

Sébastien Gambs, Frédéric Ladouceur, Antoine Laurent, and Alexandre Roy-Gaumond*

Growing synthetic data through differentially-private vine copulas

Abstract: In this work, we propose a novel approach for the synthetization of data based on copulas, which are interpretable and robust models, extensively used in the actuarial domain. More precisely, our method COPULA-SHIRLEY is based on the differentially-private training of vine copulas, which are a family of copulas allowing to model and generate data of arbitrary dimensions. The framework of COPULA-SHIRLEY is simple yet flexible, as it can be applied to many types of data while preserving the utility as demonstrated by experiments conducted on real datasets. We also evaluate the protection level of our data synthesis method through a membership inference attack recently proposed in the literature.

Keywords: Synthetic data, Copulas, Differential privacy, Privacy evaluation.

DOI 10.2478/popets-2021-0040

Received 2020-11-30; revised 2021-03-15; accepted 2021-03-16.

1 Introduction

With the advent of Big Data and the widespread development of machine learning, the sharing of datasets has become a crucial part of the knowledge discovery process. For instance, Kaggle¹, a major actor in the data portal platforms, supports more than 19 000 public datasets and the University of California at Irvine (UCI)² repository is another important data portal with more 550 curated datasets that can be used for machine learning purposes. Numerous public institutions also publicly share datasets such as the Canadian open

data portal³ and the American data portal⁴, respectively each with 83 460 and 209 765 datasets. The open data movement has also led cities to open more and more data. Concrete examples are the cities of Montréal⁵ and Toronto⁶, