5 illustrates this method for one random partition in the left box, where training data is represented in blue and the hold-out data is represented in red.

2. For each split, we simulated a federated scenario by further partitioning the training data into the number of clients specified as shown in the right box of Figure 5.5.

The partition is based on the data distribution specified (cf. Section 5.2). To ensure reproducibility, we assigned a fixed random seed.

3. Finally, we applied the synthetic data generation techniques that we adapted in this thesis to the federated setting for each scenario. We used the final global model to generate synthetic datasets with the same number of samples as the training data. These datasets (represented in yellow) were then used to assess the synthetic data generation methods in terms of fidelity, utility, and privacy. In particular, the utility evaluation used the hold-out data.

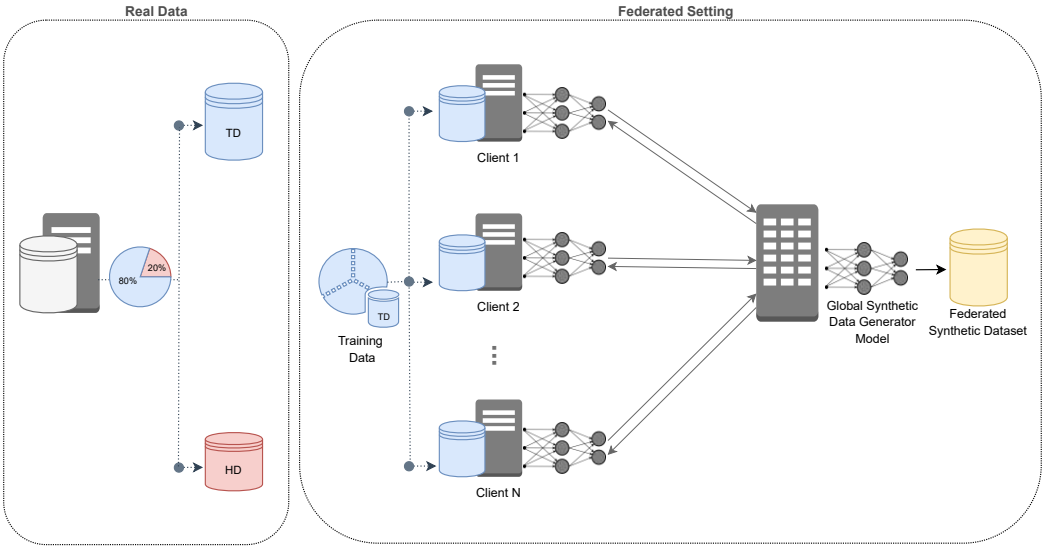


Figure 5.5: Diagram of synthetic data generation in the federated setting

5.4 Baselines

In our work, we consider two common baselines used in the literature to assess the performance of federated learning models:

- *Centralized Baseline:* In this case, we assume that the training data from all the clients can be shared and stored in a central server, which is then used to train a centralized model. This model can generate a given number of synthetic samples. This approach might not be feasible in practice due to privacy concerns and legal restrictions. However, it serves as an upper bound for the performance of federated learning models.
- *Local Baseline:* In this case, we assume that each client trains a model independently with their local data and generates a certain number of synthetic samples. The

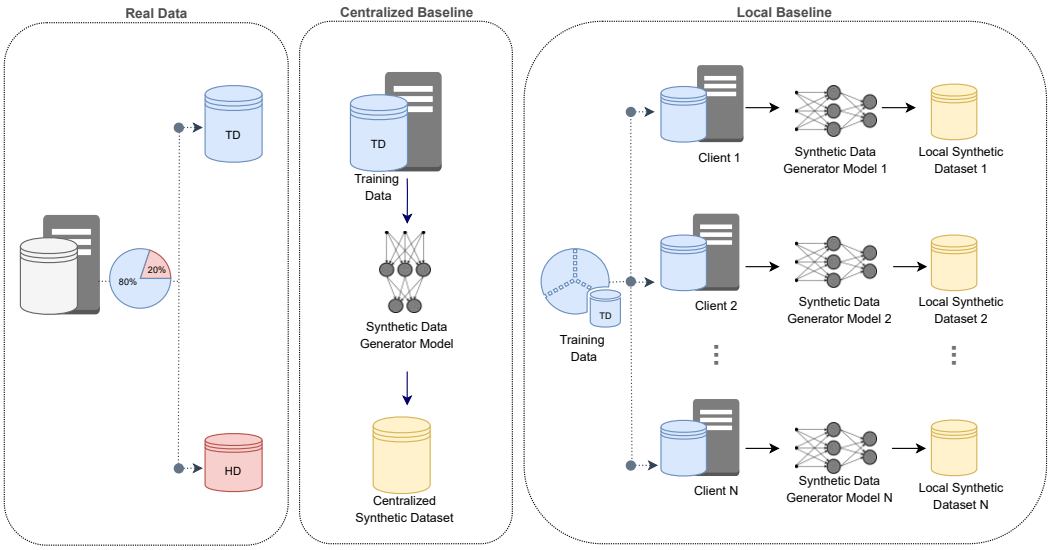


Figure 5.6: Baselines for assessing the performance of the federated approaches proposed in this work.

performance of these datasets, when used for machine learning, is then averaged, serving as a lower bound for our performance evaluation. The expectation is that federated learning outperforms the average results of local models.

Figure 5.6 illustrates the two baselines previously described. It is worth noting that to ensure consistency in the comparison, we use the same three randomly partitioned datasets containing original data, with an 80:20 split between training and testing data for all baselines. The synthetic datasets used for evaluation in each baseline are highlighted in yellow.

Once we have collected the results of all baselines for the three random data partitions in our experiments, we use significance testing when comparing the results. In this context, a commonly used statistical test is the Student's t-test [Stu08], which compares the means of two groups with an approximately normal distribution to determine whether significant differences exist. These test's hypotheses are defined as follows:

$$\begin{aligned} H_0 &: \mu_1 = \mu_2 \\ H_1 &: \mu_1 \neq \mu_2 \end{aligned} \tag{5.1}$$

This test provides a better justification for deciding whether the differences between the baselines are meaningful. If the test indicates a significant difference, we can assess whether one baseline outperforms the other. The Student's t-test returns a p-value, and if the given p-value is below 0.05, we reject the null hypothesis and conclude that the

difference is statistically significant. For this test, we round the means to three decimal places and assume that the variances of the sample groups are equal. Furthermore, since we have two baselines (Centralized and Local), multiple metrics, and different scenarios, we perform the Student's t-test independently for each metric and each scenario. Specifically, we compare the centralized and federated baselines for each metric in a given scenario; we do the same for the local baseline.

5.5 Hyperparameter Selection

Hyperparameter tuning in federated learning is quite challenging [ZFZ+23, Pre23]. In the centralized setting, different configurations are tested on a validation set, and standard methods such as grid search, random search, or Bayesian optimization can be used for hyperparameter selection. Federated learning, meanwhile, involves data distributed across clients, which increases the training overhead and the number of hyperparameters, since both the client and server side must be considered (e.g., local epochs of a client) [MPS24]. Moreover, the evaluation step for deciding the optimal configuration can be more challenging because centralizing data to create a validation set may not be available due to privacy reasons and legal restrictions, and local data at a single client may not accurately represent the overall data distribution [Pre23].

Hyperparameter tuning in federated learning is also an emerging research area [ZFZ+23] with limited works already published. Recent works have proposed using techniques such as reinforcement learning [GYH+22], swarm optimization [LLZ21], and local optimization [MPS24] to find the optimal hyperparameters. Other works propose to adapt the hyperparameters while training the federated model. For instance, Zhang et al. [ZFZ+23] proposed to adjust the hyperparameters in each training round by searching the ones that minimize an objective function considering aspects such as the number of clients, the number of training rounds, and model complexity. Mitic et al. [MPS24] proposed performing hyperparameter tuning in a privacy-preserving manner by letting each client optimize its model locally and then sending the optimal set found, along with a performance metric using multi-party homomorphic encryption, to the server. The server then combines the results from all clients using a specific strategy. Their findings showed that computing the mean of the performance metric for the top five hyperparameter configurations of each client for i.i.d settings works well. In comparison, density-based clustering for non-i.i.d settings yields better results. Many open challenges remain, including efficiency, applicability to multiple models, and the consideration of hyperparameters affecting the federated training.

Apart from the challenges mentioned above, when tuning hyperparameters for synthetic data generation models, we also need to consider that the main goal is to enhance the quality of the generated synthetic data. However, the quality of the synthetic data can be defined in terms of fidelity, utility, and privacy, and multiple metrics can be used to assess these dimensions. Therefore, we also need to choose a metric to optimize based on the specific use case of the synthetic data. One way to select this metric is by investigating

the correlation between metrics, as demonstrated by Basri et al. [BHA+23], and then choosing the one with the strongest correlations to other metrics.

It is beyond the scope of this work to determine the best strategy for hyperparameter optimization of synthetic data generation models in a federated setting. Therefore, we do not explore this topic in detail. Instead, in the implementation, we treat federated learning models as black boxes and perform standard methods such as grid search and random search over a grid of hyperparameters defined for the client and server sides using Weight& Biases[1]. We use the Log-Cluster metric as the objective function, as it provides a global assessment of the similarity of the latent structure of real and synthetic data. All configurations were evaluated against the local data on each client after the global model was trained, and the average results were then calculated on the server. In this final step, we assume we have access to a SMPC protocol for the computation.

For our FedBN approach introduced in Section 3.3, we performed a grid search with the hyperparameter values presented in Table 5.2. The hyperparameters P and S are derived from the genetic algorithm used in the structure learning step of the Bayesian Network (BN) approach. The hyperparameter T is associated with the server in the federated setting and directly impacts the aggregation of the client results, mainly the aggregation frequency.

Table 5.2: Hyperparameter grid for the FedBN approach

Hyperparameter	Possible Values
Population Size P	100,150,200
Selection Pressure S	5,10,15
Aggregation Interval T	5,10,20,25

In contrast to the BN approach proposed, the VAE approach in the federated setting has more hyperparameters that can significantly influence the model's performance, making the tuning process more complex. Therefore, we used a random grid search with 30 iterations to limit the number of configurations since running each iteration is computationally expensive. The hyperparameters were also selected depending on the characteristics of the datasets. The grid used for the Adult and Bank datasets was the same and is shown in Table 5.3. Meanwhile, the grid used in the Cardio dataset, which has fewer columns, is shown in Table 5.4.

Figure 5.7 shows the results of the hyperparameter tuning in the FedBN approach for each dataset using five clients with a non-i.i.d data partition, specifically with a label skew distribution. The optimal configuration is obtained by minimizing the Mean Log Cluster metric from three runs with different data partitions. The aim is to find a model that generates synthetic data that closely resembles real data in its latent structure. The model that provides the best results for the Adult dataset (Figure 5.7a) obtained a score of -19.0. On the other hand, for the Bank dataset (Figure 5.7b), the best score is -21.68.

[1] https://wandb.ai/

Table 5.3: Hyperparameter grid for the FedVAE approach for the Adult and Bank datasets

Hyperparameter	Possible Values
Loss Factor	2,4,6,8
Local Epochs	1,5,10,20
Global Epochs	50,100,200,300
Batch size	50,100,200,300
Embedding dimension	16,32,56,64
Compress dimension	[128],[64],[128,128],[128, 64],[64, 64]
Weight Decay	0.00001, 0.0001, 0.001, 0.01, 0.1

Table 5.4: Hyperparameter grid for the FedVAE approach for the Cardio dataset

Hyperparameter	Possible Values
Loss Factor	2,4,6,8
Local Epochs	1,5,10,20
Global Epochs	50,80,100
Batch size	100,200,300
Embedding dimension	5,10,20,25,30
Compress dimension	[32],[56],[64],[64, 32],[64, 56]
Weight Decay	0.00001, 0.0001, 0.001, 0.01, 0.1

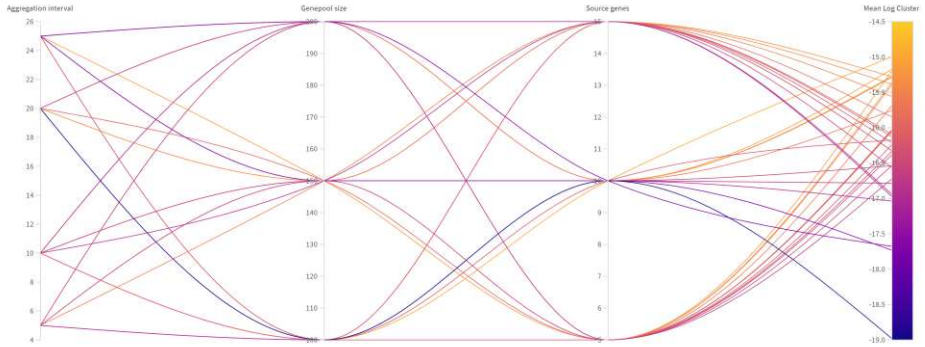
Finally, the best score for the Cardio dataset (Figure 5.7c) is -12.679. In particular, for this last dataset, the differences between multiple hyperparameter configurations are smaller compared to the best score, unlike in the other two datasets.

Table 5.5 summarizes the optimal hyperparameter configurations for each dataset according to the tuning process of the FedBN approach. Note that in our experiments, we used the same hyperparameters in all the scenarios considered in the federated setting with different numbers of clients and data partitions to ensure consistency. However, depending on the use case of the synthetic data, it would be beneficial to run the process for each scenario. We can conclude from the table that the hyperparameters are highly dependent on the dataset.

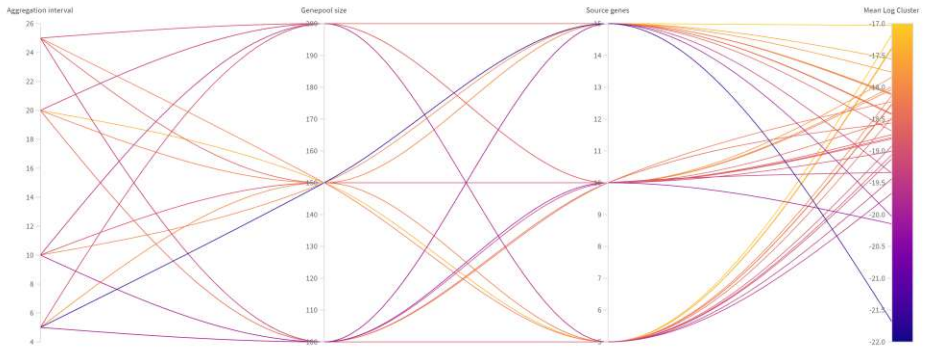
Table 5.5: Best training hyperparameter settings for FedBN

Dataset	Aggregation Interval	Population Size	Selection Pressure
Adult	20	100	10
Bank	10	200	10
Cardio	25	150	15

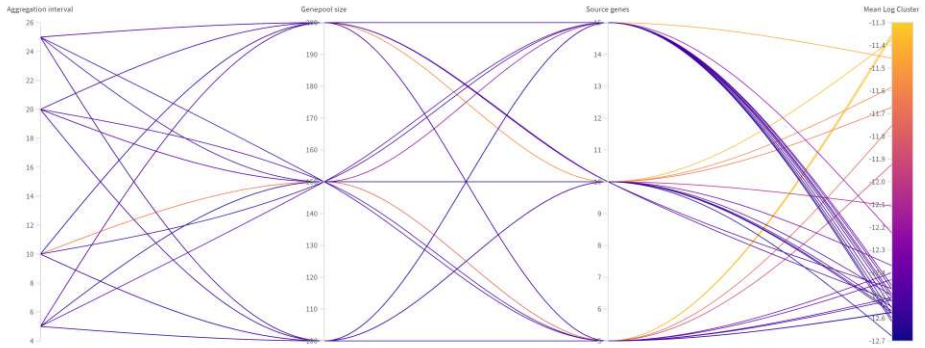
Similar to the approach used for the FedBN model, we also perform hyperparameter tuning for the FedVAE model for each dataset using five clients with a non-i.i.d data partition, specifically with a label skew distribution. Figure 5.8 shows the results obtained



(a) Adult Dataset



(b) Bank Dataset



(c) Cardio Dataset

Figure 5.7: Results of hyperparameter tuning to determine the optimal configuration for training the FedBN model in a scenario with five clients using non-i.i.d data partitions with label distribution skew for the different datasets. Each line is a hyperparameter configuration. The vertical axes represent the hyperparameter values, with the rightmost axis depicting the performance metric (Mean Log Cluster) used for selecting the optimal configuration.

for all datasets. As we can see, the FedVAE model explores a more extensive range of hyperparameters. The main reason is that VAEs are composed of neural networks. Therefore, we have hyperparameters related to the architecture (e.g., the embedding dimension and compression dimensions), the training of the local models (e.g., the batch size, the weight decay, and the loss factor), as well as parameters directly tied to the federated learning process (e.g., global epochs and local epochs of the clients). It is necessary to clarify that the compression dimension is a list of numbers indicating the number of hidden layers and the dimension of each layer in the encoder. In our implementation, we replicate the architecture of the encoder in the decoder, following the approach proposed by Xu et al. [XSCIV19].

As shown in Figure 5.8a, the best-performing model for the Adult dataset obtained a score of -20.927. On the other hand, for the Bank dataset (Figure 5.8b) and the Cardio dataset (Figure 5.8c), the best scores are -18.118 and -15.969 respectively. It is apparent from the figures that smaller values in the weight decay yield better results in all datasets. This is not surprising, as weight decay imposes a penalty on the model's weights. Smaller weight decay values result in a less severe penalty, allowing the model to learn better while benefiting from regularization. We also observe that the differences between hyperparameter configurations near the optimal set are smaller for this model than the FedBN model. Especially for the Bank dataset, the closest configuration to the optimal hyperparameter set has a difference of approximately 0.06 in terms of the evaluation metric used.

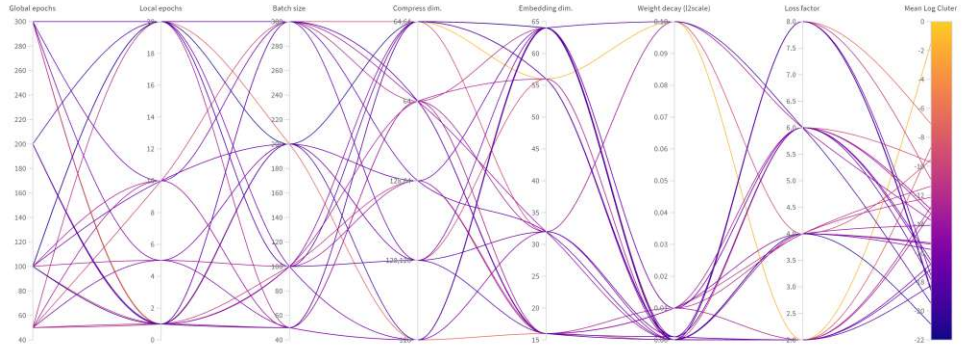
Table 5.6: Best training hyperparameter settings for FedVAE

Dataset	Global Epochs	Local Epochs	Batch Size	Compress Dim.	Embedding Dim.	Weight Decay	Loss Factor
Adult	100	20	200	[64,64]	64	0.0001	4
Bank	200	20	200	[128]	56	0.001	4
Cardio	50	1	100	[64,56]	20	0.001	2

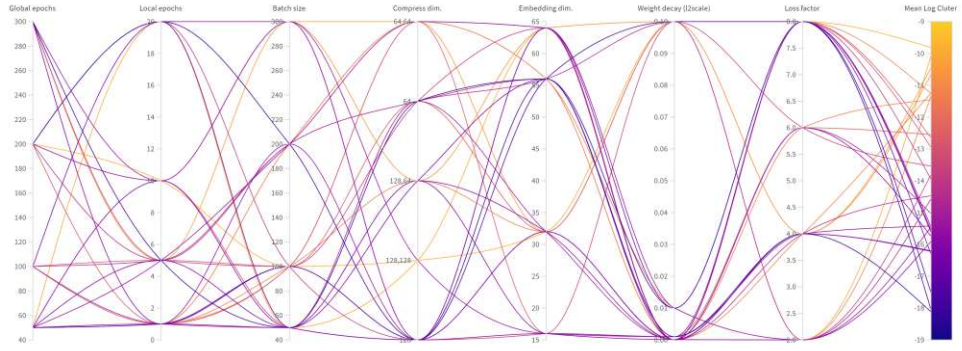
Table 5.5 presents the best training hyperparameter settings for the FedVAE model for all datasets. In a similar fashion, as described for the FedBN model, the hyperparameters reported in the table are the ones we used for all the scenarios explored (i.e., with different numbers of clients and data partitions). The only remark is that the global epochs were modified in the scenarios using local differential privacy due to problems with the model convergence. A detailed explanation is provided in Chapter 6.

5.6 Evaluation

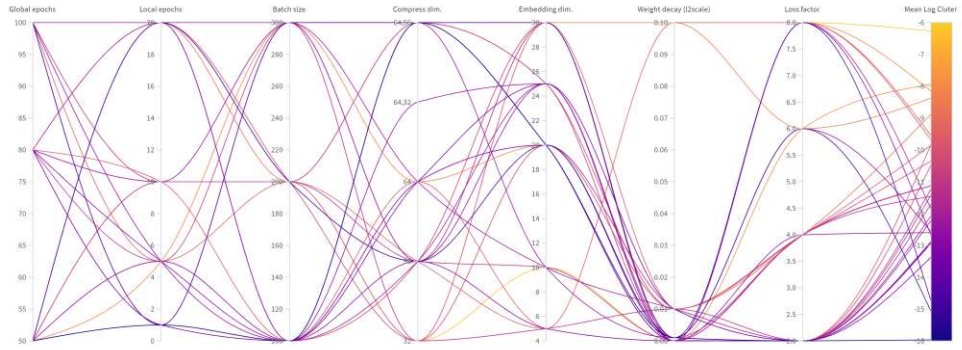
We evaluate properties of synthetic data across three dimensions: fidelity, utility, and privacy. We investigate popular metrics used in the literature for each dimension and select ones that cover multiple aspects of the synthetic data. The metrics chosen are depicted in Figure 5.9 grouped by the dimension they belong to, and a more detailed



(a) Adult Dataset



(b) Bank Dataset



(c) Cardio Dataset

Figure 5.8: Results of hyperparameter tuning to determine the optimal configuration for training the FedVAE model in a scenario with five clients using non-i.i.d data partitions with label distribution skew for the different datasets. Each line is a hyperparameter configuration. The vertical axes represent the hyperparameter values, with the rightmost axis depicting the performance metric (Mean Log Cluster) used for selecting the optimal configuration.

categorization that is based on further attributes. Furthermore, the definitions for each metric are provided in Section 2.1.3.

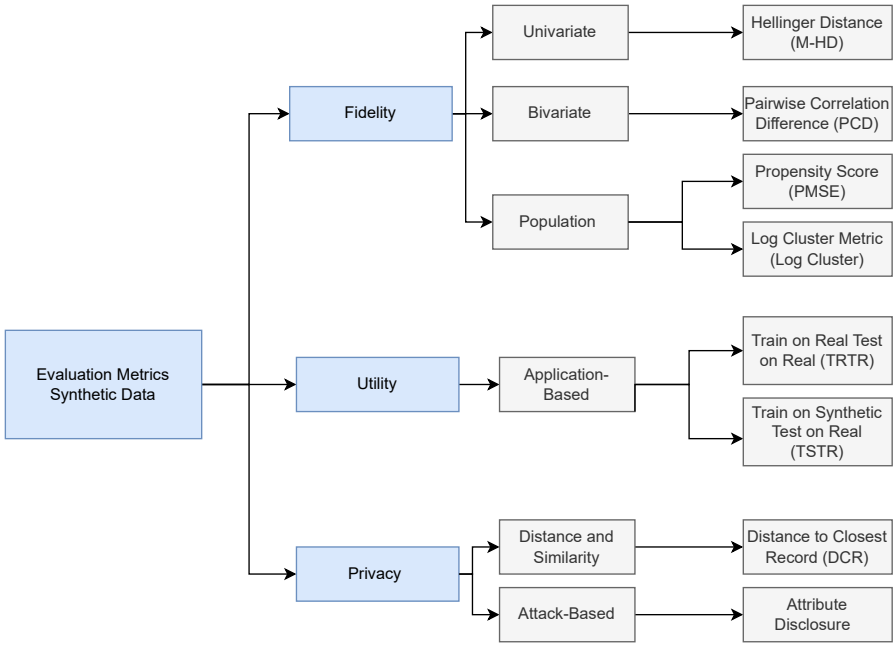


Figure 5.9: Metrics used for evaluation

5.6.1 Privacy Risk Assessment

The two metrics considered to evaluate the privacy of synthetic data are Distance to Closest Record (DCR) and Attribute Disclosure (AD). The literature analyzes these metrics in multiple ways. Therefore, we provide a detailed explanation of the methodology used in our work.

For the DCR metric, we consider the hold-out assessment proposed in [PR21]. This assessment compares the DCR results obtained from synthetic and training data with those from the holdout data. Specifically, it analyzes the average DCRs and calculates the *share of records* that are closer to the training data than to the hold-out data. A share close to 0.5 indicates that the distances to the training and hold-out data are similar. This suggests that the synthetic data does not reveal whether a particular record was part of the training data.

In the proposed assessment, the authors split the real data into training and holdout datasets with a 50:50 ratio. However, in our case, the holdout dataset contains only 20% of the real records. Therefore, to avoid bias when computing the *share of records*, and in lack of a directly comparable baseline, we use bootstrapping to sample an equal number of records from the training data across ten iterations and calculate the mean share of all iterations. This procedure is repeated for each random split used in the experiments.

Additionally, we calculate the minimum distance of a synthetic record to a training record to check for exact matches (i.e., records with DCR=0).

On the other hand, for the attribute disclosure experiments, we follow the approach presented in [HME20], which defines possible attack scenarios for a given dataset. Each scenario assumes that the attacker has access to a set of quasi-identifiers from an individual that might be in the original dataset and tries to learn the value of a sensitive attribute, denoted as the target attribute, using the synthetic dataset provided. The attack is performed as a classification task, and the risk is estimated for each record in the original dataset. We use the machine learning algorithms previously reported in the centralized setting [HME20] to perform the attack. All the algorithms are trained using the default parameters from the Python library (scikit-learn). Furthermore, we use two baselines to estimate the disclosure risk: the real data and the dummy classifier. The real data serves as an upper bound on the performance of the synthetic data, and the dummy classifier, which predicts the most frequent class, serves as a lower bound. In particular, we are interested in the risk reduction provided by the synthetic data with respect to the real data.

The scenarios defined for each dataset are depicted in Table 5.7. In each scenario, we define the key length, denoted as n, which represents the number of quasi-identifiers the attacker is assumed to know about a particular individual from one of the client's datasets. Note that the individual may or may not be part of these datasets. We also define a quasi-identifier set, denoted as QI, which can be larger than the key length, to simulate the attacker's knowledge and a target attribute representing sensitive information about the target individual. After defining these elements, we run the aforementioned ML algorithms and compute the accuracy for all combinations of QI with the given key length using the synthetic data as the training set and real data as the test set. Finally, we report the average and standard deviation of all the runs.

Scenario	Dataset	Key-length(n)	Quasi-Identifiers(QI)	Target(t)
1	Adult	3	{'age, marital-status', 'sex','race'}	marital-status
2	Bank	3	{'age', 'job', 'marital', 'education'}	housing
3	Cardio	2	{'age','gender','height'}	cholesterol

Table 5.7: Attribute Disclosure Scenarios

5.7 Experimental Setup

The experiments for this thesis were performed in Python 3.9 using the Flower Library version 1.6 with the virtual engine available for simulations. Since the models we used have different characteristics, the experiments were run on two machines.

For the BN model, we used a server with the following specifications:

- CPU: 56 cores (with hyperthreading), Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz

- RAM: 256 GB
- Operating System: Ubuntu Linux

On the other side, for the VAE model, we used GPUs with the following specifications:

- GPU: NVIDIA RTX 2080 Ti
- Operating System: Ubuntu Linux

The code that has been used for each of the methods will be available at the following repositories:

- `https://gitlab.sba-research.org/machine-learning/federated-ga-datasynthesizer` (FedBN approach)
- `https://github.com/danielamartinezd02/Federated-SyntheticTabularDataGeneration` (FedVAE approach)

CHAPTER 6

Results and Evaluation

This chapter analyzes the results obtained following the experiment design described in Chapter 5. It is organized as follows. The first part investigates the quality of the synthetic data generated with Bayesian Networks in the distributed setting. The quality is analyzed in terms of fidelity, utility, and privacy. The results include different settings and comparisons against two baselines, namely the centralized and the local setting. The second part of the results presents the VAE experiments conducted in the distributed setting. It also includes an analysis of different settings and metrics in multiple dimensions. The last part compares the Bayesian Networks and the VAEs in the distributed setting. One important aspect to note in this analysis is that a centralized dataset to assess synthetic data quality is usually not available in a distributed setting. Therefore, the evaluation is much more challenging, and one must consider combining the local assessment to get a global overview. For experimental purposes, however, we use the real centralized dataset to benchmark the performance of the federated setting, as it is commonly done in the literature [DLH+23, ZBKC21, FV22]. In real scenarios, due to privacy concerns, SMPC and other privacy-preserving techniques could be leveraged to evaluate this baseline performance.

To run the experiments, we used the hyperparameter settings presented in Section 5.5 for each dataset. To provide statistically more robust evidence for the analysis, we further contact statistical significance testing on the differences of the classifier performance, as explained in Section 5.4. Since there are two baselines, we conducted two separate tests for each of them, i.e., the first test compares the federated baseline against the centralized baseline, while the second test compares the local baseline against the federated setting. The results of these tests are color-coded for the tables following in this chapter. If the differences are statistically significant, we highlight the result in green if it performs better or in red if it performs worse. Otherwise, we do not highlight the result. As there are two baselines, the color-coding in rows showing federated (labeled as "Fed") results indicate the significance of the federated result compared to the centralized result.

Likewise, in rows providing local results (labeled as "Loc"), the color-coding indicates the performance of federated learning against the local baseline.

To improve readability, this chapter presents only the tables and figures that offer new insights. The remaining results are provided in the Appendix (see the Appendix A).

6.1 Federated Bayesian Networks

6.1.1 Fidelity Evaluation

The aim of the fidelity evaluation is to determine whether the synthetic data preserves the structure and statistics of the real data. This section analyses fidelity results of 27 federated learning scenarios tested on three datasets. The scenarios combine settings for different numbers of clients NC $\in [3, 5, 10]$, different data partitions (i.i.d, non-i.i.d quantity skew, and non-i.i.d label skew), and two privacy budgets ($\epsilon = 0.5$ and $\epsilon = 1.6$). We also include scenarios without DP to showcase the impact of adding privacy-preserving techniques on the synthetic data generation process. The metrics used for the fidelity evaluation are the following: mean Hellinger distance (M-HD), pair correlation difference (PCD), propensity mean squared error (pMSE), and log-cluster metric (LC). A detailed description of each metric was provided in Section 2.1.3. The result for each metric shows the mean and the standard deviation across three different iterations with distinct splits of the real data. Note that the lower the value of all the metrics, the better the performance in terms of fidelity. It should be emphasized that the epochs and other applicable hyperparameters for the two baselines are set to the same values, and the number of synthetic samples generated in each case is equal to the total number of samples across all clients.

Table 6.1 shows the results on the Adult dataset. It can be observed from these results that the variations in the metrics M-HD and pMSE across different numbers of clients and data partitions are not significant within the federated setting for the same epsilon value. In contrast, PCD and LC metrics show significant differences across various scenarios. When comparing the different results across the scenarios, we can observe no significant differences in most cases between the centralized and federated settings for the non-DP case. This means that the fidelity of the synthetic data is comparable for both baselines or slightly better as desired. On the other hand, statistical tests reveal significant differences between the federated and local settings in two fidelity metrics: M-HD and LC. Notably, in the non-i.i.d, label skew partitions, federated results outperform the average local results for LC when using 5 and 10 clients.

Interestingly, the synthetic data generated with a privacy budget $\epsilon = 1.6$ for the Adult dataset in the federated setting shows significant differences compared to the centralized and local settings. Specifically, the federated setting outperforms the centralized setting in terms of PCD and LC across six scenarios. Additionally, it provides comparable results for these metrics in the rest of the scenarios with the centralized setting. Meanwhile, the local setting results for this privacy budget are worse than the federated setting for all

Table 6.1: Fidelity results for the Adult dataset with 3, 5, and 10 clients under different data partitions (i.i.d, non-i.i.d quantity skew, and non-i.i.d label skew) using FedBN. Statistical significance is highlighted as follows: green indicates significantly better performance than the baseline, and red indicates worse performance than the baseline. Federated results are compared to the centralized one, while local results are compared to the federated ones.

		Model BN											
		Non-DP				DP ($\epsilon = 0.5$)				DP ($\epsilon = 1.6$)			
NC	Bas.	M-HD ↓	PCD ↓	pMSE ↓	LC ↓	M-HD ↓	PCD ↓	pMSE ↓	LC ↓	M-HD ↓	PCD ↓	pMSE ↓	LC ↓
	Cen	0.186±0.000	0.409±0.077	0.249±0.000	-15.328±1.135	0.419±0.006	2.045±0.021	0.250±0.000	-3.031±0.010	0.379±0.009	1.788±0.022	0.250±0.000	-3.116±0.004
i.i.d													
3	Fed	0.185±0.000	0.365±0.024	0.249±0.000	-17.772±0.739	0.451±0.006	2.205±0.050	0.250±0.000	-3.027±0.002	0.374±0.035	1.634±0.184	0.250±0.000	-3.201±0.071
	Loc	0.190±0.001	0.363±0.041	0.249±0.000	-17.470±2.189	0.454±0.006	2.126±0.043	0.250±0.000	-3.023±0.016	0.418±0.010	1.874±0.070	0.250±0.000	-3.045±0.022
5	Fed	0.186±0.000	0.318±0.023	0.249±0.000	-16.969±0.928	0.426±0.010	1.930±0.094	0.250±0.000	-3.031±0.003	0.319±0.010	1.259±0.108	0.249±0.000	-3.430±0.040
	Loc	0.192±0.001	0.408±0.070	0.249±0.000	-16.673±4.456	0.468±0.005	2.254±0.042	0.250±0.000	-3.027±0.031	0.448±0.008	2.083±0.058	0.250±0.000	-3.032±0.028
10	Fed	0.186±0.000	0.502±0.117	0.249±0.000	-19.161±3.942	0.457±0.003	2.194±0.070	0.250±0.000	-3.036±0.016	0.378±0.018	1.615±0.039	0.250±0.000	-3.281±0.097
	Loc	0.196±0.002	0.558±0.099	0.249±0.000	-14.572±2.315	0.471±0.005	2.352±0.066	0.250±0.000	-3.015±0.035	0.462±0.008	2.279±0.099	0.250±0.000	-3.017±0.035
non-i.i.d (Quantity Skew)													
3	Fed	0.186±0.000	0.320±0.027	0.249±0.000	-18.608±2.186	0.442±0.006	2.096±0.099	0.250±0.000	-3.034±0.011	0.349±0.011	1.443±0.132	0.249±0.000	-3.269±0.038
	Loc	0.190±0.001	0.348±0.023	0.249±0.000	-15.550±2.166	0.461±0.008	2.116±0.052	0.250±0.000	-3.021±0.027	0.431±0.015	1.903±0.089	0.250±0.000	-3.037±0.026
5	Fed	0.186±0.000	0.387±0.059	0.249±0.000	-17.470±1.750	0.441±0.011	2.066±0.126	0.250±0.000	-3.023±0.011	0.349±0.036	1.421±0.205	0.250±0.000	-3.337±0.134
	Loc	0.194±0.003	0.489±0.091	0.249±0.000	-16.501±4.020	0.464±0.012	2.282±0.122	0.250±0.000	-3.021±0.031	0.447±0.024	2.158±0.191	0.250±0.000	-3.031±0.031
10	Fed	0.185±0.000	0.505±0.105	0.249±0.000	-18.215±1.364	0.458±0.004	2.227±0.088	0.250±0.000	-3.029±0.008	0.376±0.018	1.596±0.076	0.250±0.000	-3.287±0.060
	Loc	0.179±0.026	0.716±0.295	0.236±0.029	-14.141±2.497	0.462±0.011	2.360±0.101	0.249±0.002	-2.992±0.069	0.453±0.010	2.286±0.163	0.249±0.002	-2.996±0.072
non-i.i.d (Label Skew)													
3	Fed	0.186±0.001	0.369±0.028	0.249±0.000	-16.556±1.484	0.455±0.015	2.247±0.135	0.250±0.000	-3.019±0.007	0.376±0.029	1.622±0.167	0.250±0.000	-3.194±0.061
	Loc	0.247±0.035	1.045±0.535	0.249±0.000	-13.129±2.142	0.460±0.012	2.306±0.160	0.250±0.000	-3.024±0.017	0.437±0.019	2.153±0.179	0.250±0.000	-3.047±0.015
5	Fed	0.185±0.000	0.507±0.167	0.249±0.000	-18.090±2.149	0.448±0.019	2.184±0.194	0.250±0.000	-3.026±0.007	0.361±0.045	1.561±0.275	0.250±0.000	-3.270±0.136
	Loc	0.257±0.032	1.116±0.592	0.248±0.003	-12.627±2.092	0.463±0.017	2.372±0.226	0.250±0.000	-3.008±0.043	0.445±0.021	2.263±0.255	0.250±0.000	-3.022±0.046
10	Fed	0.186±0.001	0.554±0.104	0.249±0.000	-20.346±3.426	0.457±0.005	2.187±0.104	0.250±0.000	-3.032±0.015	0.385±0.017	1.653±0.064	0.250±0.000	-3.255±0.066
	Loc	0.274±0.048	1.261±0.444	0.248±0.003	-12.139±2.191	0.469±0.009	2.452±0.101	0.250±0.000	-2.991±0.078	0.460±0.014	2.392±0.131	0.250±0.000	-2.999±0.084

the metrics except for pMSE, excluding the scenario with three clients and i.i.d partition. These observations suggest that using distributed DP for synthetic data generation in the federated setting can provide better results than DP applied in the centralized setting in some instances. A possible explanation is that local data requires less noise to achieve the same privacy level, leading to more useful information being preserved and better results after aggregation, especially for certain privacy budgets. However, further experiments are required to validate these findings.

The results obtained for the Bank dataset (Table 6.2) show statistically significant differences in multiple metrics between the federated results and the centralized baseline for the non-DP case. Specifically, the federated results outperform the centralized baseline in the M-HD metric for the non-DP version in 6 out of 9 scenarios and the PCD metric in 3 out of 9 scenarios, achieving comparable performance in the remaining cases. However, the pMSE is worse than the centralized baseline in 8 out of 9 scenarios. Despite these statistically significant differences, the variations in M-HD and pMSE are only 0.01 in all cases, which is not meaningful given the range of these metrics. However, the PCD results suggest that synthetic data in the federated setting may better capture correlations compared to the centralized setting. Additionally, when comparing the local baseline and federated settings, we can observe that the average results in the local baseline even outperform federated learning across multiple metrics in the non-DP case for the i.i.d and non-i.i.d quantity skew data partitions. This can occur when the distribution of client's data resembles the distribution of aggregated (centralized) data – then, federated learning provides a more general model, that is however not better fitting for the specific

local data. In such cases, the federated approach may not provide additional benefits over local data generation.

Table 6.2: Fidelity results for the Bank dataset with 3, 5, and 10 clients under different data partitions (i.i.d., non-i.i.d. quantity skew, and non-i.i.d. label skew) using FedBN. Statistical significance is highlighted as follows: green indicates significantly better performance than the baseline, and red indicates worse performance than the baseline. Federated results are compared to the centralized one, while local results are compared to the federated ones.

Model BN													
		Non-DP				DP ($\epsilon = 0.5$)				DP ($\epsilon = 1.6$)			
NC	Bas.	M-HD ↓	PCD ↓	pMSE ↓	LC ↓	M-HD ↓	PCD ↓	pMSE ↓	LC ↓	M-HD ↓	PCD ↓	pMSE ↓	LC ↓
	Cen	0.148±0.000	1.039±0.080	0.226±0.000	-20.746±2.368	0.379±0.026	1.895±0.229	0.246±0.001	-8.087±0.371	0.320±0.038	1.780±0.052	0.245±0.002	-9.513±1.052
i.i.d													
3	Fed	0.147±0.000	0.928±0.036	0.227±0.000	-18.020±0.505	0.423±0.013	2.262±0.243	0.248±0.000	-7.715±0.256	0.349±0.050	1.803±0.227	0.245±0.002	-9.584±2.283
	Loc	0.143±0.006	0.794±0.255	0.224±0.004	-20.021±3.024	0.404±0.005	2.131±0.073	0.247±0.001	-7.708±0.171	0.360±0.009	1.761±0.059	0.246±0.001	-8.290±0.291
5	Fed	0.147±0.000	0.890±0.020	0.227±0.000	-19.807±1.030	0.423±0.001	2.334±0.013	0.248±0.000	-7.412±0.097	0.330±0.001	1.759±0.029	0.241±0.003	-8.258±0.050
	Loc	0.138±0.006	0.703±0.224	0.221±0.004	-19.211±1.574	0.420±0.005	2.439±0.096	0.247±0.001	-7.416±0.137	0.404±0.011	2.252±0.183	0.246±0.001	-7.556±0.181
10	Fed	0.147±0.000	0.884±0.005	0.227±0.000	-21.758±1.113	0.414±0.004	2.371±0.120	0.248±0.000	-7.515±0.131	0.330±0.028	1.950±0.068	0.240±0.001	-8.353±0.660
	Loc	0.135±0.006	0.569±0.106	0.219±0.003	-17.904±2.535	0.420±0.005	2.553±0.076	0.246±0.001	-7.445±0.083	0.413±0.009	2.485±0.149	0.246±0.001	-7.465±0.136
non-i.i.d (Quantity Skew)													
3	Fed	0.148±0.000	0.953±0.049	0.227±0.000	-18.496±0.801	0.406±0.014	2.148±0.223	0.248±0.000	-7.812±0.334	0.298±0.031	1.702±0.193	0.245±0.003	-9.971±1.059
	Loc	0.139±0.007	0.756±0.177	0.222±0.004	-18.032±1.560	0.409±0.015	2.241±0.308	0.247±0.001	-7.915±0.512	0.379±0.035	2.038±0.390	0.246±0.001	-8.549±0.885
5	Fed	0.147±0.000	0.894±0.008	0.227±0.000	-19.657±1.583	0.432±0.001	2.555±0.016	0.248±0.000	-7.390±0.046	0.391±0.001	2.073±0.007	0.244±0.000	-7.716±0.053
	Loc	0.138±0.006	0.614±0.162	0.221±0.004	-19.614±2.927	0.419±0.006	2.468±0.139	0.247±0.001	-7.507±0.160	0.41±0.013	2.336±0.233	0.205±0.017	-8.514±0.541
10	Fed	0.147±0.000	0.965±0.079	0.227±0.000	-19.513±1.377	0.417±0.009	2.445±0.112	0.248±0.000	-7.650±0.295	0.327±0.042	1.859±0.117	0.239±0.001	-9.156±1.589
	Loc	0.129±0.011	0.711±0.223	0.216±0.006	-18.255±4.049	0.411±0.012	2.558±0.113	0.244±0.003	-7.506±0.156	0.404±0.010	2.461±0.220	0.244±0.003	-7.570±0.218
non-i.i.d (Label Skew)													
3	Fed	0.147±0.000	0.943±0.083	0.227±0.000	-23.485±6.252	0.432±0.001	2.561±0.029	0.249±0.000	-7.804±0.351	0.380±0.009	1.912±0.162	0.244±0.001	-7.959±0.265
	Loc	0.182±0.033	1.123±0.431	0.225±0.005	-14.413±2.433	0.408±0.019	2.428±0.212	0.246±0.001	-7.763±0.453	0.384±0.042	2.336±0.256	0.245±0.001	-8.272±1.048
5	Fed	0.148±0.001	0.988±0.058	0.227±0.000	-18.776±0.197	0.431±0.001	2.572±0.013	0.249±0.000	-7.394±0.021	0.388±0.004	2.055±0.024	0.244±0.000	-7.703±0.035
	Loc	0.180±0.034	1.136±0.345	0.224±0.006	-13.669±2.235	0.414±0.012	2.556±0.102	0.246±0.002	-7.506±0.162	0.399±0.024	2.450±0.174	0.245±0.002	-7.683±0.345
10	Fed	0.148±0.000	0.990±0.069	0.226±0.000	-18.928±1.312	0.405±0.019	2.228±0.233	0.247±0.001	-7.691±0.288	0.306±0.046	1.868±0.119	0.240±0.001	-9.178±0.996
	Loc	0.165±0.030	1.105±0.441	0.221±0.005	-15.744±3.444	0.412±0.013	2.670±0.093	0.245±0.004	-7.510±0.157	0.406±0.012	2.602±0.130	0.244±0.004	-7.555±0.234

Moreover, when considering the results with DP for the Bank dataset, we observe that the federated setting performs worse than the centralized baseline in several metrics. For instance, the federated setting with $\epsilon = 1.6$ performs worse than the centralized baseline for the PCD metric in 3 out of 9 scenarios, while in the others, the difference is not significant. This contrasts with the findings from the Adult dataset, where the federated setting performed better. This indicates that the distributed DP approach does not consistently outperform centralized DP. Its effectiveness is also related to the characteristics of the dataset. On the other hand, the local baseline results with DP exhibit worse or comparable results than the federated baseline with no clear trend.

Table 6.3 shows the results obtained with the Cardio dataset. Contrary to the previous two datasets, there are fewer differences across the scenarios between the baselines and federated learning. For the non-DP case, we observe that the federated setting outperforms the local baseline in the LC metric across all scenarios with a non-i.i.d. label skew data partition. This means the federated approach provides better similarity to the real data regarding clustering. Similarly, when applying DP, the federated setting outperforms the local baseline in 5 out of 9 scenarios for the LC metric, particularly in non-i.i.d. partitions. In other cases, the results are not significantly different.

Conversely, observing the other metrics, we can see that the federated setting with a privacy budget $\epsilon = 1.6$ outperforms the centralized baseline in the pMSE metric in 7 out of 9 scenarios by 0.01, which as previously stated is not representative for this metric.

Table 6.3: Fidelity results for the Cardio dataset with 3, 5, and 10 clients under different data partitions (i.i.d, non-i.i.d quantity skew, and non-i.i.d label skew) using FedBN. Statistical significance is highlighted as follows: green indicates significantly better performance than the baseline, and red indicates worse performance than the baseline. Federated results are compared to the centralized one, while local results are compared to the federated ones.

		Model BN											
		Non-DP				DP ($\epsilon = 0.5$)				DP ($\epsilon = 1.6$)			
NC	Bas.	M-HD ↓	PCD ↓	pMSE ↓	LC ↓	M-HD ↓	PCD ↓	pMSE ↓	LC ↓	M-HD ↓	PCD ↓	pMSE ↓	LC ↓
	Cen	0.212±0.001	0.251±0.032	0.242±0.000	-19.025±0.646	0.229±0.013	0.915±0.049	0.244±0.000	-12.215±0.514	0.219±0.007	0.710±0.044	0.243±0.000	-14.847±0.047
i.i.d													
3	Fed	0.212±0.001	0.232±0.028	0.242±0.000	-21.702±2.632	0.251±0.017	0.882±0.061	0.245±0.000	-12.948±2.302	0.219±0.004	0.643±0.090	0.242±0.000	-18.215±1.648
	Loc	0.209±0.003	0.269±0.019	0.239±0.003	-18.919±2.120	0.253±0.032	1.184±0.200	0.244±0.002	-10.174±2.491	0.232±0.023	1.045±0.289	0.242±0.003	-13.238±2.938
5	Fed	0.212±0.001	0.277±0.008	0.242±0.000	-18.701±1.973	0.275±0.045	1.220±0.260	0.245±0.001	-10.308±3.101	0.224±0.012	0.908±0.384	0.242±0.001	-15.975±4.282
	Loc	0.205±0.003	0.302±0.061	0.235±0.007	-19.010±2.963	0.279±0.032	1.216±0.175	0.244±0.005	-7.231±1.259	0.242±0.021	1.162±0.289	0.240±0.007	-11.227±3.678
10	Fed	0.212±0.001	0.279±0.003	0.242±0.000	-22.063±4.099	0.235±0.003	1.168±0.048	0.243±0.000	-13.533±0.953	0.218±0.005	0.765±0.340	0.242±0.000	-16.762±1.234
	Loc	0.203±0.004	0.341±0.121	0.231±0.008	-17.872±2.875	0.312±0.035	1.167±0.101	0.244±0.004	-5.623±0.693	0.269±0.033	1.240±0.189	0.239±0.008	-7.937±1.382
non-i.i.d (Quantity Skew)													
3	Fed	0.212±0.001	0.238±0.032	0.242±0.000	-17.379±1.248	0.262±0.004	1.123±0.133	0.245±0.000	-11.505±2.261	0.220±0.001	0.791±0.065	0.242±0.000	-17.431±1.393
	Loc	0.208±0.002	0.264±0.013	0.239±0.003	-18.751±1.978	0.276±0.037	1.217±0.164	0.245±0.004	-8.113±1.652	0.245±0.024	1.149±0.222	0.242±0.004	-10.696±1.892
5	Fed	0.212±0.001	0.278±0.002	0.242±0.000	-16.852±1.629	0.267±0.005	1.175±0.084	0.245±0.000	-10.663±1.485	0.220±0.001	0.771±0.061	0.242±0.000	-20.392±2.393
	Loc	0.206±0.004	0.318±0.069	0.234±0.007	-18.298±2.822	0.275±0.037	1.200±0.151	0.245±0.002	-7.260±1.367	0.241±0.026	1.145±0.268	0.241±0.004	-10.284±1.948
10	Fed	0.212±0.001	0.283±0.015	0.242±0.000	-17.268±1.038	0.239±0.007	1.027±0.070	0.244±0.000	-13.640±1.038	0.215±0.001	0.600±0.018	0.242±0.000	-19.213±3.012
	Loc	0.201±0.006	0.424±0.200	0.222±0.015	-16.421±2.722	0.318±0.054	1.189±0.122	0.244±0.005	-5.705±1.516	0.285±0.053	1.172±0.208	0.239±0.009	-7.585±2.850
non-i.i.d (Label Skew)													
3	Fed	0.212±0.001	0.303±0.031	0.242±0.000	-19.207±0.650	0.248±0.031	0.948±0.235	0.245±0.001	-9.773±1.857	0.217±0.007	0.653±0.222	0.243±0.000	-15.444±2.079
	Loc	0.234±0.019	0.453±0.165	0.237±0.004	-9.707±1.581	0.277±0.042	1.142±0.147	0.245±0.002	-7.556±1.734	0.253±0.030	1.065±0.204	0.242±0.002	-11.777±4.026
5	Fed	0.212±0.001	0.310±0.027	0.242±0.000	-21.373±2.303	0.263±0.004	1.253±0.052	0.245±0.000	-9.786±1.219	0.219±0.002	0.817±0.027	0.242±0.000	-16.412±1.264
	Loc	0.234±0.021	0.584±0.327	0.231±0.012	-10.483±3.298	0.297±0.031	1.218±0.185	0.243±0.006	-6.128±1.323	0.269±0.026	1.195±0.215	0.237±0.012	-8.336±2.477
10	Fed	0.212±0.001	0.317±0.023	0.242±0.000	-19.218±0.608	0.234±0.012	0.983±0.169	0.244±0.000	-12.489±1.259	0.214±0.001	0.643±0.093	0.242±0.000	-18.954±2.351
	Loc	0.233±0.020	0.553±0.284	0.226±0.013	-9.885±2.373	0.333±0.035	1.226±0.222	0.246±0.003	-5.375±1.295	0.295±0.042	1.219±0.229	0.241±0.006	-6.465±1.480

Furthermore, in some scenarios of the non-DP case, the local baseline outperforms the federated setting in the M-HD metric for the i.i.d and non-i.i.d quantity skew partitions. However, the remaining scenarios found no significant differences between the baselines and the federated setting. Also, we can conclude from this table that the impact of noise with different privacy budgets is subtle across all metrics.

Overall, the results demonstrate a clear trend: increasing the number of clients often leads to worse outcomes for the PCD metric. This may be due to the experimental setup, where having more clients results in less local data per client. Therefore, learning the correlations becomes more challenging since individual datasets contribute less to the global model. While this holds for the PCD metric, we can observe that the same conclusion does not hold for the LC metric, where it is difficult to say whether the number of clients significantly impacts the results. Moreover, when comparing scenarios with the same number of clients but different data partitions in the non-DP case, we find that, for all datasets, the non-i.i.d. label skew partition leads to worse results for the PCD metric.

Another interesting observation is that the fidelity metrics M-HD and pMSE show slight variation across different scenarios within each DP case in the federated setting for all datasets. This occurs for the M-HD metric because our BN approach uses privacy-preserving techniques to share counts in all scenarios during the parameter learning step. Consequently, neither the data partition nor the number of clients affects the inference process used to calculate the marginal distributions of variables.

On the other hand, for the pMSE metric, we observe that in most datasets and scenarios, results are close to 0.25, indicating that a CART model can distinguish which records

come from the synthetic data instead of the real data. These results suggest that some characteristics and structures are not well preserved with the BN model. We suspect that the observed values are mainly due to the numeric attributes in the datasets, which are uniformly binned during preprocessing and uniformly sampled during generation. Thus, some bins contain more values, even for highly skewed distributions, making it easier for a CART model to distinguish synthetic records based on these attributes. We confirm this observation by examining the CART model used to calculate the pMSE metric. Results highlight the following attributes as key decision nodes for distinguishing between synthetic and real data: capital-loss for the Adult dataset, balance for the Bank dataset, and ap_hi for the Cardio dataset. Note that all these attributes are highly skewed, as shown in Section 5.1. Furthermore, our findings align with previous work [PMS+23], which demonstrated that BN models lead to pMSE scores close to 0.25 in datasets with mixed attributes.

Finally, not surprisingly, if we compare the results for different privacy budgets, we can observe a direct impact on the fidelity results, with a lower privacy budget providing worse results. Bear in mind, however, that the impact of privacy budgets varies significantly for the datasets, with the Adult dataset showing the greatest differences in the fidelity metrics. It should be noted, though, that for all datasets, the values obtained for the metrics with a privacy budget $\epsilon = 1.6$ are still far from the reference values obtained without DP. We can conclude that balancing the privacy budget against the requirements of each application is beneficial to obtain a better fidelity-privacy trade-off.

6.1.2 Utility Evaluation

So far, we have seen synthetic data's performance in terms of fidelity. However, one common objective of generating synthetic data is to use it as a replacement in ML tasks where sensitive data cannot be used. In this section, we assess the effectiveness of synthetic data on binary classification tasks using the TRTR and TSTR methods described in Section 2.1.3. Furthermore, we report the ROC AUC scores obtained for the real and synthetic data in all the datasets and compare the results against the baselines (central and local) using statistical significance.

Table 6.4 shows the results obtained for the Adult dataset in terms of utility. Unlike the fidelity results, the differences between the centralized and federated settings and between the federated and local settings are not significant across all non-DP case metrics, indicating comparable performance. On the other hand, when analyzing the results with a privacy budget $\epsilon = 1.6$, we can spot significant differences in some of the scenarios between the federated and local settings. Especially for the KNN model, the federated setting outperforms the local results in 6 out of 9 cases whereas in other 3 scenarios the difference is not significant. Meanwhile, the NB model performs poorly with this privacy budget for all scenarios, giving results close to 0.5, indicating that the model performance is close to random guessing.

Another observation from the results is that we cannot tell, as in the case of fidelity, that

Table 6.4: Utility results for Adult dataset with 3, 5, and 10 clients under different data partitions (i.i.d, non-i.i.d quantity skew, and non-i.i.d label skew) using FedBN. Statistical significance is highlighted as follows: green indicates significantly better performance than the baseline, and red indicates worse performance than the baseline. Federated results are compared to the centralized one, while local results are compared to the federated ones.

Model BN										
NC	Ref	Non-DP			DP ($\epsilon = 0.5$)			DP ($\epsilon = 1.6$)		
		RF	NB	KNN	RF	NB	KNN	RF	NB	KNN
	Real	0.771±0.003	0.640±0.008	0.749±0.003	0.771±0.003	0.640±0.008	0.749±0.003	0.771±0.003	0.640±0.008	0.749±0.003
	Cen	0.724±0.043	0.696±0.015	0.712±0.026	0.685±0.031	0.508±0.003	0.629±0.019	0.705±0.014	0.515±0.002	0.657±0.024
i.i.d										
3	Fed	0.761±0.022	0.701±0.013	0.725±0.007	0.665±0.081	0.506±0.004	0.608±0.019	0.723±0.014	0.512±0.002	0.673±0.024
	Loc	0.752±0.031	0.699±0.013	0.717±0.013	0.604±0.028	0.503±0.003	0.565±0.047	0.646±0.037	0.506±0.004	0.648±0.035
5	Fed	0.761±0.014	0.701±0.004	0.720±0.004	0.652±0.035	0.501±0.001	0.619±0.015	0.726±0.017	0.537±0.017	0.699±0.005
	Loc	0.757±0.013	0.688±0.026	0.713±0.013	0.591±0.039	0.511±0.032	0.538±0.049	0.622±0.042	0.503±0.004	0.589±0.050
10	Fed	0.700±0.028	0.691±0.023	0.688±0.027	0.642±0.015	0.502±0.001	0.586±0.018	0.687±0.041	0.509±0.001	0.656±0.029
	Loc	0.719±0.039	0.679±0.045	0.677±0.028	0.591±0.052	0.544±0.050	0.526±0.050	0.621±0.055	0.505±0.010	0.549±0.041
non-i.i.d (Quantity Skew)										
3	Fed	0.752±0.030	0.698±0.009	0.725±0.004	0.674±0.026	0.500±0.000	0.597±0.036	0.732±0.022	0.521±0.011	0.690±0.011
	Loc	0.763±0.019	0.698±0.017	0.721±0.011	0.636±0.048	0.503±0.004	0.567±0.063	0.653±0.052	0.504±0.004	0.624±0.036
5	Fed	0.758±0.015	0.697±0.010	0.717±0.009	0.630±0.036	0.504±0.003	0.636±0.006	0.716±0.024	0.522±0.008	0.690±0.005
	Loc	0.731±0.037	0.701±0.027	0.691±0.026	0.619±0.058	0.530±0.059	0.540±0.036	0.643±0.046	0.532±0.059	0.575±0.050
10	Fed	0.704±0.013	0.704±0.014	0.679±0.023	0.654±0.032	0.502±0.002	0.568±0.029	0.667±0.059	0.510±0.001	0.665±0.025
	Loc	0.712±0.035	0.684±0.037	0.677±0.030	0.572±0.056	0.514±0.056	0.523±0.048	0.595±0.062	0.511±0.055	0.541±0.050
non-i.i.d (Label Skew)										
3	Fed	0.727±0.054	0.714±0.008	0.716±0.015	0.627±0.055	0.517±0.020	0.582±0.022	0.733±0.008	0.510±0.007	0.688±0.008
	Loc	0.641±0.136	0.623±0.114	0.623±0.120	0.605±0.104	0.517±0.035	0.572±0.080	0.617±0.116	0.545±0.075	0.586±0.096
5	Fed	0.718±0.043	0.724±0.022	0.687±0.040	0.681±0.052	0.506±0.003	0.624±0.037	0.734±0.021	0.513±0.010	0.674±0.033
	Loc	0.649±0.134	0.642±0.123	0.628±0.118	0.585±0.108	0.548±0.068	0.552±0.064	0.618±0.112	0.566±0.087	0.586±0.095
10	Fed	0.683±0.066	0.724±0.012	0.688±0.036	0.575±0.058	0.501±0.001	0.586±0.016	0.664±0.034	0.506±0.004	0.671±0.026
	Loc	0.596±0.107	0.604±0.104	0.585±0.094	0.539±0.059	0.524±0.078	0.504±0.037	0.553±0.069	0.534±0.066	0.529±0.037

the performance for the same number of clients but different data partitions is always better for the i.i.d setting. In some algorithms, the non-i.i.d label skew partition results even outperform those for i.i.d scenarios. For example, if we compare the results for 5 clients in the NB algorithm without DP, we can see that the results are better than those in the other data partitions with the same number of clients.

Turning our attention to the ROC AUC scores on the real data, we observe that the synthetic data in the federated setting achieves scores close to the real data for the RF and KNN algorithms, with less than a 3% difference. In some scenarios, the synthetic data even outperforms the real data results for the NB algorithm. This indicates that the quality of the synthetic data generated without DP is good in terms of utility.

Similar conclusions to those observed in the Adult dataset for the non-DP scenarios are also evident in the utility results of the Bank dataset. On the other hand, for the results with privacy budgets ($\epsilon = 1.6$ and $\epsilon = 0.5$), most of the scenarios show ROC AUC scores close to 0.5 for the different scenarios, meaning that the noise introduced directly impacts the utility results, providing poor performance in the binary task.

Table 6.5 shows the results for the cardio dataset. Interestingly, these results show that the impact of DP on utility is negligible in all scenarios considered. Furthermore, the most significant differences between the baselines and the federated setting are observed

in the non-i.i.d label skew scenarios for the results with $\epsilon = 0.5$ and $\epsilon = 1.6$. In particular, for the scenario with 10 clients, we can observe that the federated setting outperforms the local setting for all the algorithms.

Table 6.5: Utility results for the Cardio dataset with 3, 5, and 10 clients under different data partitions (i.i.d, non-i.i.d quantity skew, and non-i.i.d label skew) using FedBN. Statistical significance is highlighted as follows: green indicates significantly better performance than the baseline, and red indicates worse performance than the baseline. Federated results are compared to the centralized one, while local results are compared to the federated ones.

Model BN										
NC	Bas.	Non-DP			DP ($\epsilon = 0.5$)			DP ($\epsilon = 1.6$)		
		RF	NB	KNN	RF	NB	KNN	RF	NB	KNN
	Real	0.715±0.003	0.589±0.004	0.652±0.006	0.715±0.003	0.589±0.004	0.652±0.006	0.715±0.003	0.589±0.004	0.652±0.006
	Cen	0.627±0.006	0.613±0.003	0.597±0.008	0.619±0.004	0.598±0.007	0.591±0.004	0.626±0.002	0.604±0.006	0.592±0.004
i.i.d										
3	Fed	0.630±0.001	0.614±0.003	0.597±0.005	0.621±0.007	0.607±0.014	0.591±0.007	0.630±0.004	0.605±0.008	0.600±0.006
	Loc	0.626±0.003	0.608±0.003	0.596±0.005	0.597±0.013	0.590±0.017	0.580±0.007	0.621±0.006	0.597±0.007	0.592±0.007
5	Fed	0.633±0.002	0.608±0.002	0.596±0.002	0.617±0.006	0.601±0.009	0.592±0.003	0.628±0.001	0.599±0.004	0.593±0.003
	Loc	0.632±0.013	0.609±0.006	0.600±0.011	0.587±0.027	0.583±0.025	0.565±0.016	0.617±0.021	0.596±0.009	0.587±0.009
10	Fed	0.624±0.001	0.605±0.001	0.592±0.006	0.623±0.006	0.591±0.000	0.592±0.003	0.628±0.005	0.596±0.005	0.596±0.005
	Loc	0.640±0.032	0.616±0.022	0.607±0.029	0.578±0.045	0.570±0.049	0.559±0.033	0.611±0.038	0.603±0.019	0.587±0.031
non-i.i.d (Quantity Skew)										
3	Fed	0.624±0.007	0.609±0.006	0.591±0.011	0.620±0.005	0.602±0.011	0.594±0.004	0.624±0.003	0.605±0.002	0.597±0.001
	Loc	0.628±0.004	0.607±0.004	0.596±0.005	0.597±0.018	0.588±0.012	0.580±0.012	0.620±0.007	0.597±0.011	0.591±0.004
5	Fed	0.627±0.002	0.605±0.003	0.600±0.003	0.620±0.006	0.589±0.000	0.591±0.003	0.628±0.005	0.598±0.001	0.592±0.002
	Loc	0.638±0.025	0.612±0.009	0.602±0.018	0.587±0.040	0.567±0.037	0.568±0.022	0.615±0.034	0.596±0.011	0.589±0.017
10	Fed	0.625±0.002	0.608±0.003	0.592±0.003	0.625±0.003	0.592±0.002	0.592±0.003	0.624±0.002	0.603±0.005	0.595±0.002
	Loc	0.642±0.038	0.616±0.025	0.607±0.034	0.573±0.047	0.560±0.047	0.556±0.040	0.601±0.047	0.581±0.033	0.573±0.039
non-i.i.d (Label Skew)										
3	Fed	0.619±0.010	0.614±0.009	0.580±0.006	0.605±0.013	0.610±0.013	0.580±0.019	0.621±0.008	0.615±0.012	0.583±0.015
	Loc	0.552±0.056	0.567±0.048	0.544±0.043	0.537±0.036	0.531±0.036	0.530±0.028	0.548±0.048	0.552±0.044	0.535±0.033
5	Fed	0.606±0.006	0.608±0.014	0.573±0.011	0.606±0.007	0.592±0.009	0.575±0.011	0.609±0.008	0.601±0.013	0.581±0.011
	Loc	0.555±0.055	0.561±0.055	0.544±0.044	0.543±0.048	0.526±0.038	0.528±0.028	0.549±0.050	0.531±0.043	0.534±0.034
10	Fed	0.614±0.009	0.612±0.013	0.582±0.015	0.612±0.010	0.606±0.020	0.579±0.005	0.614±0.005	0.606±0.015	0.580±0.008
	Loc	0.550±0.058	0.551±0.051	0.539±0.043	0.530±0.036	0.518±0.035	0.520±0.024	0.537±0.045	0.528±0.039	0.526±0.030

Overall, we can conclude that the impact of noise added in DP scenarios varies depending on the ML algorithm used and the characteristics of the datasets. In particular, we observe that the noise does not impact the utility results for the Cardio dataset. A possible explanation might be that the PCD between the real and synthetic data remains small (below or close to 1.0) with the chosen privacy budgets. Hence, the ML models can still exploit the correlations between features to make a prediction close to the one in the real data. Similarly, the data partition and the number of clients demonstrate no significant impact on the utility results.

6.1.3 Privacy Evaluation

Although fidelity and utility are relevant for evaluating the usefulness of synthetic data, it is also crucial for applications dealing with sensitive data to analyze whether the synthetic data discloses information from real data. This section evaluates synthetic data privacy using two metrics: DCR and AD (Section 2.1.3). For the first metric, we considered the 27 scenarios corresponding to federated settings reported in previous sections. For each

dataset, we report the following metrics: the minimum distance to the closest record (Min DCR), the average distance to the training data (Avg DCR Train), the average distance to holdout data (Avg DCR Holdout), and the proportion of synthetic records (Share) closer to the training data than to the holdout data. We report the minimum DCR to observe whether there are exact matches (i.e., a synthetic record that is identical to a real record). A DCR value of 0 indicates there is an exact match. It is worth noting that exact matches can occur by chance and do not necessarily imply a disclosure risk.

Table 6.6 shows the results obtained for the Adult dataset regarding the DCR metric. The generated synthetic data in the different federated settings exhibits almost identical DCR distributions for the training and holdout records with share values close to 0.5. This provides empirical evidence that the synthetic data can generalize the patterns in the real data. Furthermore, the minimum DCR is greater than 0 across all scenarios, meaning no exact matches exist. Moreover, the average distances to the training and holdout data increase when differential privacy (DP) is applied during the synthetic data generation process. Results for the remaining datasets provide similar insights.

Table 6.6: DCR results of the holdout assessment for the Adult dataset with 3, 5 and 10 clients under different data partitions (i.i.d and non-i.i.d label skew) using FedBN.

Model BN												
Clients	Non-DP				DP ($\epsilon = 0.5$)				DP ($\epsilon = 1.6$)			
	Min DCR	Share	Avg DCR Train	Avg DCR Holdout	Min DCR	Share	Avg DCR Train	Avg DCR Holdout	Min DCR	Share	Avg DCR Train	Avg DCR Holdout
i.i.d												
3	0.002±0.001	0.477±0.024	0.699±0.009	0.686±0.002	0.054±0.017	0.468±0.008	4.278±0.034	4.232±0.021	0.005±0.002	0.466±0.007	3.538±0.308	3.485±0.307
5	0.003±0.001	0.476±0.024	0.705±0.008	0.694±0.003	0.031±0.005	0.462±0.009	4.177±0.052	4.117±0.060	0.004±0.003	0.466±0.011	2.931±0.074	2.882±0.081
10	0.002±0.001	0.477±0.024	0.717±0.010	0.704±0.005	0.027±0.009	0.465±0.010	4.267±0.033	4.219±0.025	0.005±0.003	0.469±0.008	3.525±0.200	3.476±0.208
non-i.i.d (Quantity Skew)												
3	0.002±0.001	0.476±0.024	0.702±0.008	0.689±0.001	0.032±0.015	0.463±0.006	4.242±0.041	4.186±0.047	0.005±0.000	0.465±0.006	3.281±0.082	3.223±0.084
5	0.001±0.001	0.481±0.024	0.704±0.004	0.691±0.005	0.046±0.033	0.464±0.007	4.223±0.056	4.171±0.064	0.004±0.000	0.467±0.007	3.231±0.393	3.183±0.403
10	0.002±0.001	0.482±0.028	0.707±0.011	0.699±0.006	0.040±0.019	0.466±0.009	4.285±0.025	4.234±0.028	0.005±0.001	0.464±0.009	3.495±0.153	3.444±0.162
non-i.i.d (Label Skew)												
3	0.002±0.000	0.481±0.022	0.695±0.002	0.688±0.002	0.066±0.023	0.468±0.009	4.290±0.043	4.240±0.039	0.006±0.002	0.465±0.003	3.545±0.296	3.493±0.296
5	0.002±0.001	0.477±0.023	0.707±0.008	0.696±0.005	0.034±0.013	0.467±0.007	4.248±0.078	4.196±0.085	0.009±0.005	0.467±0.007	3.411±0.456	3.364±0.464
10	0.002±0.001	0.482±0.023	0.710±0.008	0.698±0.004	0.028±0.002	0.468±0.006	4.274±0.035	4.226±0.038	0.004±0.001	0.463±0.012	3.598±0.168	3.548±0.178

On the other hand, for the AD metric, we analyze one attacker scenario per dataset as described in Section 5.6.1 and select a subset of the 27 federated settings considered in previous experiments. Specifically, we report settings with 10 clients, and i.i.d and non-i.i.d label skew data partitions. The DP cases remain the same.

Table 6.7 shows the average attribute disclosure risk for all real records in the attack scenarios considered. It includes accuracy scores obtained for the real and synthetic data. The real data results serve as a point of comparison to see how well an attacker can infer a sensitive attribute with access to the complete information [HME20]. The columns in each row represent the following ML algorithms: Random Forest (RF), Support Vector Machine (SVM), Naive Bayes (NB), K-Nearest Neighbors (KNN), Logistic Regression (LR), an Ensemble algorithm (ENS), and the Dummy Classifier (DUM). The ENS refers to an algorithm that combines all other algorithms' predictions using majority voting. We include the ENS to simulate a stronger attacker that leverages multiple classifiers to infer the sensitive attribute. The DUM classifier predicts the most frequent value in real and synthetic data. Additionally, each row includes the mean results of all algorithms

listed (RMEAN), excluding the DUM algorithm.

We can make the following observations from Table 6.7. For the Adult dataset, the results of the non-DP scenarios using an ENS classifier on the synthetic data show scores close to the real data for both data partitions, with only a 0.006 difference. Similar results are observed for the Bank dataset, with a difference of around 0.002. It is also evident that smaller privacy budgets lead to smaller accuracy scores in all the algorithms, reducing the risk of disclosure. Nevertheless, it should be noted that in the Adult and Bank datasets, the accuracy scores for the given privacy budgets are still close to those in the real data. The risk reduction concerning the ENS model with a privacy budget $\epsilon = 0.5$ is 0.046, and the RMEAN still beats the DUM classifier by 0.066, meaning that the attacker can still exploit the structure of the synthetic data. Unfortunately, this is a consequence of the close relationship between attribute disclosure risk and data utility, which has been discussed in several works [HME20, TEPS18]. On the other hand, the results obtained for the cardio dataset show that the DUM classifier provides the same results as the ENS algorithm used by the attacker, and the real data also provides similar results to the dummy classifier for all algorithms. Therefore, in this case, the algorithms do not provide extra information to the attacker in any scenarios considered.

Table 6.7: Attribute Disclosure results for the different datasets with 3, 5 and 10 clients under different data partitions (i.i.d and non-i.i.d label skew) using FedBN.

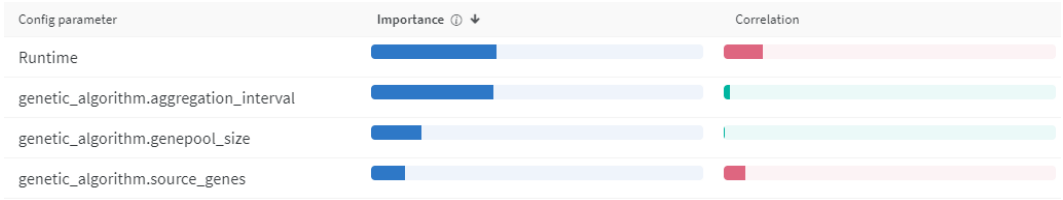
Scenario	Baseline	Model BN							
		RF	SVM	NB	KNN	LR	ENS	DUM	RMEAN
Scenario 1 (Adult)	Real	0.648±0.053	0.648±0.049	0.618±0.049	0.592±0.067	0.629±0.049	0.636±0.050	0.458	0.626±0.007
	i.i.d								
	DP ($\epsilon = 0.5$)	0.460±0.077	0.595±0.045	0.592±0.048	0.347±0.016	0.559±0.050	0.590±0.048	0.458	0.524±0.021
	DP ($\epsilon = 1.6$)	0.596±0.046	0.610±0.054	0.609±0.037	0.514±0.053	0.603±0.048	0.613±0.049	0.458	0.591±0.008
	Non-DP	0.624±0.047	0.630±0.047	0.621±0.047	0.579±0.053	0.627±0.047	0.630±0.049	0.458	0.619±0.002
	non i.i.d (Label Skew)								
	DP ($\epsilon = 0.5$)	0.465±0.085	0.615±0.050	0.607±0.051	0.341±0.047	0.581±0.037	0.605±0.052	0.458	0.535±0.020
	DP ($\epsilon = 1.6$)	0.595±0.049	0.620±0.043	0.613±0.036	0.491±0.072	0.610±0.036	0.620±0.042	0.458	0.592±0.014
	Non-DP	0.623±0.047	0.631±0.048	0.622±0.047	0.574±0.060	0.627±0.048	0.630±0.049	0.458	0.618±0.007
Scenario 2 (Bank)	Real	0.642±0.016	0.618±0.005	0.603±0.019	0.594±0.021	0.595±0.020	0.616±0.020	0.554	0.611±0.006
	i.i.d								
	DP ($\epsilon = 0.5$)	0.555±0.016	0.586±0.006	0.572±0.011	0.532±0.015	0.563±0.010	0.579±0.007	0.518	0.565±0.006
	DP ($\epsilon = 1.6$)	0.600±0.006	0.590±0.007	0.571±0.011	0.560±0.016	0.563±0.019	0.586±0.006	0.518	0.578±0.006
	Non-DP	0.617±0.006	0.615±0.005	0.599±0.022	0.583±0.009	0.593±0.025	0.614±0.011	0.554	0.603±0.009
	non i.i.d (Label Skew)								
	DP ($\epsilon = 0.5$)	0.556±0.008	0.583±0.005	0.568±0.013	0.525±0.012	0.570±0.013	0.580±0.005	0.518	0.564±0.007
	DP ($\epsilon = 1.6$)	0.603±0.008	0.600±0.006	0.569±0.018	0.557±0.019	0.566±0.024	0.589±0.009	0.518	0.581±0.009
	Non-DP	0.619±0.006	0.616±0.005	0.599±0.024	0.587±0.010	0.595±0.022	0.615±0.010	0.554	0.605±0.008
Scenario 3 (Cardio)	Real	0.820±0.084	0.748±0.000	0.748±0.073	0.747±0.015	0.748±0.000	0.748±0.000	0.748	0.760±0.034
	i.i.d								
	DP ($\epsilon = 0.5$)	0.679±0.053	0.748±0.000	0.748±0.000	0.715±0.013	0.748±0.000	0.748±0.000	0.748	0.731±0.022
	DP ($\epsilon = 1.6$)	0.677±0.056	0.748±0.000	0.748±0.000	0.727±0.006	0.748±0.000	0.748±0.000	0.748	0.733±0.023
	Non-DP	0.679±0.055	0.748±0.000	0.748±0.000	0.724±0.003	0.748±0.000	0.748±0.000	0.748	0.733±0.022
	non i.i.d (Label Skew)								
	DP ($\epsilon = 0.5$)	0.678±0.055	0.748±0.000	0.748±0.000	0.725±0.006	0.748±0.000	0.748±0.000	0.748	0.733±0.022
	DP ($\epsilon = 1.6$)	0.679±0.055	0.748±0.000	0.748±0.000	0.724±0.012	0.748±0.000	0.748±0.000	0.748	0.732±0.022
	Non-DP	0.677±0.056	0.748±0.000	0.748±0.000	0.716±0.008	0.748±0.000	0.748±0.000	0.748	0.731±0.023

6.1.4 Sensitivity Analysis of Hyperparameters

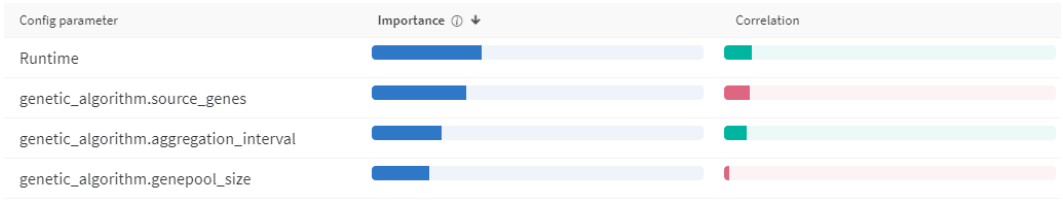
In this section, we investigate the impact of hyperparameters on the quality of the synthetic data generated using FedBN. For this analysis, we select the same scenario

used for hyperparameter tuning (5 clients with a non-i.i.d label skew partition). However, instead of finding the optimal configuration, we concentrate on the importance and correlation of hyperparameters concerning the LC fidelity metric computed with respect to the real data for three random splits. The importance and correlation values are obtained using Weight & Biases. The former values are obtained by training a random forest model using all hyperparameters and the runtime as features, with the mean LC metric as the target, and then computing the feature importance. The latter values correspond to the linear correlation between the hyperparameter and the LC metric.

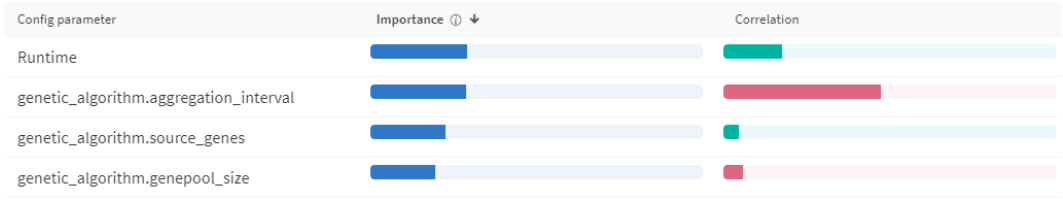
Figure 6.1 shows the results obtained for all datasets. The hyperparameter with the highest importance in two of the datasets is the aggregation interval, showing a slight positive correlation with the LC metric in the Adult dataset and a strong negative correlation in the Cardio dataset. This implies that depending on the dataset, the frequency at which clients share their best individuals (i.e., best BN structures in the GA algorithm) with the server directly impacts the fidelity results. From this analysis, we can conclude that the importance of the order of hyperparameters and correlations is highly dependent on the characteristics of the datasets. However, the aggregation interval shows high importance across most analyzed experiments.



(a) Adult Dataset



(b) Bank Dataset



(c) Cardio Dataset

Figure 6.1: Sensitivity analysis of hyperparameters on synthetic data fidelity using the Log Cluster Metric in a federated setting with 5 clients and a non-i.i.d. label skew partition across all datasets: (a) Adult, (b) Bank, and (c) Cardio, using FedBN.

6.2 Federated Variational Autoencoder

Here, we present the results for the FedVAE model described in Section 4.3. The results cover fidelity, utility, and privacy metrics and follow the same structure as those reported with the FedBN model. Note that differences between models may impact the computation of results. Therefore, we clarify these differences in the relevant sections.

6.2.1 Fidelity Evaluation

The tables here follow the same structure explained in Section 6.1. Nevertheless, there are some differences in how the privacy budget is interpreted for DP settings. Note that DP guarantees for the VAE model are obtained using DP-SGD, a commonly used method for training neural networks with DP. In particular, we employ local DP in the federated setting following the approach presented in previous work [NHDC22], where each client applies the noise locally using DP-SGD before sharing the parameters with the server. In this context, the privacy budgets corresponding to the federated setting ($\epsilon = 0.5$ or $\epsilon = 1.6$) indicate the DP guarantees satisfied by the models of each client.

Furthermore, the number of epochs used in the experiments with DP differs from the values reported in the hyperparameter tuning section. This is mainly because we observe when running the experiments in the federated setting that the model fails to converge with the same number of global epochs and that increasing this value can improve the results. Nonetheless, it also increases the computational cost of the experiments. Taking this trade-off into account, we set the number of global epochs for the DP settings as follows: 300 epochs for the Adult dataset, 300 epochs for the Bank dataset, and 100 epochs for the Cardio dataset. It is important to note that the number of local epochs and other hyperparameters remained unchanged.

Table 6.8 shows the results for the Adult dataset. We can observe that for the non-DP case, the centralized setting outperforms the federated setting in all the scenarios in the PCD metric. In particular, we note more significant differences in the scenarios with a non-i.i.d label skew data partition. This indicates that preserving correlations is more challenging for the FedVAE model when label distributions differ among clients. Furthermore, when examining the results across the other metrics, we observe that the federated setting outperforms the local setting in the LC, pMSE, and M-HD metrics in two scenarios each. At the same time, it performs worse than the centralized setting in the same metrics in exactly one scenario each. The remaining scenarios show comparable results between the baselines.

The results for a privacy budget of $\epsilon = 1.6$ demonstrate that the federated setting outperforms the centralized setting in the PCD metric for 3 scenarios. However, the federated setting performs worse than the centralized setting in the pMSE metric in 4 out of 9 scenarios. Similarly, the results with an $\epsilon = 0.5$ show that the federated setting consistently outperforms the centralized setting in most scenarios for the PCD metric, and the local setting performs worse than the centralized setting for the M-HD and PCD

in several scenarios. Additionally, if we compare the metrics results for both privacy budgets, we observe that the results are sometimes worse for a higher epsilon. At first glance, these results seem counterintuitive. However, we suspect this discrepancy could be attributed to the low quality of the latent representations caused by the introduction of noise during the training process. This leads to significant fluctuations in outcomes and generally indicates poor performance.

Table 6.8: Fidelity results for the Adult Dataset for 3, 5, and 10 clients under different data partitions (i.i.d, non-i.i.d quantity skew, and non-i.i.d label skew) using FedVAE. Statistical significance is highlighted as follows: green indicates significantly better performance than the baseline, and red indicates worse performance than the baseline. Federated results are compared to the centralized one, while local results are compared to the federated ones.

		Model VAE											
		Non-DP				DP ($\epsilon = 0.5$)				DP ($\epsilon = 1.6$)			
NC	Bas.	M-HD a	PCD a	pMSE a	LC a	M-HD a	PCD a	pMSE a	LC a	M-HD a	PCD a	pMSE a	LC a
	Cen	0.150±0.018	1.014±0.031	0.108±0.017	-11.391±1.054	0.309±0.004	2.525±0.145	0.178±0.006	-7.671±1.286	0.344±0.032	2.673±0.306	0.191±0.013	-6.955±2.798
		i.i.d											
3	Fed	0.193±0.036	1.693±0.102	0.160±0.033	-10.962±2.953	0.267±0.030	2.046±0.151	0.193±0.005	-10.952±4.738	0.405±0.013	2.142±0.156	0.224±0.009	-5.355±0.482
	Loc	0.172±0.019	1.220±0.093	0.125±0.012	-10.988±3.668	0.348±0.012	2.991±0.072	0.197±0.008	-4.869±0.831	0.368±0.006	3.181±0.073	0.195±0.006	-6.434±1.501
5	Fed	0.180±0.009	1.514±0.059	0.159±0.016	-15.956±3.931	0.267±0.024	2.054±0.081	0.184±0.015	-7.698±0.970	0.253±0.019	2.172±0.044	0.171±0.003	-7.752±2.924
	Loc	0.196±0.022	1.330±0.153	0.148±0.015	-8.640±1.533	0.351±0.029	2.238±0.124	0.209±0.019	-7.015±2.038	0.397±0.013	3.218±0.085	0.217±0.011	-4.693±2.017
10	Fed	0.168±0.012	1.472±0.045	0.128±0.017	-14.390±2.368	0.442±0.006	2.186±0.158	0.232±0.011	-3.149±0.093	0.401±0.018	2.967±0.174	0.227±0.009	-3.553±0.609
	Loc	0.247±0.018	2.243±0.208	0.170±0.006	-7.201±3.173	0.478±0.041	2.157±0.107	0.243±0.006	-3.210±0.326	0.423±0.035	2.184±0.091	0.232±0.009	-4.357±1.847
		non-i.i.d (Quantity Skew)											
3	Fed	0.160±0.035	1.565±0.036	0.112±0.016	-10.996±4.939	0.232±0.022	2.027±0.119	0.174±0.018	-5.880±0.586	0.382±0.006	1.980±0.177	0.214±0.007	-5.442±2.127
	Loc	0.175±0.027	1.131±0.147	0.132±0.016	-12.150±2.232	0.356±0.021	2.926±0.177	0.196±0.011	-7.486±2.355	0.374±0.007	3.149±0.105	0.200±0.006	-5.349±0.898
5	Fed	0.159±0.002	1.470±0.044	0.111±0.004	-7.787±2.832	0.258±0.014	2.004±0.045	0.186±0.015	-11.776±3.609	0.387±0.017	2.302±0.205	0.217±0.009	-3.151±0.217
	Loc	0.201±0.026	1.537±0.229	0.148±0.024	-8.782±3.745	0.391±0.042	2.925±0.363	0.213±0.018	-4.118±1.173	0.388±0.023	3.232±0.150	0.212±0.015	-5.820±2.049
10	Fed	0.150±0.002	1.413±0.028	0.107±0.011	-16.594±4.803	0.301±0.010	2.618±0.101	0.182±0.000	-8.801±0.969	0.359±0.016	2.802±0.154	0.192±0.004	-5.894±1.991
	Loc	0.270±0.087	2.076±0.759	0.170±0.030	-7.492±3.499	0.448±0.064	2.670±0.494	0.230±0.018	-5.612±3.942	0.436±0.051	2.866±0.527	0.224±0.019	-4.608±2.588
		non-i.i.d (Label Skew)											
3	Fed	0.203±0.036	1.862±0.246	0.162±0.058	-8.766±1.218	0.239±0.003	1.893±0.065	0.185±0.011	-10.795±2.943	0.388±0.018	2.039±0.246	0.219±0.011	-3.339±0.252
	Loc	0.218±0.034	1.684±0.316	0.143±0.026	-11.915±7.521	0.367±0.040	3.101±0.337	0.197±0.012	-4.564±1.366	0.371±0.035	3.183±0.196	0.199±0.008	-8.123±3.341
5	Fed	0.187±0.009	2.118±0.154	0.170±0.052	-10.059±4.225	0.274±0.032	2.133±0.074	0.182±0.020	-9.571±1.456	0.347±0.012	2.304±0.155	0.192±0.002	-6.899±2.558
	Loc	0.262±0.085	1.937±0.492	0.159±0.036	-8.179±3.031	0.398±0.065	2.929±±0.383	0.215±0.017	-9.238±3.756	0.406±0.053	3.159±0.395	0.212±0.022	-5.642±2.388
10	Fed	0.191±0.031	2.107±0.164	0.159±0.060	-11.367±4.796	0.339±0.015	2.763±0.094	0.185±0.008	-4.57±1.552	0.360±0.017	3.016±0.059	0.192±0.013	-7.679±0.683
	Loc	0.304±0.086	2.305±0.280	0.179±0.032	-7.007±2.960	0.427±0.070	2.772±0.460	0.228±0.015	-5.007±2.383	0.426±0.040	3.170±0.414	0.222±0.013	-5.419±2.450

Table 6.9 exhibits the results for the Bank dataset. Different insights to those observed in the Adult dataset are noted here. In particular, for the non-DP case, no significant differences were found between the centralized and the federated baseline for the experiments with i.i.d and non-i.i.d quantity skew data partitions, excluding the scenario with 10 clients for the non-i.i.d case were the federated setting outperformed the centralized setting. In contrast, for the non-i.i.d label skew scenarios, the centralized setting outperformed the federated setting in all cases. Furthermore, for scenarios with a privacy budget $\epsilon = 1.6$ the centralized setting consistently outperforms the federated setting in all M-HD, PCD, and pMSE metrics in all the scenarios. Likewise, for a privacy budget $\epsilon = 0.5$ the centralized setting outperforms the federated setting in 6 out of 9 scenarios for the M-HD and the pMSE metrics. Fluctuations in the metrics for the DP results are also observed in this dataset. Similar observations are obtained for the Cardio dataset. Except in the non-DP case were no significant differences is evident in the baselines compared.

Overall, we conclude that the heterogeneity in data distribution among clients in the federated setting directly impacts the performance of the FedVAE model, particularly in terms of the PCD metric, showing worse results for the non-i.i.d. label skew data

Table 6.9: Fidelity results for the Bank Dataset with 3, 5, and 10 clients under different data partitions (i.i.d, non-i.i.d quantity skew, and non-i.i.d label skew) using FedVAE. Statistical significance is highlighted as follows: green indicates significantly better performance than the baseline, and red indicates worse performance than the baseline. Federated results are compared to the centralized one, while local results are compared to the federated ones.

Model VAE													
		Non-DP				DP ($\epsilon = 0.5$)				DP ($\epsilon = 1.6$)			
NC	Bas.	M-HD a	PCD a	pMSE a	LC a	M-HD a	PCD a	pMSE a	LC a	M-HD a	PCD a	pMSE a	LC a
	Cen	0.056±0.006	1.014±0.053	0.032±0.003	-12.927±1.548	0.185±0.008	2.408±0.100	0.118±0.004	-9.063±1.938	0.088±0.002	1.502±0.056	0.063±0.002	-12.475±1.793
i.i.d													
3	Fed	0.061±0.004	1.055±0.025	0.025±0.001	-14.650±3.268	0.302±0.009	2.352±0.036	0.220±0.005	-12.266±1.923	0.227±0.014	2.168±0.031	0.185±0.017	-9.183±3.139
	Loc	0.075±0.007	0.813±0.046	0.042±0.004	-12.835±1.726	0.211±0.005	2.528±0.074	0.144±0.007	-7.926±2.451	0.192±0.006	2.483±0.075	0.115±0.007	-8.120±1.411
5	Fed	0.054±0.004	1.073±0.054	0.027±0.005	-15.751±0.726	0.198±0.017	2.377±0.055	0.144±0.022	-5.975±1.105	0.207±0.012	2.427±0.103	0.132±0.013	-6.634±0.868
	Loc	0.072±0.007	0.798±0.081	0.043±0.003	-12.206±2.496	0.221±0.007	2.495±0.025	0.151±0.013	-7.415±1.844	0.228±0.009	2.675±0.090	0.149±0.009	-6.560±1.400
10	Fed	0.056±0.004	0.953±0.088	0.038±0.008	-11.526±1.726	0.210±0.001	2.357±0.041	0.160±0.009	-10.425±3.522	0.220±0.015	2.532±0.034	0.145±0.022	-6.896±1.780
	Loc	0.076±0.007	0.753±0.076	0.043±0.004	-13.250±2.550	0.218±0.016	2.350±0.035	0.171±0.019	-8.326±2.976	0.231±0.010	2.536±0.078	0.158±0.011	-8.628±3.780
non-i.i.d (Quantity Skew)													
3	Fed	0.056±0.002	1.003±0.043	0.025±0.001	-12.386±1.458	0.273±0.005	2.280±0.102	0.198±0.018	-9.251±0.420	0.237±0.006	2.188±0.046	0.160±0.019	-9.965±2.824
	Loc	0.070±0.010	0.812±0.029	0.041±0.006	-13.314±1.345	0.206±0.017	2.519±0.062	0.134±0.014	-9.589±2.946	0.191±0.025	2.371±0.294	0.119±0.018	-7.750±1.415
5	Fed	0.053±0.002	0.964±0.034	0.026±0.004	-17.435±5.049	0.263±0.021	2.318±0.039	0.213±0.009	-9.500±1.404	0.239±0.007	2.135±0.083	0.173±0.020	-8.878±1.360
	Loc	0.080±0.006	0.835±0.105	0.044±0.003	-13.435±1.776	0.217±0.019	2.475±0.051	0.146±0.013	-8.453±2.737	0.236±0.038	2.789±0.328	0.152±0.032	-5.839±1.250
10	Fed	0.049±0.004	0.899±0.016	0.024±0.002	-16.747±2.952	0.199±0.005	2.411±0.054	0.173±0.025	-9.777±3.152	0.197±0.018	2.557±0.135	0.127±0.008	-7.658±1.362
	Loc	0.099±0.036	1.001±0.342	0.061±0.025	-12.861±2.562	0.255±0.046	2.425±0.156	0.186±0.035	-8.758±3.567	0.247±0.026	2.606±0.273	0.180±0.026	-6.754±1.649
non-i.i.d (Label Skew)													
3	Fed	0.075±0.002	1.622±0.016	0.036±0.005	-13.765±3.108	0.305±0.020	2.311±0.027	0.215±0.001	-9.397±6.104	0.298±0.032	2.251±0.008	0.200±0.005	-10.171±2.332
	Loc	0.110±0.040	1.208±0.305	0.085±0.049	-12.925±2.111	0.228±0.041	2.528±0.075	0.167±0.043	-9.356±1.540	0.224±0.068	2.614±0.361	0.153±0.060	-10.282±2.799
5	Fed	0.070±0.004	1.636±0.011	0.038±0.003	-13.182±3.086	0.261±0.018	2.338±0.038	0.209±0.026	-8.225±1.972	0.257±0.029	2.223±0.030	0.191±0.015	-7.006±1.822
	Loc	0.117±0.047	1.222±0.327	0.098±0.057	-13.583±2.552	0.245±0.052	2.485±0.144	0.171±0.044	-8.141±2.033	0.239±0.055	2.646±0.223	0.162±0.050	-8.988±1.684
10	Fed	0.067±0.004	1.631±0.012	0.036±0.001	-13.302±1.807	0.186±0.010	2.414±0.039	0.127±0.008	-8.985±2.084	0.191±0.005	2.393±0.058	0.113±0.008	-7.588±1.208
	Loc	0.121±0.060	1.269±0.449	0.083±0.060	-13.174±3.193	0.252±0.053	2.512±0.185	0.173±0.034	-7.775±2.947	0.263±0.036	2.741±0.275	0.178±0.027	-7.321±2.572

partitions. This implies that correlations between features are not well preserved. On the other hand, the number of clients showed no effect on the fidelity results for the chosen metrics. Finally, as pointed out previously, the results obtained when applying local DP exhibited significant fluctuations, leading to poor results. These findings are consistent with previous work [Mar21]. An alternative to improve these results is to apply central DP in conjunction with an SMPC protocol. However, the privacy guarantees are less strict compared to those provided by local DP [PHK+23].

6.2.2 Utility Evaluation

This section analyzes the utility results of synthetic data generated with the VAE model for the different baselines. The main challenge when using local DP in the federated setting is that the model requires more epochs to converge, making the process computationally expensive.

Table 6.10 presents the results obtained from the utility analysis of the Adult dataset. The results indicate that in most non-DP scenarios, the centralized setting outperforms the federated setting. The differences between these two baselines are subtle in the i.i.d. settings. However, the non-i.i.d. label skew data partition exhibits ROC-AUC scores close to 0.5, indicating that the model performs nearly at the level of random guessing. This suggests that heterogeneity among clients' data distribution directly impacts the utility of synthetic data generated with the FedVAE approach proposed. This aligns with the results obtained in the case of fidelity.

Similarly, we observe no apparent effect on the utility results when considering the

number of clients. Furthermore, both privacy budgets yield poor utility results for the RF and KNN models, not only in the federated but also in the centralized setting. Possible reasons for this behavior include redundant noise aggregation during the training process and the fact that the hyperparameters specifically targeting DP-SGD were not optimized, as optimizing them was outside the scope of this thesis. Similar conclusions can be drawn from the results obtained in the Bank dataset.

Table 6.10: Utility Results Adult Dataset for 3, 5, and 10 clients under different data partitions (i.i.d, non-i.i.d quantity skew, and non-i.i.d label skew) using FedVAE. Statistical significance is highlighted as follows: green indicates significantly better performance than the baseline, and red indicates worse performance than the baseline. Federated results are compared to the centralized one, while local results are compared to the federated ones.

Model VAE										
NC	Ref	Non-DP			DP ($\epsilon = 0.5$)			DP ($\epsilon = 1.6$)		
		RF	NB	KNN	RF	NB	KNN	RF	NB	KNN
	Real	0.771±0.003	0.640±0.008	0.749±0.003	0.771±0.003	0.640±0.008	0.749±0.003	0.771±0.003	0.640±0.008	0.749±0.003
	Cen	0.755±0.011	0.679±0.059	0.749±0.005	0.499±0.001	0.514±0.069	0.500±0.000	0.529±0.024	0.631±0.082	0.520±0.019
i.i.d										
3	Fed	0.715±0.027	0.696±0.021	0.721±0.014	0.508±0.015	0.546±0.043	0.500±0.032	0.512±0.014	0.501±0.030	0.512±0.024
	Loc	0.751±0.013	0.714±0.036	0.745±0.018	0.499±0.002	0.545±0.063	0.500±0.001	0.500±0.000	0.512±0.021	0.500±0.000
5	Fed	0.704±0.030	0.655±0.047	0.715±0.015	0.484±0.060	0.491±0.029	0.484±0.029	0.487±0.018	0.487±0.087	0.487±0.014
	Loc	0.734±0.014	0.709±0.037	0.733±0.019	0.487±0.021	0.481±0.056	0.483±0.024	0.588±0.196	0.497±0.052	0.498±0.025
10	Fed	0.737±0.009	0.642±0.017	0.744±0.008	0.500±0.000	0.531±0.010	0.498±0.004	0.497±0.004	0.510±0.017	0.499±0.002
	Loc	0.705±0.031	0.679±0.037	0.702±0.029	0.511±0.032	0.497±0.052	0.498±0.025	0.498±0.021	0.516±0.075	0.502±0.021
non-i.i.d (Quantity Skew)										
3	Fed	0.710±0.015	0.695±0.060	0.712±0.012	0.511±0.011	0.561±0.010	0.523±0.030	0.499±0.004	0.471±0.013	0.481±0.021
	Loc	0.740±0.017	0.713±0.027	0.732±0.014	0.498±0.002	0.505±0.048	0.500±0.001	0.500±0.000	0.519±0.043	0.500±0.000
5	Fed	0.734±0.032	0.654±0.022	0.733±0.029	0.562±0.054	0.543±0.055	0.547±0.029	0.500±0.008	0.475±0.026	0.499±0.002
	Loc	0.741±0.017	0.701±0.038	0.736±0.015	0.505±0.018	0.506±0.040	0.503±0.016	0.505±0.037	0.503±0.058	0.503±0.018
10	Fed	0.724±0.012	0.700±0.034	0.734±0.008	0.498±0.005	0.480±0.032	0.498±0.003	0.500±0.000	0.477±0.052	0.500±0.000
	Loc	0.649±0.102	0.671±0.065	0.658±0.108	0.505±0.037	0.503±0.058	0.503±0.018	0.501±0.030	0.510±0.030	0.502±0.016
non-i.i.d (Label Skew)										
3	Fed	0.533±0.024	0.669±0.119	0.519±0.015	0.526±0.016	0.581±0.030	0.522±0.011	0.531±0.021	0.503±0.006	0.509±0.008
	Loc	0.622±0.133	0.630±0.104	0.621±0.123	0.504±0.020	0.487±0.055	0.513±0.037	0.491±0.036	0.534±0.094	0.495±0.013
5	Fed	0.500±0.000	0.502±0.003	0.500±0.000	0.523±0.017	0.573±0.064	0.516±0.012	0.500±0.002	0.548±0.110	0.498±0.002
	Loc	0.630±0.130	0.599±0.093	0.602±0.125	0.505±0.018	0.506±0.040	0.503±0.016	0.501±0.010	0.531±0.065	0.497±0.015
10	Fed	0.500±0.000	0.556±0.079	0.500±0.000	0.502±0.003	0.486±0.014	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000
	Loc	0.564±0.096	0.558±0.083	0.553±0.088	0.504±0.028	0.506±0.044	0.504±0.015	0.498±0.007	0.498±0.028	0.498±0.008

On the other hand, Table 6.11 shows no significant differences between the federated and centralized baselines for the non-DP case in the Cardio dataset. The same holds for the federated and local baselines. Also, in terms of utility, this dataset has better results for the non-i.i.d label skew partition. This is mainly because the target variable in this dataset is balanced, contrary to the Adult and Bank datasets. Therefore, the VAE works well even in this case. However, if we introduce noise with the considered privacy budgets, the results are worse than in the centralized setting for all the algorithms. More precisely, all the results are close to 0.5, meaning the algorithms have no discriminatory power to distinguish between the two classes.

In summary, these results suggest that the performance of the FedVAE model depends on the data partition distribution among clients, as observed in the fidelity results. However, it is also affected by the imbalance of the target variable. On the other hand, the number of clients does not affect the results, while the noise introduction with the privacy

Table 6.11: Utility Results for the Cardio Dataset with 3, 5, and 10 clients under different data partitions (i.i.d, non-i.i.d quantity Skew, and non-i.i.d label skew) using FedVAE. Statistical significance is highlighted as follows: green indicates significantly better performance than the baseline, and red indicates worse performance than the baseline. Federated results are compared to the centralized one, while local results are compared to the federated ones.

Model VAE										
NC	Ref	Non-DP			DP ($\epsilon = 0.5$)			DP ($\epsilon = 1.6$)		
		RF	NB	KNN	RF	NB	KNN	RF	NB	KNN
	Real	0.715±0.003	0.589±0.004	0.652±0.006	0.715±0.003	0.589±0.004	0.652±0.006	0.715±0.003	0.589±0.004	0.652±0.006
	Cen	0.649±0.020	0.659±0.018	0.650±0.019	0.639±0.042	0.565±0.004	0.625±0.036	0.694±0.004	0.660±0.035	0.684±0.009
i.i.d										
3	Fed	0.676±0.020	0.632±0.026	0.678±0.010	0.514±0.019	0.482±0.046	0.467±0.025	0.511±0.012	0.555±0.009	0.512±0.025
	Loc	0.678±0.019	0.654±0.033	0.674±0.014	0.508±0.055	0.512±0.046	0.534±0.071	0.539±0.066	0.536±0.050	0.502±0.069
5	Fed	0.684±0.017	0.644±0.018	0.675±0.004	0.476±0.016	0.510±0.025	0.510±0.040	0.506±0.011	0.531±0.034	0.510±0.014
	Loc	0.680±0.023	0.662±0.024	0.679±0.017	0.517±0.052	0.530±0.060	0.524±0.054	0.513±0.048	0.501±0.051	0.496±0.039
10	Fed	0.683±0.017	0.651±0.039	0.685±0.008	0.495±0.020	0.519±0.071	0.478±0.028	0.518±0.038	0.530±0.026	0.520±0.036
	Loc	0.679±0.018	0.635±0.030	0.673±0.014	0.500±0.031	0.495±0.051	0.492±0.045	0.504±0.038	0.528±0.062	0.508±0.040
non-i.i.d (Quantity Skew)										
3	Fed	0.679±0.011	0.681±0.009	0.682±0.004	0.460±0.020	0.524±0.019	0.440±0.038	0.525±0.022	0.517±0.015	0.511±0.013
	Loc	0.672±0.028	0.658±0.025	0.672±0.021	0.567±0.062	0.511±0.067	0.554±0.057	0.549±0.077	0.550±0.036	0.548±0.052
5	Fed	0.679±0.022	0.686±0.004	0.679±0.013	0.497±0.014	0.465±0.034	0.476±0.040	0.488±0.027	0.456±0.042	0.493±0.046
	Loc	0.671±0.020	0.653±0.029	0.673±0.015	0.516±0.045	0.507±0.050	0.504±0.027	0.541±0.046	0.524±0.042	0.529±0.048
10	Fed	0.680±0.023	0.649±0.029	0.686±0.010	0.505±0.020	0.539±0.059	0.472±0.015	0.488±0.027	0.456±0.042	0.493±0.046
	Loc	0.682±0.028	0.640±0.029	0.674±0.041	0.509±0.039	0.487±0.047	0.509±0.044	0.541±0.046	0.524±0.042	0.529±0.048
non-i.i.d (Label Skew)										
3	Fed	0.663±0.036	0.664±0.015	0.655±0.035	0.518±0.024	0.546±0.013	0.509±0.012	0.514±0.045	0.521±0.041	0.513±0.025
	Loc	0.588±0.075	0.597±0.078	0.589±0.075	0.499±0.001	0.500±0.010	0.500±0.000	0.520±0.038	0.525±0.034	0.520±0.041
5	Fed	0.642±0.059	0.628±0.035	0.656±0.046	0.514±0.020	0.505±0.008	0.501±0.001	0.507±0.010	0.510±0.009	0.488±0.017
	Loc	0.581±0.076	0.580±0.069	0.583±0.077	0.506±0.016	0.501±0.020	0.503±0.008	0.497±0.012	0.507±0.030	0.498±0.007
10	Fed	0.649±0.021	0.637±0.027	0.641±0.025	0.501±0.002	0.505±0.007	0.503±0.004	0.500±0.000	0.462±0.054	0.500±0.000
	Loc	0.580±0.079	0.575±0.065	0.579±0.076	0.502±0.017	0.504±0.047	0.499±0.017	0.497±0.012	0.500±0.040	0.499±0.003

budgets considered provides poor results. Finally, we observe that training classifiers with synthetic data without DP yields similar performance to training on real data for most classifiers across both i.i.d. and non-i.i.d. quantity skew partitions. For instance, in the Adult dataset, the ROC AUC scores for synthetic data are close to those for real data across all algorithms (i.e., the difference is less than 2%). In some cases, results even outperform those obtained with the real data with the NB classifier.

6.2.3 Privacy Evaluation

Table 6.12 shows the results obtained for the Adult dataset regarding the DCR metric. We can observe that the generated synthetic data in the different federated settings exhibits almost identical DCR distributions for the training and holdout records with share values close to 0.5 and even below. This provides empirical evidence that the synthetic data can generalize the patterns in the real data. However, the minimum DCR is 0 across all scenarios, indicating that exact matches exist. In future work, it would be interesting to investigate whether these matches are inliers or outliers in the real dataset using an outlier detection method to assess the potential risk of disclosure. The results also show that average distances to the training and holdout data change fluctuate significantly when differential privacy (DP) is applied during the synthetic data

generation process. Results for the remaining datasets provide similar insights, except for exact matches, which occur in a few scenarios of the synthetic data generated for the Bank dataset.

Table 6.12: DCR results of the holdout assessment for the Adult dataset with 3, 5 and 10 clients under different data partitions (i.i.d and non-i.i.d label skew) using FedVAE.

Model VAE												
Clients	Non-DP				DP (ϵ = 0.5)				DP (ϵ = 1.6)			
	Min DCR	Share	Avg DCR Train	Avg DCR Holdout	Min DCR	Share	Avg DCR Train	Avg DCR Holdout	Min DCR	Share	Avg DCR Train	Avg DCR Holdout
i.i.d												
3	0.000±0.000	0.467±0.026	0.951±0.209	0.930±0.218	0.000±0.000	0.385±0.044	0.736±0.041	0.720±0.057	0.000±0.000	0.353±0.183	0.173±0.082	0.112±0.009
5	0.000±0.000	0.461±0.007	0.796±0.145	0.767±0.119	0.000±0.000	0.474±0.012	0.987±0.078	0.962±0.085	0.000±0.000	0.431±0.022	0.657±0.065	0.625±0.082
10	0.000±0.000	0.452±0.016	0.557±0.037	0.533±0.028	0.014±0.019	0.457±0.015	3.356±0.066	3.327±0.101	0.000±0.000	0.513±0.205	0.390±0.145	0.376±0.099
non-i.i.d (Quantity Skew)												
3	0.000±0.000	0.470±0.010	0.836±0.126	0.818±0.123	0.000±0.000	0.424±0.045	0.942±0.089	0.910±0.110	0.000±0.000	0.289±0.039	0.274±0.126	0.245±0.149
5	0.000±0.000	0.471±0.012	0.803±0.060	0.793±0.072	0.000±0.000	0.411±0.013	0.751±0.127	0.754±0.119	0.000±0.000	0.391±0.121	0.135±0.035	0.135±0.044
10	0.000±0.000	0.458±0.018	0.635±0.103	0.621±0.115	0.000±0.000	0.427±0.018	0.591±0.076	0.578±0.074	0.000±0.000	0.409±0.010	0.407±0.048	0.392±0.037
non-i.i.d (Label Skew)												
3	0.000±0.000	0.467±0.003	0.841±0.057	0.821±0.058	0.000±0.000	0.414±0.010	1.058±0.309	1.023±0.285	0.000±0.000	0.207±0.120	0.151±0.022	0.119±0.033
5	0.000±0.000	0.465±0.015	0.816±0.080	0.799±0.087	0.000±0.000	0.437±0.020	0.705±0.120	0.687±0.136	0.000±0.000	0.376±0.011	0.253±0.116	0.248±0.123
10	0.000±0.000	0.458±0.004	0.706±0.036	0.684±0.036	0.000±0.000	0.420±0.040	0.540±0.035	0.523±0.026	0.000±0.000	0.428±0.053	0.559±0.072	0.558±0.076

Table 6.13 shows the average attribute disclosure risk for all real records in the attack scenarios considered. It is worth noting that the Cardio dataset is not included in the table because the synthetic data generated by the FedVAE fails to create different classes for the sensitive attribute, given its imbalanced nature. For the attack scenario of the Adult dataset, the attribute disclosure risk is smaller on the synthetic data compared to the real data in all the cases considered. This implies that an attacker that only has access to the synthetic data obtains worse predictions on average than the real data. However, for the non-DP cases, the RMEAN still beats the performance of the DUM classifier by 0.14 and 0.172 for the i.i.d. and non-i.i.d. partitions considered, respectively. This indicates that the attacker can still exploit the structure of the synthetic dataset to gain knowledge about the real data. This is, however, directly related to the fact that synthetic data tries to mimic the correlations of real data. Notably, when introducing noise, the RMEAN results are close to or below those from the Dummy Classifier, which means the attacker gets no advantage from accessing the synthetic data in this model. However, as observed from the utility and fidelity results, this indicates that the quality of the synthetic data is poor. Therefore, there is a trade-off when introducing DP to protect privacy, and it needs to be balanced to ensure that the data is still helpful but does not leak sensitive information.

6.3 Sensitivity Analysis of Hyperparameters

As explained in the previous model, we investigate the impact of hyperparameters on the quality of the synthetic data generated in this section. In Figure 6.2, we can see that the hyperparameter with the highest importance concerning the LC metric is the weight decay(l2scale) in all the datasets exhibiting a strong positive correlation in all cases. Furthermore, the loss factor is also significant for the Cardio and the Bank datasets, showing a positive correlation with the performance metric. Other hyperparameters exhibit different trends depending on the dataset.

Table 6.13: Attribute disclosure results with the FedVAE model across different data partitions (i.i.d. and non-i.i.d. label skew) for ten clients

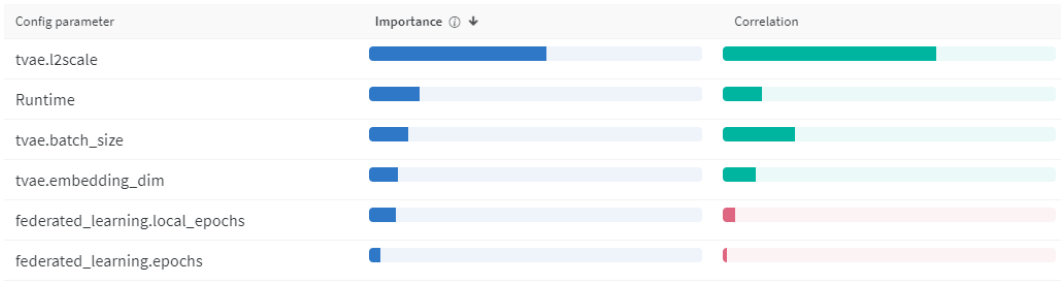
Scenario	Baseline	Model VAE							
		RF	SVM	NB	KNN	LR	ENS	DUM	RMEAN
Scenario 1 (Adult)	Real	0.648±0.053	0.634±0.050	0.618±0.049	0.573±0.097	0.629±0.048	0.637±0.050	0.458	0.623±0.019
	i.i.d								
	DP ($\epsilon = 0.5$)	0.407±0.036	0.448±0.016	0.235±0.057	0.398±0.027	0.456±0.003	0.448±0.013	0.458	0.399±0.021
	DP ($\epsilon = 1.6$)	0.448±0.009	0.458±0.000	0.310±0.060	0.450±0.009	0.453±0.004	0.453±0.005	0.458	0.429±0.024
	Non-DP	0.604±0.039	0.622±0.042	0.599±0.046	0.558±0.052	0.603±0.035	0.621±0.042	0.458	0.601±0.007
	non i.i.d (Label Skew)								
	DP ($\epsilon = 0.5$)	0.409±0.038	0.411±0.041	0.312±0.072	0.398±0.038	0.493±0.072	0.443±0.051	0.329	0.411±0.021
	DP ($\epsilon = 1.6$)	0.398±0.030	0.362±0.035	0.307±0.061	0.397±0.038	0.337±0.042	0.379±0.042	0.329	0.364±0.015
	Non-DP	0.561±0.059	0.575±0.070	0.519±0.069	0.514±0.049	0.562±0.057	0.576±0.065	0.372	0.551±0.019
Scenario 2 (Bank)	Real	0.642±0.016	0.618±0.005	0.603±0.019	0.594±0.021	0.595±0.020	0.616±0.020	0.554	0.611±0.006
	i.i.d								
	DP ($\epsilon = 0.5$)	0.546±0.007	0.545±0.010	0.556±0.012	0.532±0.010	0.558±0.005	0.553±0.006	0.554	0.548±0.005
	DP ($\epsilon = 1.6$)	0.549±0.011	0.554±0.011	0.559±0.014	0.521±0.020	0.562±0.005	0.557±0.011	0.554	0.550±0.006
	Non-DP	0.603±0.004	0.610±0.003	0.599±0.018	0.556±0.011	0.593±0.022	0.609±0.007	0.554	0.595±0.008
	non i.i.d (Label Skew)								
	DP ($\epsilon = 0.5$)	0.509±0.016	0.512±0.018	0.490±0.011	0.510±0.013	0.486±0.012	0.498±0.012	0.518	0.501±0.006
	DP ($\epsilon = 1.6$)	0.568±0.006	0.563±0.005	0.571±0.017	0.552±0.010	0.566±0.005	0.567±0.007	0.554	0.564±0.005
	Non-DP	0.597±0.006	0.597±0.017	0.596±0.021	0.544±0.015	0.594±0.016	0.598±0.016	0.554	0.588±0.005

6.4 Comparison of the two federated approaches

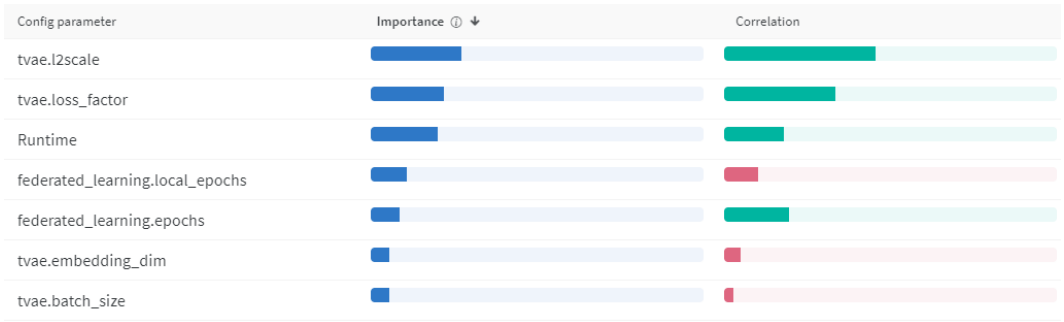
Although we have already seen the results for both FedBN and FedVAE in different scenarios, in this section, we examine the differences between the two methods in more depth to understand their advantages and pitfalls concerning the different dimensions (fidelity, utility, and privacy). For this, we provide additional visual comparisons to complement the results. Note that the visualizations correspond to one of the random splits of the training data used for the synthetic data generation.

We investigate the univariate distributions of different features, taking the Adult dataset as an example. Specifically, we select numeric and categorical features from this dataset and compare the results obtained with the FedBN and FedVAE approaches for the scenario with ten clients and non-i.i.d label skew data partition without DP. This scenario is chosen because our previous analysis showed that it is one of the more challenging for both models. The results obtained from this comparison are presented in Figure 6.3. Interestingly, we can observe that the FedBN model struggles to represent the distribution of highly skewed numerical features as the *capital-loss*. Conversely, the FedVAE model generates synthetic data with similar distributions for the numerical features but encounters challenges with the categorical features, especially when the features are highly imbalanced across the clients, as in the case of the *income* feature, it fails in generating the minority classes.

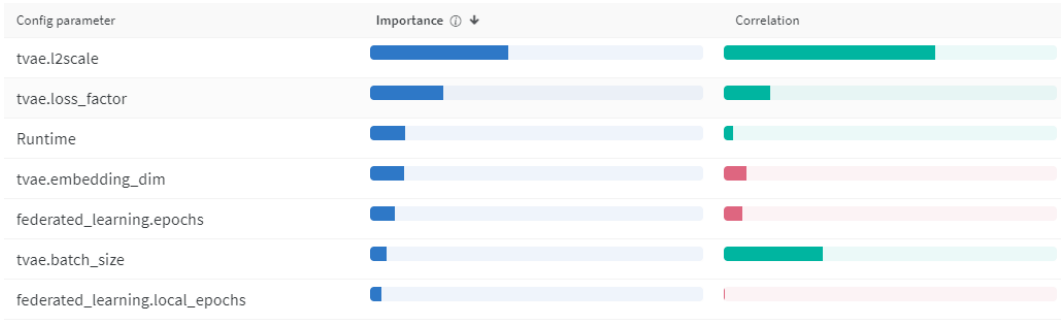
We also investigate the bivariate relationships between features in terms of their correlations. Figure 6.4 shows the results for the same federated scenario previously analyzed. More significant differences in the correlation values are observed for the FedVAE model, particularly for features like the *native-country*, which is highly imbalanced, the model only generates the most frequent class. Therefore, the correlation values are always 0. On the other hand, the FedBN model exhibits more errors in the correlations involving numerical features.



(a) Adult Dataset



(b) Bank Dataset



(c) Cardio Dataset

Figure 6.2: Sensitivity analysis of hyperparameters on synthetic data fidelity using the Log Cluster Metric in a federated setting with 5 clients and a non-i.i.d. label skew partition across all datasets: (a) Adult, (b) Bank, and (c) Cardio, using FedVAE.

In general, when comparing fidelity results for the non-DP cases reported in the last sections for both models, we observe that the FedBN model consistently outperforms the FedVAE model in the PCD and LC metrics. Meanwhile, the M-HD is better for the FedVAE model in the Bank dataset, because this dataset has more numerical features than categorical ones. Similarly, FedVAE outperforms the FedBN model in the pMSE metric in all datasets. As discussed, this is due to the uniform binning applied during preprocessing in the FedBN model. On the other hand, when comparing the utility results between the two models, we observe that the FedBN model ROC AUC scores are better than those from the FedVAE in most scenarios of the Adult and Bank datasets.

However, the FedVAE offers better results for the Cardio dataset.

Moreover, regarding privacy, the attribute disclosure risks are higher for the FedBN model. On the other hand, both models generate similar DCR results. However, the synthetic data generated with the FedVAE contains exact matches, unlike the data generated by the FedBN model.

Finally, it is important to note that the DP mechanisms used for both models are not comparable since they operate at a distinct level, and therefore, we don't analyze the settings with DP here. In particular, the local DP approach used in FedVAE results in the worst results because it injects noise closer to the data. As discussed by [PHK+23], the DP impact on utility is less when applied further from the data. Therefore, the distributed DP approach in FedBN has an advantage over local DP. However, local DP provides stronger privacy guarantees.

In summary, we can conclude that the quality of the synthetic data generated using the FedBN exhibits better performance across datasets and federated settings than the FedVAE model.

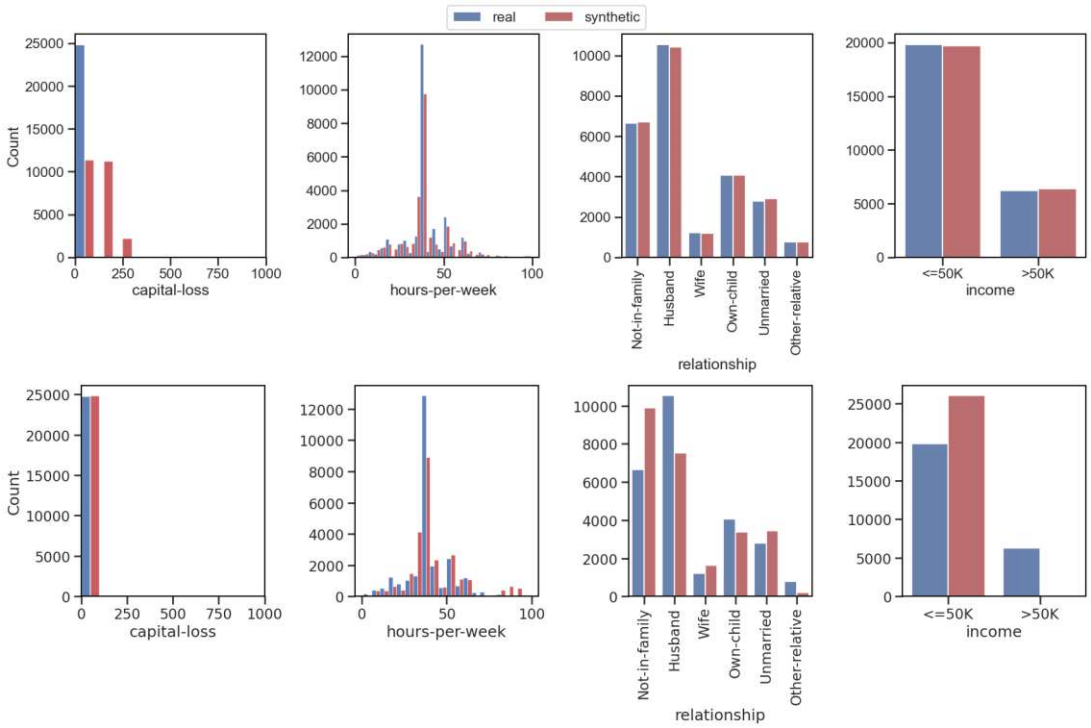


Figure 6.3: Univariate distributions for the non-i.i.d. (label skew) scenario with 10 clients without DP. The top row shows the synthetic data generated with the FedBN model, while the bottom row shows the data generated with the FedVAE model. Each column in the figure corresponds to a variable, with the first two columns corresponding to numerical features and the last two categorical features. The red color distinguishes the synthetic data distribution from the real one (in blue).

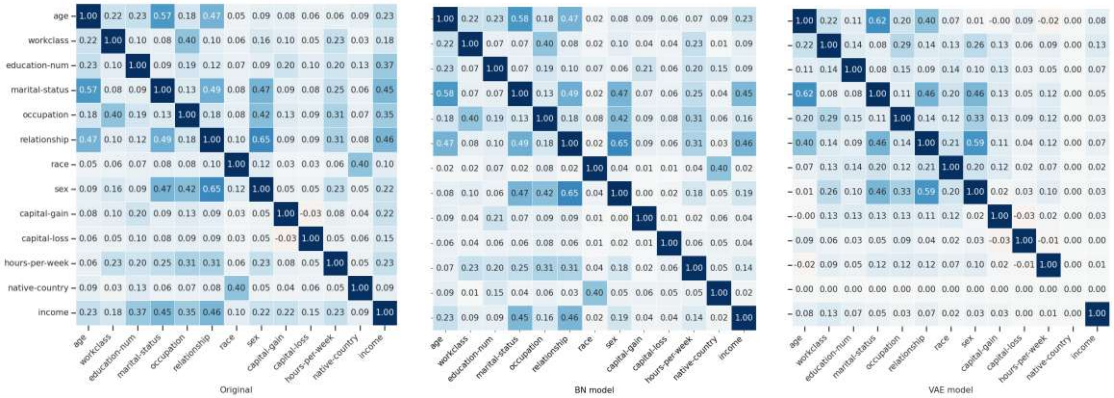


Figure 6.4: Heatmaps showing the correlation between attributes in the real and synthetic datasets for the BN and VAE models, respectively, across 10 clients with non-i.i.d. label skew distribution.

CHAPTER 7

Conclusion

In this chapter, we present the main contributions and insights from this thesis, including a summary of the results, answers to the research questions introduced in Chapter 1, and possible directions for future work. We also discuss the main limitations and challenges encountered during this work.

7.1 Summary and Contributions

In this work, we have explored synthetic data generation for distributed tabular data partitioned horizontally. We adapted two well-known methods used in the centralized setting for generating synthetic tabular data, namely Bayesian Networks and Variational Autoencoders, and evaluated their performance in different settings using three dimensions: fidelity, utility, and privacy. We compare the results against two baselines (the centralized and local). We also highlighted the main findings from the experiments conducted in this work.

More precisely, the main contributions of this thesis are the following:

- We propose a novel aggregation strategy that combines the knowledge of different clients to construct a Bayesian network for synthetic data generation in the distributed setting. The approach is inspired by the works of De Falco et al. [DFDCK+23] and Hittmeir et al. [HME22].
- We adapt the TVAE model introduced by Xu et al. [XSCIV19] from the centralized to the federated setting. In particular, we used the FedAvg strategy to aggregate the encoder and decoder parameters simultaneously from all clients.
- We validate the quality of the synthetic data generated in terms of fidelity, privacy, and utility of both models across multiple scenarios, including different numbers

of clients ($NC \in 3, 5, and 10$), three data partitions (i.i.d, non-i.i.d quantity skew, and non-i.i.d label skew), and cases with and without differential privacy (DP), using two different privacy budgets.

- We compare the results of each model against two baselines: the centralized setting and the local setting. Additionally, we compare the results of both models.
- We investigate the impact of hyperparameters on the fidelity results of the synthetic data. Specifically, with respect to the Log Cluster metric.
- We also discuss the challenges of adapting the centralized models to the federating setting concerning privacy.

7.2 Research Questions

To wrap up the main findings obtained in this thesis, we answer the research questions defined in Section 1.3.

1. **To what extent is federated synthetic tabular data useful?**

   This question covers all the experiments conducted concerning fidelity and utility metrics. Our findings suggest that the usefulness of the synthetic tabular data generated depends on several factors, including the model used, the metric evaluated, the dataset considered, and the variations in data distribution among clients in the federated scenario. Detailed insights are provided in the following two sub-questions.

   a) **How does federated synthetic tabular data compare with centralized synthetic data and local synthetic data from a single client?**

   For the FedBN model, we observe that the federated settings provide comparable or even better results than the centralized ones for the following fidelity metrics in the not differentially private case: M-HD, PCD, and LC. Meanwhile, the average local results from the clients also demonstrate comparable results to those of the federated setting. Furthermore, the differences in pMSE scores across the various baselines are negligible in most scenarios. When distributed DP is used in the synthetic data generation process, we observe that the fidelity results for the Adult dataset are better in the federated setting than in the centralized and local settings. Conversely, the results on the Bank and Cardio datasets are comparable or perform worse in specific cases and metrics. Regarding utility, the baselines exhibit less significant differences when comparing the ROC AUC scores with respect to the federated setting. On the other hand, for the VAE model, federated results exhibit more significant differences compared to the baselines in fidelity and utility metrics depending on the scenario evaluated.

   While we expected the federated setting to outperform the local baseline, this was not always the case. Our findings suggest that average local results

can provide comparable or better results than the federated setting when the clients' data distribution is representative of the overall distribution. In such cases, the federated approach may not provide additional benefits over local data generation.

b) **How do fidelity and utility of federated synthetic tabular data compare for the different synthetic data generation techniques and datasets?**

Our findings regarding fidelity metrics suggest that the FedBN model consistently outperforms the FedVAE model in the PCD and LC metrics. Meanwhile, the M-HD is better for the FedVAE model in the Bank dataset because this dataset has more numerical features than categorical ones. Similarly, FedVAE outperforms the FedBN model in the pMSE metric in all datasets.

On the other hand, when comparing the utility results between the two models, the findings demonstrate that the ROC AUC scores for the FedBN model outperform the results from the FedVAE in most scenarios of the Adult and Bank datasets. However, the FedVAE offers better results for the Cardio dataset.

2. **To what extent is the federated generation of synthetic tabular data sensitive to hyperparameters and data distribution?**

This question considers the results for the three dimensions explored in the experiments. Our findings suggested that the hyperparameters and data distribution significantly impact the quality of the synthetic data generated in the federated setting.

a) **To what degree do the model hyperparameters used in the federated synthetic data generator affect the fidelity?**

The model hyperparameters have a direct impact on the LC metric results. For the FedBN, we observed that the aggregation interval was highly important for the performance. However, the correlation can be positive or negative depending on the dataset. Meanwhile, for the FedVAE, the hyperparameter with the highest importance concerning the LC metric was the weight decay (scale) for all the datasets, exhibiting a strong positive correlation in all cases. Furthermore, the loss factor was also significant for the Cardio and the Bank datasets, showing a positive correlation with the performance metric.

b) **To what extent do the number of clients and the data heterogeneity affect the fidelity and utility of synthetic tabular data generated in a federated setting?**

The number of clients shows no significant effect in most datasets and scenarios. On the other hand, we found that the heterogeneity in data distribution among clients in the federated setting has a direct impact on the performance of the FedVAE model, particularly in terms of the PCD metric, showing worse results

for the non-i.i.d. label skew data partitions. Similar findings are observed from the FedBN model. However, the impact is highly dependent on the dataset considered.

3. **To what extent does federated synthetic tabular data preserve privacy?**

   This question covers all experiments conducted using the holdout assessment for the DCR and attribute disclosure results. Detailed insights are provided in the following sub-questions.

   a) **To what extent do records in federated synthetic tabular data resemble those in real data?**

      We observe that the share of records that are closer to the training data than to holdout data is close to 0.5 for the FedBN and FedVAE models. This provides evidence of these models' capabilities to generalize the patterns found in the real data. However, in many scenarios using FedVAE, the generated synthetic data contained exact matches, requiring further analysis to determine whether the model is overfitting the training data, potentially leading to disclosure risks, or if it occurs by chance.

   b) **To what extent does federated synthetic tabular data prevent attribute disclosure?**

      We observe that the attribute disclosure risks are higher for the FedBN model for the attack scenarios analyzed in the Adult and Bank datasets, even when using distributed DP in the parameter learning step. Specifically, for the Adult dataset, the results of the non-DP scenarios using an ENS classifier on the synthetic data show ROC AUC scores close to the real data for both data partitions, with only a 0.006 difference. Similar results are observed for the Bank dataset, with a difference of around 0.002. When using DP, results still outperform the RMEAN and still beat the DUM classifier, meaning that the attacker can still exploit the structure of the synthetic data. On the other hand, for the FedVAE model, the attribute disclosure risk is less compared to FedBN. However, the RMEAN score still beats the DUM classifier, indicating the attacker can still exploit the structure of the synthetic data.

7.3 Future Work

Several directions can be considered to extend this work. Indeed, federated learning for generating synthetic data is an emerging research area, and the amount of work in this regard is still limited, particularly for tabular data. While working on this thesis, we encountered several challenges. For instance, there is no agreement on how to evaluate and compare synthetic data generators consistently. Usually, the metrics and dimensions considered differ in the papers, limiting the comparability of the results. Furthermore, the computation details are not always specified, even when using the same metrics. Based on these challenges, we provide a list of possible directions for future work:

- Develop guidelines guidelines or standards for evaluating synthetic data consistently.
- Investigate hyperparameter tuning for synthetic data generation in the federated setting using multiple metrics and a different strategy to the one used in this work.
- Evaluate the impact on computational resources of using SMPC and HE in the stages of the federated training.
- VAE models suffer when there are imbalanced columns in centralized and federated settings. Therefore, a possible way to extend this work is to leverage ideas proposed for other methods to improve the generation of synthetic tabular data. For instance, one potential option is to consider conditional sampling as previously proposed for the CTGAN model [DLH+23].
- When comparing federated approaches with the baselines, running more iterations to generate synthetic datasets can yield more robust results and enable a more representative comparison of statistical differences.
- The evaluation of other aggregation strategies can also be a possible direction to extend this work.
- Investigating in detail different DP approaches in the federated setting and comparing their impact on the synthetic data generated.

APPENDIX A

Complementary Results

Here, we present additional results from the experiments conducted that offer similar insights to the ones discussed in chapter 6.

Table A.1: Utility results for the Bank Dataset with 3, 5, and 10 clients under different data partitions (i.i.d, non-i.i.d Quantity Skew, and non-i.i.d Label Skew) using the FedBN model. Statistical significance is highlighted as follows: green indicates significantly better performance than the baseline, and red indicates worse performance than the baseline. Federated results are compared to the centralized one, while local results are compared to the federated ones.

Model BN										
NC	**Ref**	**Non-DP**			**DP ($\epsilon = 0.5$)**			**DP ($\epsilon = 1.6$)**		
		RF	**NB**	**KNN**	**RF**	**NB**	**KNN**	**RF**	**NB**	**KNN**
	Real	0.695±0.005	0.679±0.002	0.651±0.004	0.695±0.005	0.679±0.002	0.651±0.004	0.695±0.005	0.679±0.002	0.651±0.004
	Cen	0.626±0.011	0.600±0.012	0.591±0.000	0.509±0.005	0.500±0.000	0.519±0.005	0.517±0.003	0.501±0.001	0.508±0.006
i.i.d										
3	Fed	0.617±0.010	0.599±0.015	0.590±0.009	0.568±0.015	0.501±0.000	0.554±0.019	0.550±0.032	0.501±0.001	0.522±0.006
	Loc	0.625±0.017	0.593±0.025	0.589±0.008	0.546±0.022	0.501±0.001	0.531±0.010	0.528±0.016	0.500±0.001	0.516±0.008
5	Fed	0.617±0.018	0.603±0.002	0.591±0.004	0.546±0.010	0.500±0.000	0.531±0.008	0.527±0.003	0.500±0.000	0.511±0.002
	Loc	0.630±0.018	0.600±0.030	0.590±0.009	0.574±0.031	0.510±0.031	0.532±0.020	0.560±0.034	0.512±0.034	0.546±0.022
10	Fed	0.617±0.006	0.606±0.001	0.579±0.002	0.522±0.005	0.500±0.000	0.523±0.004	0.536±0.007	0.500±0.000	0.513±0.003
	Loc	0.636±0.018	0.618±0.022	0.594±0.011	0.576±0.037	0.514±0.038	0.522±0.021	0.583±0.040	0.519±0.041	0.535±0.016
non-i.i.d (Quantity Skew)										
3	Fed	0.625±0.014	0.600±0.015	0.589±0.003	0.520±0.008	0.500±0.000	0.519±0.004	0.517±0.002	0.501±0.001	0.508±0.002
	Loc	0.630±0.014	0.595±0.022	0.593±0.009	0.551±0.025	0.510±0.025	0.530±0.018	0.542±0.034	0.501±0.002	0.528±0.015
5	Fed	0.611±0.003	0.603±0.003	0.583±0.001	0.520±0.006	0.500±0.000	0.529±0.008	0.520±0.005	0.500±0.000	0.518±0.005
	Loc	0.635±0.016	0.611±0.016	0.590±0.013	0.570±0.021	0.515±0.027	0.529±0.018	0.555±0.035	0.508±0.014	0.527±0.018
10	Fed	0.633±0.007	0.612±0.004	0.585±0.002	0.529±0.013	0.500±0.000	0.531±0.009	0.575±0.059	0.500±0.000	0.519±0.008
	Loc	0.624±0.021	0.617±0.030	0.588±0.014	0.549±0.043	0.509±0.046	0.518±0.027	0.551±0.035	0.508±0.036	0.523±0.017
non-i.i.d (Label Skew)										
3	Fed	0.610±0.031	0.614±0.005	0.579±0.006	0.534±0.019	0.501±0.001	0.527±0.015	0.544±0.023	0.500±0.000	0.526±0.014
	Loc	0.594±0.113	0.599±0.090	0.569±0.089	0.528±0.055	0.495±0.016	0.508±0.016	0.543±0.052	0.500±0.000	0.503±0.029
5	Fed	0.632±0.010	0.606±0.009	0.586±0.004	0.552±0.034	0.547±0.067	0.528±0.000	0.563±0.036	0.500±0.000	0.522±0.012
	Loc	0.594±0.112	0.586±0.089	0.571±0.085	0.530±0.038	0.505±0.029	0.509±0.021	0.518±0.075	0.499±0.026	0.505±0.024
10	Fed	0.637±0.008	0.618±0.004	0.592±0.006	0.579±0.009	0.500±0.000	0.545±0.021	0.604±0.015	0.501±0.001	0.523±0.002
	Loc	0.602±0.112	0.580±0.074	0.574±0.079	0.527±0.061	0.506±0.044	0.503±0.016	0.548±0.060	0.509±0.045	0.514±0.022

Table A.2: DCR results of the holdout assessment for the Adult dataset with 3, 5 and 10 clients under different data partitions (i.i.d and non-i.i.d label skew) for the FedBN model.

Model BN												
Clients	Non-DP				DP ($\epsilon = 0.5$)				DP ($\epsilon = 1.6$)			
	Min DCR	Share	Avg DCR Train	Avg DCR Holdout	Min DCR	Share	Avg DCR Train	Avg DCR Holdout	Min DCR	Share	Avg DCR Train	Avg DCR Holdout
i.i.d												
3	0.010±0.001	0.483±0.005	0.444±0.004	0.435±0.002	0.075±0.031	0.470±0.003	2.157±0.070	2.145±0.071	0.024±0.004	0.471±0.007	1.766±0.313	1.755±0.312
5	0.008±0.001	0.483±0.001	0.445±0.002	0.438±0.004	0.069±0.019	0.476±0.003	2.166±0.002	2.155±0.004	0.012±0.004	0.473±0.003	1.694±0.007	1.682±0.010
10	0.008±0.003	0.481±0.007	0.447±0.004	0.440±0.001	0.045±0.020	0.470±0.000	2.131±0.011	2.118±0.012	0.016±0.002	0.473±0.002	1.680±0.183	1.669±0.183
non-i.i.d (Quantity Skew)												
3	0.009±0.001	0.479±0.004	0.447±0.004	0.439±0.004	0.047±0.016	0.475±0.006	2.089±0.072	2.079±0.071	0.018±0.003	0.473±0.002	1.462±0.208	1.452±0.207
5	0.011±0.001	0.487±0.001	0.447±0.004	0.439±0.002	0.086±0.035	0.475±0.001	2.203±0.001	2.192±0.003	0.022±0.002	0.475±0.003	2.026±0.004	2.016±0.006
10	0.010±0.003	0.475±0.004	0.470±0.034	0.462±0.036	0.057±0.016	0.474±0.003	2.143±0.029	2.132±0.028	0.015±0.004	0.476±0.002	1.692±0.202	1.682±0.202
non-i.i.d (Label Skew)												
3	0.008±0.001	0.486±0.005	0.449±0.004	0.442±0.006	0.170±0.071	0.469±0.005	2.206±0.001	2.192±0.004	0.019±0.005	0.471±0.004	1.967±0.055	1.955±0.053
5	0.010±0.001	0.484±0.005	0.452±0.001	0.444±0.003	0.126±0.048	0.476±0.004	2.202±0.002	2.191±0.005	0.023±0.002	0.474±0.007	2.007±0.013	1.996±0.016
10	0.009±0.003	0.488±0.002	0.458±0.011	0.452±0.016	0.043±0.015	0.474±0.004	2.072±0.104	2.061±0.102	0.014±0.003	0.475±0.001	1.478±0.317	1.468±0.317

Table A.3: DCR results of the holdout assessment for the Adult dataset with 3, 5 and 10 clients under different data partitions (i.i.d and non-i.i.d label skew) for the FedBN model.

Model BN												
Clients	Non-DP				DP ($\epsilon = 0.5$)				DP ($\epsilon = 1.6$)			
	Min DCR	Share	Avg DCR Train	Avg DCR Holdout	Min DCR	Share	Avg DCR Train	Avg DCR Holdout	Min DCR	Share	Avg DCR Train	Avg DCR Holdout
i.i.d												
3	0.010±0.001	0.483±0.005	0.444±0.004	0.435±0.002	0.075±0.031	0.470±0.003	2.157±0.070	2.145±0.071	0.024±0.004	0.471±0.007	1.766±0.313	1.755±0.312
5	0.008±0.001	0.483±0.001	0.445±0.002	0.438±0.004	0.069±0.019	0.476±0.003	2.166±0.002	2.155±0.004	0.012±0.004	0.473±0.003	1.694±0.007	1.682±0.010
10	0.008±0.003	0.481±0.007	0.447±0.004	0.440±0.001	0.045±0.020	0.470±0.000	2.131±0.011	2.118±0.012	0.016±0.002	0.473±0.002	1.680±0.183	1.669±0.183
non-i.i.d (Quantity Skew)												
3	0.009±0.001	0.479±0.004	0.447±0.004	0.439±0.004	0.047±0.016	0.475±0.006	2.089±0.072	2.079±0.071	0.018±0.003	0.473±0.002	1.462±0.208	1.452±0.207
5	0.011±0.001	0.487±0.001	0.447±0.004	0.439±0.002	0.086±0.035	0.475±0.001	2.203±0.001	2.192±0.003	0.022±0.002	0.475±0.003	2.026±0.004	2.016±0.006
10	0.010±0.003	0.475±0.004	0.470±0.034	0.462±0.036	0.057±0.016	0.474±0.003	2.143±0.029	2.132±0.028	0.015±0.004	0.476±0.002	1.692±0.202	1.682±0.202
non-i.i.d (Label Skew)												
3	0.008±0.001	0.486±0.005	0.449±0.004	0.442±0.006	0.170±0.071	0.469±0.005	2.206±0.001	2.192±0.004	0.019±0.005	0.471±0.004	1.967±0.055	1.955±0.053
5	0.010±0.001	0.484±0.005	0.452±0.001	0.444±0.003	0.126±0.048	0.476±0.004	2.202±0.002	2.191±0.005	0.023±0.002	0.474±0.007	2.007±0.013	1.996±0.016
10	0.009±0.003	0.488±0.002	0.458±0.011	0.452±0.016	0.043±0.015	0.474±0.004	2.072±0.104	2.061±0.102	0.014±0.003	0.475±0.001	1.478±0.317	1.468±0.317

Table A.4: Fidelity Results Cardio Dataset for 3, 5, and 10 clients under different data partitions (i.i.d, non-i.i.d Quantity Skew, and non-i.i.d Label Skew) with FedVAE. Statistical significance is highlighted as follows: green indicates significantly better performance than the baseline, and red indicates worse performance than the baseline. Federated results are compared to the centralized one, while local results are compared to the federated ones.

Model VAE													
NC	Bas.	Non-DP				DP ($\epsilon = 0.5$)				DP ($\epsilon = 1.6$)			
		M-HD a	PCD a	pMSE a	LC a	M-HD a	PCD a	pMSE a	LC a	M-HD a	PCD a	pMSE a	LC a
	Cen	0.278±0.014	1.995±0.194	0.147±0.014	-12.457±1.970	0.406±0.028	2.780±0.276	0.191±0.021	-9.774±2.006	0.375±0.009	2.941±0.173	0.176±0.006	-13.414±1.607
i.i.d													
3	Fed	0.286±0.003	2.336±0.170	0.152±0.004	-10.363±0.892	0.449±0.026	2.800±0.094	0.232±0.003	-8.070±2.310	0.447±0.019	2.859±0.004	0.239±0.003	-4.945±1.179
	Loc	0.273±0.010	2.249±0.223	0.146±0.007	-10.877±1.571	0.387±0.021	2.571±0.143	0.191±0.013	-8.535±3.846	0.414±0.022	2.684±0.042	0.207±0.010	-7.656±2.997
5	Fed	0.283±0.026	2.168±0.087	0.153±0.017	-12.296±1.299	0.460±0.023	2.786±0.088	0.237±0.002	-7.619±4.836	0.451±0.030	2.849±0.003	0.241±0.002	-5.832±2.979
	Loc	0.269±0.016	2.440±0.149	0.149±0.008	-13.514±2.529	0.377±0.027	2.634±0.087	0.202±0.012	-7.810±2.687	0.395±0.029	2.668±0.033	0.202±0.016	-9.181±3.166
10	Fed	0.287±0.024	2.448±0.067	0.151±0.017	-12.585±1.888	0.436±0.022	2.777±0.093	0.218±0.016	-5.294±1.673	0.460±0.031	2.857±0.021	0.238±0.002	-5.428±0.955
	Loc	0.276±0.019	2.464±0.163	0.159±0.025	-12.080±2.415	0.384±0.028	2.661±0.029	0.196±0.022	-7.160±2.963	0.392±0.028	2.742±0.094	0.195±0.016	-7.858±2.622
non-i.i.d (Quantity Skew)													
3	Fed	0.280±0.020	2.379±0.027	0.150±0.011	-11.808±2.680	0.448±0.028	2.662±0.007	0.234±0.005	-6.675±1.726	0.444±0.018	2.866±0.020	0.239±0.002	-5.644±1.265
	Loc	0.276±0.010	2.268±0.205	0.150±0.011	-11.356±1.492	0.393±0.014	2.597±0.067	0.195±0.015	-6.376±1.586	0.406±0.020	2.699±0.086	0.204±0.008	-9.460±3.311
5	Fed	0.269±0.014	2.237±0.078	0.145±0.013	-14.165±3.808	0.453±0.017	2.670±0.009	0.236±0.004	-6.358±4.220	0.461±0.023	2.849±0.004	0.240±0.004	-5.872±1.006
	Loc	0.272±0.010	2.399±0.164	0.151±0.013	-12.540±1.706	0.387±0.028	2.645±0.096	0.201±0.013	-6.239±2.587	0.398±0.025	2.684±0.052	0.202±0.015	-7.499±2.607
10	Fed	0.299±0.017	2.423±0.046	0.154±0.017	-13.463±2.425	0.453±0.028	2.783±0.096	0.238±0.004	-6.928±1.073	0.463±0.021	2.852±0.001	0.241±0.002	-5.800±1.331
	Loc	0.296±0.035	2.549±0.246	0.160±0.022	-11.612±2.865	0.368±0.038	2.257±0.564	0.198±0.017	-6.099±3.221	0.379±0.034	2.488±0.405	0.201±0.014	-7.005±3.535
non-i.i.d (Label Skew)													
3	Fed	0.279±0.033	2.240±0.107	0.153±0.026	-12.016±5.188	0.461±0.032	2.655±0.012	0.232±0.003	-4.275±0.967	0.474±0.038	2.730±0.085	0.240±0.003	-4.079±1.036
	Loc	0.296±0.031	2.310±0.205	0.160±0.014	-8.699±2.395	0.421±0.024	2.712±0.202	0.201±0.006	-3.215±0.547	0.440±0.015	2.756±0.099	0.213±0.008	-2.922±0.098
5	Fed	0.294±0.031	2.301±0.251	0.161±0.013	-12.969±6.356	0.472±0.032	2.668±0.007	0.238±0.001	-2.851±0.048	0.478±0.029	2.797±0.086	0.241±0.001	-4.126±1.804
	Loc	0.300±0.025	2.395±0.270	0.161±0.021	-9.370±3.241	0.392±0.034	2.635±0.246	0.198±0.011	-3.473±1.004	0.411±0.035	2.769±0.135	0.209±0.011	-3.014±0.336
10	Fed	0.314±0.010	2.223±0.032	0.177±0.004	-6.284±1.540	0.477±0.043	2.973±0.074	0.237±0.003	-2.893±0.076	0.492±0.030	2.970±0.080	0.244±0.000	-2.870±0.055
	Loc	0.323±0.032	2.526±0.412	0.175±0.018	-8.237±3.354	0.392±0.034	2.612±0.438	0.202±0.014	-3.612±1.586	0.419±0.031	2.838±0.180	0.211±0.017	-3.685±1.985

Table A.5: Utility Results for the Bank Dataset with 3, 5, and 10 clients under different data partitions (i.i.d, non-i.i.d quantity skew, and non-i.i.d label skew) using the FedVAE model. Statistical significance is highlighted as follows: green indicates significantly better performance than the baseline, and red indicates worse performance than the baseline. Federated results are compared to the centralized one, while local results are compared to the federated ones.

Model VAE										
NC	Ref	Non-DP			DP ($\epsilon = 0.5$)			DP ($\epsilon = 1.6$)		
		RF	NB	KNN	RF	NB	KNN	RF	NB	KNN
	Real	0.695±0.005	0.679±0.002	0.651±0.004	0.695±0.005	0.679±0.002	0.651±0.004	0.695±0.005	0.679±0.002	0.651±0.004
	Cen	0.562±0.038	0.661±0.030	0.555±0.026	0.501±0.001	0.497±0.089	0.500±0.001	0.507±0.002	0.632±0.019	0.515±0.010
i.i.d										
3	Fed	0.534±0.026	0.648±0.014	0.536±0.024	0.504±0.017	0.499±0.033	0.505±0.014	0.539±0.055	0.556±0.056	0.529±0.038
	Loc	0.600±0.014	0.676±0.025	0.578±0.015	0.500±0.000	0.468±0.047	0.500±0.000	0.500±0.000	0.453±0.055	0.501±0.003
5	Fed	0.522±0.007	0.648±0.016	0.544±0.007	0.500±0.000	0.501±0.026	0.499±0.002	0.500±0.000	0.550±0.029	0.500±0.000
	Loc	0.613±0.033	0.680±0.027	0.593±0.023	0.500±0.000	0.422±0.033	0.500±0.000	0.500±0.000	0.460±0.056	0.500±0.000
10	Fed	0.570±0.028	0.656±0.002	0.565±0.022	0.500±0.000	0.544±0.026	0.501±0.001	0.500±0.000	0.419±0.040	0.500±0.000
	Loc	0.636±0.022	0.692±0.044	0.604±0.018	0.501±0.003	0.503±0.039	0.501±0.006	0.500±0.001	0.448±0.045	0.500±0.000
non-i.i.d (Quantity Skew)										
3	Fed	0.514±0.003	0.645±0.014	0.530±0.007	0.500±0.000	0.507±0.010	0.503±0.002	0.508±0.010	0.510±0.040	0.515±0.019
	Loc	0.594±0.010	0.681±0.025	0.576±0.009	0.500±0.000	0.455±0.062	0.500±0.001	0.500±0.000	0.520±0.077	0.500±0.000
5	Fed	0.533±0.008	0.632±0.006	0.534±0.008	0.499±0.001	0.497±0.012	0.504±0.006	0.510±0.011	0.555±0.039	0.513±0.016
	Loc	0.617±0.035	0.680±0.028	0.592±0.025	0.500±0.000	0.448±0.067	0.500±0.000	0.500±0.001	0.500±0.072	0.500±0.000
10	Fed	0.531±0.010	0.647±0.012	0.539±0.006	0.500±0.000	0.542±0.043	0.499±0.001	0.500±0.000	0.426±0.055	0.500±0.000
	Loc	0.608±0.045	0.683±0.031	0.584±0.032	0.504±0.014	0.469±0.047	0.501±0.006	0.504±0.013	0.481±0.066	0.501±0.004
non-i.i.d (Label Skew)										
3	Fed	0.500±0.000	0.500±0.000	0.500±0.000	0.527±0.036	0.520 ±0.015	0.508±0.008	0.500±0.000	0.505±0.012	0.506±0.003
	Loc	0.562±0.038	0.661±0.030	0.555±0.026	0.500±0.001	0.511±0.042	0.500±0.001	0.501±0.003	0.519±0.045	0.501±0.002
5	Fed	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000	0.503±0.037	0.502±0.002	0.500±0.000	0.510±0.016	0.505±0.004
	Loc	0.584±0.117	0.610±0.092	0.564±0.090	0.507±0.030	0.500±0.022	0.502±0.012	0.501±0.002	0.500±0.031	0.499±0.004
10	Fed	0.500±0.000	0.500±0.000	0.500±0.000	0.500±0.000	0.532±0.094	0.501±0.001	0.500±0.000	0.439±0.005	0.500±0.000
	Loc	0.595±0.114	0.615±0.107	0.575±0.091	0.503±0.010	0.499±0.056	0.503±0.009	0.504±0.014	0.494±0.029	0.502±0.017

Table A.6: DCR results of the holdout assessment for the Bank dataset with 10 clients under different data partitions (i.i.d and non-i.i.d label skew) using the FedVAE model.

Model BN												
Clients	Non-DP				DP ($\epsilon = 0.5$)				DP ($\epsilon = 1.6$)			
	Min DCR	Share	Avg DCR Train	Avg DCR Holdout	Min DCR	Share	Avg DCR Train	Avg DCR Holdout	Min DCR	Share	Avg DCR Train	Avg DCR Holdout
i.i.d												
3	0.001±0.000	0.485±0.016	0.499±0.011	0.490±0.012	0.014±0.001	0.471±0.037	1.280±0.006	1.271±0.018	0.004±0.001	0.472±0.025	1.012±0.104	0.996±0.118
5	0.001±0.000	0.488±0.015	0.569±0.023	0.557±0.028	0.002±0.001	0.487±0.042	0.738±0.101	0.725±0.085	0.001±0.000	0.477±0.050	0.453±0.052	0.451±0.078
10	0.001±0.000	0.491±0.015	0.573±0.043	0.563±0.037	0.003±0.000	0.474±0.027	1.015±0.053	0.994±0.043	0.000±0.000	0.464±0.031	0.450±0.045	0.427±0.052
non-i.i.d (Quantity Skew)												
3	0.001±0.000	0.482±0.020	0.491±0.016	0.480±0.018	0.008±0.003	0.495±0.040	1.059±0.125	1.050±0.133	0.002±0.001	0.484±0.036	0.845±0.135	0.833±0.151
5	0.001±0.000	0.484±0.014	0.475±0.015	0.465±0.015	0.006±0.000	0.453±0.039	1.080±0.228	1.056±0.232	0.003±0.001	0.480±0.012	0.897±0.053	0.884±0.059
10	0.000±0.000	0.486±0.014	0.489±0.009	0.478±0.009	0.003±0.001	0.471±0.009	0.704±0.042	0.689±0.030	0.001±0.000	0.482±0.046	0.484±0.039	0.471±0.030
non-i.i.d (Label Skew)												
3	0.001±0.000	0.489±0.018	0.434±0.019	0.426±0.024	0.011±0.003	0.478±0.014	1.235±0.128	1.227±0.122	0.005±0.002	0.472±0.024	1.335±0.218	1.315±0.224
5	0.001±0.000	0.488±0.014	0.438±0.004	0.429±0.005	0.011±0.003	0.476±0.017	1.136±0.052	1.128±0.047	0.006±0.005	0.450±0.063	1.098±0.215	1.064±0.195
10	0.001±0.000	0.486±0.012	0.487±0.012	0.475±0.010	0.002±0.000	0.456±0.015	0.709±0.044	0.681±0.060	0.001±0.000	0.491±0.023	0.367±0.019	0.359±0.027

Table A.7: DCR results of the holdout assessment for the Cardio dataset with 3, 5 and 10 clients under different data partitions (i.i.d and non-i.i.d label skew) using the FedVAE model.

Model BN												
Clients	Non-DP				DP ($\epsilon = 0.5$)				DP ($\epsilon = 1.6$)			
	Min DCR	Share	Avg DCR Train	Avg DCR Holdout	Min DCR	Share	Avg DCR Train	Avg DCR Holdout	Min DCR	Share	Avg DCR Train	Avg DCR Holdout
i.i.d												
3	0.000±0.000	0.404±0.102	0.019±0.001	0.017±0.001	0.000±0.000	0.350±0.131	0.014±0.001	0.012±0.001	0.000±0.000	0.365±0.155	0.014±0.002	0.012±0.001
5	0.000±0.000	0.397±0.111	0.019±0.001	0.017±0.002	0.000±0.000	0.375±0.126	0.013±0.000	0.012±0.002	0.000±0.000	0.366±0.091	0.013±0.001	0.011±0.002
10	0.000±0.000	0.406±0.099	0.019±0.001	0.017±0.001	0.000±0.000	0.426±0.142	0.017±0.006	0.015±0.003	0.000±0.000	0.373±0.127	0.014±0.001	0.012±0.002
non-i.i.d (Quantity Skew)												
3	0.000±0.000	0.402±0.099	0.019±0.001	0.017±0.001	0.000±0.000	0.363±0.135	0.014±0.002	0.012±0.001	0.000±0.000	0.345±0.141	0.014±0.002	0.012±0.001
5	0.000±0.000	0.400±0.101	0.021±0.001	0.018±0.002	0.000±0.000	0.357±0.141	0.014±0.002	0.012±0.001	0.000±0.000	0.350±0.128	0.014±0.001	0.012±0.001
10	0.000±0.000	0.412±0.101	0.018±0.001	0.017±0.001	0.000±0.000	0.347±0.136	0.014±0.002	0.012±0.001	0.000±0.000	0.357±0.127	0.014±0.001	0.012±0.001
non-i.i.d (Label Skew)												
3	0.000±0.000	0.404±0.112	0.020±0.001	0.018±0.002	0.000±0.000	0.302±0.146	0.016±0.003	0.013±0.001	0.000±0.000	0.302±0.165	0.016±0.004	0.013±0.001
5	0.000±0.000	0.413±0.109	0.018±0.002	0.017±0.000	0.000±0.000	0.321±0.132	0.016±0.003	0.013±0.002	0.000±0.000	0.327±0.093	0.014±0.002	0.012±0.002
10	0.000±0.000	0.403±0.120	0.018±0.002	0.016±0.001	0.000±0.000	0.304±0.153	0.016±0.004	0.013±0.002	0.000±0.000	0.291±0.160	0.016±0.004	0.013±0.001

List of Figures

List of Tables

List of Algorithms

Bibliography

[AAUC18] Abbas Acar, Hidayet Aksu, A Selcuk Uluagac, and Mauro Conti. A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (Csur)*, 51(4):1–35, 2018.

[AC15] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, 2(1):1–18, 2015.

[ACG+16] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[AMR+19] Sean Augenstein, H Brendan McMahan, Daniel Ramage, Swaroop Ramaswamy, Peter Kairouz, Mingqing Chen, Rajiv Mathews, et al. Generative models for effective ml on private, decentralized datasets. *arXiv preprint arXiv:1911.06679*, 2019.

[AVBSvdS22] Ahmed Alaa, Boris Van Breugel, Evgeny S. Saveliev, and Mihaela van der Schaar. How faithful is your synthetic data? Sample-level metrics for evaluating and auditing generative models. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 290–306. PMLR, 17–23 Jul 2022.

[BBG+20] James Henry Bell, Kallista A Bonawitz, Adrià Gascón, Tancrède Lepoint, and Mariana Raykova. Secure single-server aggregation with (poly) logarithmic overhead. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1253–1269, 2020.

[BDH18] David Basin, Søren Debois, and Thomas Hildebrandt. On purpose and by necessity: compliance under the gdpr. In *Financial Cryptography and Data Security: 22nd International Conference, FC 2018, Nieuwpoort, Curaçao, February 26–March 2, 2018, Revised Selected Papers 22*, pages 20–37. Springer, 2018.

[BHA+23] Mohammad Ahmed Basri, Bing Hu, Abu Yousuf Md Abdullah, Shu-Feng Tsao, Zahid Butt, and Helen Chen. A hyperparameter tuning framework for tabular synthetic data generation methods. *Journal of Computational Vision and Imaging Systems*, 9(1):76–79, 2023.

[BIK+17] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[BK96] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[BKKZ22] Alexander Brecko, Erik Kajati, Jiri Koziorek, and Iveta Zolotova. Federated learning for edge computing: A survey. *Applied Sciences*, 12(18):9124, 2022.

[BLS+22] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[BM20] Adam Bohr and Kaveh Memarzadeh. Chapter 2 - the rise of artificial intelligence in healthcare applications. In Adam Bohr and Kaveh Memarzadeh, editors, *Artificial Intelligence in Healthcare*, pages 25–60. Academic Press, 2020.

[BTM+20] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque de Gusmão, et al. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*, 2020.

[BTM+22] Daniel J. Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque de Gusmão, and Nicholas D. Lane. Flower: A friendly federated learning research framework, 2022.

[BUS+22] Monik Raj Behera, Sudhir Upadhyay, Suresh Shetty, Sudha Priyadarshini, Palka Patel, and Ker Farn Lee. Fedsyn: Synthetic data generation using federated learning. *arXiv preprint arXiv:2203.05931*, 2022.

[CCL23] Yi-Hsien Chiang, Lin-Huang Chang, and Tsung-Han Lee. Federated learning for network traffic classification: Impact of non-iid distribution on model performance. In *Proceedings of the 2023 8th International Conference on Cloud Computing and Internet of Things*, pages 71–77, 2023.

[CD+15] Ronald Cramer, Ivan Bjerre Damgård, et al. *Secure multiparty computation*. Cambridge University Press, 2015.

[CGH94] David Maxwell Chickering, Dan Geiger, and David Heckerman. Learning bayesian networks is np-hard (technical report msr-tr-94-17). *Redmond, WA: Microsoft Research*, 1994.

[CH92] Gregory F Cooper and Edward Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9:309–347, 1992.

[CHM04] Max Chickering, David Heckerman, and Chris Meek. Large-sample learning of bayesian networks is np-hard. *Journal of Machine Learning Research*, 5:1287–1330, 2004.

[CQZ+20] Qi Chang, Hui Qu, Yikai Zhang, Mert Sabuncu, Chao Chen, Tong Zhang, and Dimitris N Metaxas. Synthetic learning: Learn from distributed asynchronized discriminator gan without sharing medical image data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13856–13866, 2020.

[Cra99] Harald Cramér. *Mathematical methods of statistics*, volume 26. Princeton university press, 1999.

[CTM+22] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, Mukund Lahoti, and Pratik Narang. Tabsyndex: A universal metric for robust evaluation of synthetic tabular data. *arXiv preprint arXiv:2207.05295*, 2022.

[CTS+19] Xiang Cheng, Peng Tang, Sen Su, Rui Chen, Zequn Wu, and Binyuan Zhu. Multi-party high-dimensional data publishing under differential privacy. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1557–1571, 2019.

[DBR07] Jörg Drechsler, Stefan Bender, and Susanne Rässler. Comparing fully and partially synthetic data sets for statistical disclosure control in the german iab establishment panel: supporting paper für die work session on data confdentiality 2007 in manchester. *EUNECE/Programmes*, 2007.

[DC19] Jessamyn Dahmen and Diane Cook. Synsys: A synthetic data generation system for healthcare applications. *Sensors*, 19(5):1181, 2019.

[DCF06] Luis M De Campos and Nir Friedman. A scoring function for learning bayesian networks based on mutual information and conditional independence tests. *Journal of Machine Learning Research*, 7(10), 2006.

[DFDCK+23] Ivanoe De Falco, Antonio Della Cioppa, Tomas Koutny, Martin Ubl, Michal Krcma, Umberto Scafuri, and Ernesto Tarantino. A federated

learning-inspired evolutionary algorithm: Application to glucose prediction. *Sensors*, 23(6):2957, 2023.

[DI21] Fida K Dankar and Mahmoud Ibrahim. Fake it till you make it: Guidelines for effective synthetic data generation. *Applied Sciences*, 11(5):2158, 2021.

[DI22] Fida K Dankar and Mahmoud K Ibrahim. A new pca-based utility measure for synthetic data evaluation. *arXiv preprint arXiv:2212.05595*, 2022.

[Dig18] Vassilis V. Digalakis. Data analytics with differential privacy. 2018.

[DII22] Fida K Dankar, Mahmoud K Ibrahim, and Leila Ismail. A multi-dimensional evaluation of synthetic data generators. *IEEE Access*, 10:11147–11158, 2022.

[DLH+23] Shaoming Duan, Chuanyi Liu, Peiyi Han, Xiaopeng Jin, Xinyi Zhang, Tianyu He, Hezhong Pan, and Xiayu Xiang. Ht-fed-gan: Federated generative model for decentralized tabular data synthesis. *Entropy*, 25(1):88, 2023.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

[Doe16] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[DR+14] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[EEMFEH22] Khaled El Emam, Lucy Mosquera, Xi Fang, and Alaa El-Hussuna. Utility metrics for evaluating synthetic health data generation methods: validation study. *JMIR medical informatics*, 10(4):e35734, 2022.

[EEMH20] Khaled El Emam, Lucy Mosquera, and Richard Hoptroff. *Practical synthetic data generation: balancing privacy and the broad availability of data.* O'Reilly Media, 2020.

[EF23] Erica Espinosa and Alvaro Figueira. On the quality of synthetic generated tabular data. *Mathematics*, 11(15):3278, 2023.

[EW22] Jan Ehrhardt and Matthias Wilms. Autoencoders and variational autoencoders in medical image analysis. In *Biomedical Image Synthesis and Simulation*, pages 129–162. Elsevier, 2022.

[FB23] Joao Fonseca and Fernando Bacao. Tabular and latent space synthetic data generation: a literature review. *Journal of Big Data*, 10(1):115, 2023.

[FDK22] Mei Ling Fang, Devendra Singh Dhami, and Kristian Kersting. Dp-ctgan: Differentially private medical data generation using ctgans. In *International Conference on Artificial Intelligence in Medicine*, pages 178–188. Springer, 2022.

[FTPR+21] David Froelicher, Juan R Troncoso-Pastoriza, Jean Louis Raisaro, Michel A Cuendet, Joao Sa Sousa, Hyunghoon Cho, Bonnie Berger, Jacques Fellay, and Jean-Pierre Hubaux. Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption. *Nature communications*, 12(1):5910, 2021.

[FV22] Alvaro Figueira and Bruno Vaz. Survey on synthetic data generation, evaluation methods and gans. *Mathematics*, 10(15):2733, 2022.

[FZM+23] Wei Fang, Weijian Zhang, Li Ma, Yunlin Wu, Kefei Yan, Hengyang Lu, Jun Sun, Xiaojun Wu, and Bo Yuan. An efficient bayesian network structure learning algorithm based on structural information. *Swarm and Evolutionary Computation*, 76:101224, 2023.

[GG14] Samuel Gershman and Noah Goodman. Amortized inference in probabilistic reasoning. In *Proceedings of the annual meeting of the cognitive science society*, volume 36, 2014.

[GPAM+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[GRS+20] Andre Goncalves, Priyadip Ray, Braden Soper, Jennifer Stevens, Linda Coyle, and Ana Paula Sales. Generation and evaluation of synthetic patient data. *BMC medical research methodology*, 20:1–40, 2020.

[GX15] Slawomir Goryczka and Li Xiong. A comprehensive comparison of multiparty secure additions with differential privacy. *IEEE transactions on dependable and secure computing*, 14(5):463–477, 2015.

[GYH+22] Pengfei Guo, Dong Yang, Ali Hatamizadeh, An Xu, Ziyue Xu, Wenqi Li, Can Zhao, Daguang Xu, Stephanie Harmon, Evrim Turkbey, et al. Auto-fedrl: Federated hyperparameter optimization for multi-institutional medical image segmentation. In *European Conference on Computer Vision*, pages 437–455. Springer, 2022.

[HAPC17] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 603–618, 2017.

[HEA+22] Mikel Hernandez, Gorka Epelde, Ane Alberdi, Rodrigo Cilla, and Debbie Rankin. Synthetic data generation for tabular health records: A systematic review. *Neurocomputing*, 493:28–45, 2022.

[HEA+23] Mikel Hernadez, Gorka Epelde, Ane Alberdi, Rodrigo Cilla, and Debbie Rankin. Synthetic tabular data evaluation in the health domain covering resemblance, utility, and privacy dimensions. *Methods of Information in Medicine*, 2023.

[HEM19] Markus Hittmeir, Andreas Ekelhart, and Rudolf Mayer. On the utility of synthetic data: An empirical evaluation on machine learning tasks. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pages 1–6, 2019.

[HGC95] David Heckerman, Dan Geiger, and David M Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20:197–243, 1995.

[HL05] Jin Huang and Charles X Ling. Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on knowledge and Data Engineering*, 17(3):299–310, 2005.

[HME20] Markus Hittmeir, Rudolf Mayer, and Andreas Ekelhart. A baseline for attribute disclosure risk in synthetic data. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, pages 133–143, 2020.

[HME22] Markus Hittmeir, Rudolf Mayer, and Andreas Ekelhart. Efficient bayesian network construction for increased privacy on synthetic data. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 5721–5730. IEEE, 2022.

[Hua05] Jonathan Huang. Maximum likelihood estimation of dirichlet distribution parameters. *CMU Technique report*, 76, 2005.

[JSH+22] James Jordon, Lukasz Szpruch, Florimond Houssiau, Mirko Bottarelli, Giovanni Cherubin, Carsten Maple, Samuel N Cohen, and Adrian Weller. Synthetic data–what, why and how? *arXiv preprint arXiv:2205.03257*, 2022.

[JSP+22] Saksham Jain, Gautam Seth, Arpit Paruthi, Umang Soni, and Girish Kumar. Synthetic data augmentation for surface defect detection and

classification using deep learning. *Journal of Intelligent Manufacturing*, pages 1–14, 2022.

[JXM15] Zhiwei Ji, Qibiao Xia, and Guanmin Meng. A review of parameter learning methods in bayesian network. In *Advanced Intelligent Computing Theories and Applications: 11th International Conference, ICIC 2015, Fuzhou, China, August 20-23, 2015. Proceedings, Part III 11*, pages 3–12. Springer, 2015.

[Kag18] Kaggle. Cardiovascular disease dataset, 2018. Accessed: 2024-05-03.

[KC07] Barbara Kitchenham and Stuart Charters. Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, 2007.

[KCG+23] Neville Kenneth Kitson, Anthony C Constantinou, Zhigao Guo, Yang Liu, and Kiattikun Chobtham. A survey of bayesian network structure learning. *Artificial Intelligence Review*, pages 1–94, 2023.

[KF09] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[KFE17] Konstantin G Kogos, Kseniia S Filippova, and Anna V Epishkina. Fully homomorphic encryption schemes: The state of the art. In *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pages 463–466. IEEE, 2017.

[KHK21] Asif Khan, Hyunho Hwang, and Heung Soo Kim. Synthetic data augmentation and deep learning for the fault diagnosis of rotating machines. *Mathematics*, 9(18):2336, 2021.

[KKH18] Yuta Kawachi, Yuma Koizumi, and Noboru Harada. Complementary set variational autoencoder for supervised anomaly detection. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2366–2370. IEEE, 2018.

[KMA+21] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[KRKR20] Swanand Kadhe, Nived Rajaraman, O Ozan Koyluoglu, and Kannan Ramchandran. Fastsecagg: Scalable secure aggregation for privacy-preserving federated learning. *arXiv preprint arXiv:2009.11248*, 2020.

[KS05] Lea Kissner and Dawn Song. Privacy-preserving set operations. In *Annual International Cryptology Conference*, pages 241–257. Springer, 2005.

[KSP+21] Dhamanpreet Kaur, Matthew Sobiesk, Shubham Patil, Jin Liu, Puran Bhagat, Amar Gupta, and Natasha Markuzon. Application of bayesian networks to generate synthetic health data. *Journal of the American Medical Informatics Association*, 28(4):801–811, 2021.

[KVH+21] Brian Knott, Shobha Venkataraman, Awni Hannun, Shubho Sengupta, Mark Ibrahim, and Laurens van der Maaten. Crypten: Secure multi-party computation meets machine learning. *Advances in Neural Information Processing Systems*, 34:4961–4973, 2021.

[KW13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[KW+19] Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[LC11] Jaewoo Lee and Chris Clifton. How much is enough? choosing ε for differential privacy. In *Information Security: 14th International Conference, ISC 2011, Xi'an, China, October 26-29, 2011. Proceedings 14*, pages 325–340. Springer, 2011.

[LDCH22] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 965–978. IEEE, 2022.

[LdGBL21] Kwing Hei Li, Pedro Porto Buarque de Gusmão, Daniel J Beutel, and Nicholas D Lane. Secure aggregation for federated learning in flower. In *Proceedings of the 2nd ACM International Workshop on Distributed Machine Learning*, pages 8–14, 2021.

[LDQ+22] Haodong Lu, Miao Du, Kai Qian, Xiaoming He, and Kun Wang. Gan-based data augmentation strategy for sensor anomaly detection in industrial robots. *IEEE Sensors Journal*, 22(18):17464–17474, 2022.

[LEA23] Claire Little, Mark Elliot, and Richard Allmendinger. Federated learning for generating synthetic data: a scoping review. *International Journal of Population Data Science*, 8(1), 2023.

[LFTL20] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854, 2020.

[Lit93] Roderick J. A. Little. Statistical analysis of masked data. *J. Off. Stat.*, 9:407–407, 1993.

[LLZ21] Zhiyuan Li, Hao Li, and Mingyang Zhang. Hyper-parameter tuning of federated learning based on particle swarm optimization. In *2021 IEEE 7th International Conference on Cloud Computing and Intelligent Systems (CCIS)*, pages 99–103. IEEE, 2021.

[LXW22] Pengrui Liu, Xiangrui Xu, and Wei Wang. Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives. *Cybersecurity*, 5(1):1–19, 2022.

[MAK+23] Hajra Murtaza, Musharif Ahmed, Naurin Farooq Khan, Ghulam Murtaza, Saad Zafar, and Ambreen Bano. Synthetic data generation: State of the art in health care domain. *Computer Science Review*, 48:100546, 2023.

[Mar21] Georgios Margaritis. Differentially private data synthesis using variational autoencoders. Diploma thesis, School of Electrical and Computer Engineering, Technical University of Crete, 2021.

[Mic22] Umberto Michelucci. An introduction to autoencoders. *arXiv preprint arXiv:2201.03898*, 2022.

[MMR+17] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[MMRyA16] H Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*, 2(2), 2016.

[MMSG21] Sebastian Meister, Nantwin Möller, Jan Stüve, and Roger M Groves. Synthetic image data augmentation for fibre layup inspection processes: Techniques to enhance the data set. *Journal of Intelligent Manufacturing*, 32:1767–1789, 2021.

[MNH21] David Meyer, Thomas Nagler, and Robin J Hogan. Copula-based synthetic data generation for machine learning emulators in weather and climate: application to a simple radiation model. *Geoscientific Model Development Discussions*, 2021:1–21, 2021.

[MPS24] Natalija Mitic, Apostolos Pyrgelis, and Sinem Sav. How to privately tune hyperparameters in federated learning? insights from a benchmark study. *arXiv preprint arXiv:2402.16087*, 2024.

[MRC12] S. Moro, P. Rita, and P. Cortez. Bank marketing. UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C5K306.

[MS06] Jianjie Ma and Krishnamoorthy Sivakumar. Privacy-preserving bayesian network learning from heterogeneous distributed data. In *DMIN*. Citeseer, 2006.

[MSDCS19] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 691–706. IEEE, 2019.

[MSK04] Da Meng, Krishnamoorthy Sivakumar, and Hillol Kargupta. Privacy-sensitive bayesian network parameter learning. In *Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 487–490. IEEE, 2004.

[MT07] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, 2007.

[MTT+20] Chao Ma, Sebastian Tschiatschek, Richard Turner, José Miguel Hernández-Lobato, and Cheng Zhang. Vaem: a deep generative model for heterogeneous mixed type data. *Advances in Neural Information Processing Systems*, 33:11237–11247, 2020.

[NHDC22] Mohammad Naseri, Jamie Hayes, and Emiliano De Cristofaro. Local and central differential privacy for robustness and privacy in federated learning. *Network and Distributed System Security Symposium*, 2022.

[NRD16] Beata Nowok, Gillian M Raab, and Chris Dibben. synthpop: Bespoke creation of synthetic data in r. *Journal of statistical software*, 74:1–26, 2016.

[NSU+18] Adrian Nilsson, Simon Smith, Gregor Ulm, Emil Gustavsson, and Mats Jirstrand. A performance evaluation of federated learning algorithms. In *Proceedings of the second workshop on distributed infrastructures for deep learning*, pages 1–8, 2018.

[NZ22] Ignavier Ng and Kun Zhang. Towards federated bayesian network structure learning with continuous optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 8095–8111. PMLR, 2022.

[PA22] Bjarne Pfitzner and Bert Arnrich. Dpd-fvae: Synthetic data generation using federated variational autoencoders with differentially-private decoder. *arXiv preprint arXiv:2211.11591*, 2022.

[PGMB21] Ricardo Silva Peres, Magno Guedes, Fábio Miranda, and Jose Barata. Simulation-based data augmentation for the quality inspection of structural adhesive with deep learning. *IEEE Access*, 9:76532–76541, 2021.

[PHK+23] Natalia Ponomareva, Hussein Hazimeh, Alex Kurakin, Zheng Xu, Carson Denison, H Brendan McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Guha Thakurta. How to dp-fy ml: A practical guide to machine learning with differential privacy. *Journal of Artificial Intelligence Research*, 77:1113–1201, 2023.

[PM20] Anastasia Pustozerova and Rudolf Mayer. Information leaks in federated learning. In *Proceedings of the network and distributed system security symposium*, volume 10, page 122, 2020.

[PMS+23] Aryan Pathare, Ramchandra Mangrulkar, Kartik Suvarna, Aryan Parekh, Govind Thakur, and Aruna Gawade. Comparison of tabular synthetic data generation techniques using propensity and cluster log metric. *International Journal of Information Management Data Insights*, 3(2):100177, 2023.

[PR21] Michael Platzer and Thomas Reutterer. Holdout-based fidelity and privacy assessment of mixed-type synthetic data. *arXiv preprint arXiv:2104.00635*, 2021.

[PRALP+23] Harikumar Pallathadka, Edwin Hernan Ramirez-Asis, Telmo Pablo Loli-Poma, Karthikeyan Kaliyaperumal, Randy Joy Magno Ventayen, and Mohd Naved. Applications of artificial intelligence in business management, e-commerce and finance. *Materials Today: Proceedings*, 80:2610–2613, 2023.

[Pre23] Davy Preuveneers. Autofl: Towards automl in a federated learning context. *Applied Sciences*, 13(14):8019, 2023.

[PRR10] Manas Pathak, Shantanu Rane, and Bhiksha Raj. Multiparty differential privacy via aggregation of locally trained classifiers. *Advances in neural information processing systems*, 23, 2010.

[PSA21] Bjarne Pfitzner, Nico Steckhan, and Bert Arnrich. Federated learning in a medical context: a systematic literature review. *ACM Transactions on Internet Technology (TOIT)*, 21(2):1–31, 2021.

[PSH17] Haoyue Ping, Julia Stoyanovich, and Bill Howe. Datasynthesizer: Privacy-preserving synthetic datasets. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, pages 1–5, 2017.

[PWV16] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The synthetic data vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 399–410. IEEE, 2016.

[Rei03] Jerome P Reiter. Inference for partially synthetic, public use microdata sets. *Survey Methodology*, 29(2):181–188, 2003.

[Rei05] Jerome P Reiter. Using cart to generate partially synthetic public use microdata. *Journal of official statistics*, 21(3):441, 2005.

[Rei23] Jerome P Reiter. Synthetic data: A look back and a look forward. *Trans. Data Priv.*, 16(1):15–24, 2023.

[RN10] Vibhor Rastogi and Suman Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 735–746, 2010.

[RTD+18] Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, and Jonathan Passerat-Palmbach. A generic framework for privacy preserving deep learning. *arXiv preprint arXiv:1811.04017*, 2018.

[Rub93] Donald B Rubin. Statistical disclosure limitation. *Journal of official Statistics*, 9(2):461–468, 1993.

[RV20] Ali Lotfi Rezaabad and Sriram Vishwanath. Learning representations by maximizing mutual information in variational autoencoders. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 2729–2734. IEEE, 2020.

[Sch78] Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.

[SHY+22] Jinhyun So, Chaoyang He, Chien-Sheng Yang, Songze Li, Qian Yu, Ramy E Ali, Basak Guler, and Salman Avestimehr. Lightsecagg: a lightweight and versatile design for secure aggregation in federated learning. *Proceedings of Machine Learning and Systems*, 4:694–720, 2022.

[Skl59] M Sklar. Fonctions de repartition an dimensions et leurs marges. *Publ. inst. statist. univ. Paris*, 8:229–231, 1959.

[SM09] Saeed Samet and Ali Miri. Privacy-preserving bayesian network for horizontally partitioned data. In *2009 International Conference on Computational Science and Engineering*, volume 3, pages 9–16. IEEE, 2009.

[SM17] HMHS Surendra and HS Mohan. A review of synthetic data generation methods for privacy-preserving data publishing. *International Journal of Scientific & Technology Research*, 6(3):95–101, 2017.

[SOT20] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. Synthetic data-a privacy mirage. *arXiv preprint arXiv:2011.07018*, 2020.

[SRN+18] Joshua Snoke, Gillian M Raab, Beata Nowok, Chris Dibben, and Aleksandra Slavkovic. General and specific utility measures for synthetic data. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 181(3):663–688, 2018.

[SRRW23] Jayanth Sivakumar, Karthik Ramamurthy, Menaka Radhakrishnan, and Daehan Won. Generativemtd: A deep synthetic data generation framework for small datasets. *Knowledge-Based Systems*, 280:110956, 2023.

[STC+16] Sen Su, Peng Tang, Xiang Cheng, Rui Chen, and Zequn Wu. Differentially private multi-party high-dimensional data publishing. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 205–216. IEEE, 2016.

[Stu08] Student. The probable error of a mean. *Biometrika*, pages 1–25, 1908.

[SVG18] Konstantin Sozinov, Vladimir Vlassov, and Sarunas Girdzijauskas. Human activity recognition using federated learning. In *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications*, pages 1103–1111, 2018.

[SZA22] Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

[TEPS18] Jennifer Taub, Mark Elliot, Maria Pampaka, and Duncan Smith. Differential correct attribution probability for synthetic data: an exploration. In *Privacy in Statistical Databases: UNESCO Chair in Data Privacy, International Conference, PSD 2018, Valencia, Spain, September 26–28, 2018, Proceedings*, pages 122–137. Springer, 2018.

[TF20] Aleksei Triastcyn and Boi Faltings. Federated generative privacy. *IEEE Intelligent Systems*, 35(4):50–57, 2020.

[Tha20] Alaa Tharwat. Classification assessment methods. *Applied computing and informatics*, 17(1):168–192, 2020.

[VDIDB22] Florian Van Daalen, Lianne Ippel, Andre Dekker, and Inigo Bermejo. Vertibayes: Learning bayesian network parameters from vertically partitioned data with missing values. *arXiv preprint arXiv:2210.17228*, 2022.

[WH00] Rüdiger Wirth and Jochen Hipp. Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, volume 1, pages 29–39. Manchester, 2000.

[WPL+23] Jinhong Wu, Konstantinos Plataniotis, Lucy Liu, Ehsan Amjadian, and Yuri Lawryshyn. Interpretation for variational autoencoder used to generate financial synthetic tabular data. *Algorithms*, 16(2):121, 2023.

[WWB21] John Weldon, Tomas Ward, and Eoin Brophy. Generation of synthetic electronic health records using a federated gan. *arXiv preprint arXiv:2109.02543*, 2021.

[WY04] Rebecca Wright and Zhiqiang Yang. Privacy-preserving bayesian network structure computation on distributed heterogeneous data. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 713–718, 2004.

[WZL+23] Jie Wen, Zhixia Zhang, Yang Lan, Zhihua Cui, Jianghui Cai, and Wensheng Zhang. A survey on federated learning: challenges and applications. *International Journal of Machine Learning and Cybernetics*, 14(2):513–535, 2023.

[XSCIV19] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems*, 32, 2019.

[Yao82] Andrew C Yao. Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)*, pages 160–164. IEEE, 1982.

[YGP09] Jim Young, Patrick Graham, and Richard Penny. Using bayesian networks to create synthetic data. *Journal of Official Statistics*, 25(4):549–567, 2009.

[YSZ+22] Dayong Ye, Sheng Shen, Tianqing Zhu, Bo Liu, and Wanlei Zhou. One parameter defense—defending against data inference attacks via differential privacy. *IEEE Transactions on Information Forensics and Security*, 17:1466–1480, 2022.

[YW06] Zhiqiang Yang and Rebecca N Wright. Privacy-preserving computation of bayesian networks on vertically partitioned data. *IEEE Transactions on Knowledge and Data Engineering*, 18(9):1253–1264, 2006.

[ZBKC21] Zilong Zhao, Robert Birke, Aditya Kunar, and Lydia Y Chen. Fed-tgan: Federated learning framework for synthesizing tabular data. *arXiv preprint arXiv:2108.07927*, 2021.

[ZCP+17] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. 42(4), oct 2017.

[ZCZ15] Yuan Zhang, Qingjun Chen, and Sheng Zhong. Efficient and privacy-preserving min and k th min computations in mobile sensing systems. *IEEE Transactions on Dependable and Secure Computing*, 14(1):9–21, 2015.

[ZFZ+23] Huanle Zhang, Lei Fu, Mi Zhang, Pengfei Hu, Xiuzhen Cheng, Prasant Mohapatra, and Xin Liu. Federated learning hyper-parameter tuning from a system perspective. *IEEE Internet of Things Journal*, 2023.

[ZLH+23] Dun Zeng, Siqi Liang, Xiangjing Hu, Hui Wang, and Zenglin Xu. Fedlab: A flexible federated learning framework. *Journal of Machine Learning Research*, 24(100):1–7, 2023.

[ZLL23] Yi Zhang, Yunfan Lu, and Fengxia Liu. A systematic survey for differential privacy techniques in federated learning. *Journal of Information Security*, 14(2):111–135, 2023.

[ZLZP17] Tianqing Zhu, Gang Li, Wanlei Zhou, and S Yu Philip. Differentially private data publishing and analysis: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 29(8):1619–1638, 2017.

[ZXB+21] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li, and Yuan Gao. A survey on federated learning. *Knowledge-Based Systems*, 216:106775, 2021.

[ZZZ+19] Chuan Zhao, Shengnan Zhao, Minghao Zhao, Zhenxiang Chen, Chong-Zhi Gao, Hongwei Li, and Yu-an Tan. Secure multi-party computation: theory, practice and applications. *Information Sciences*, 476:357–372, 2019.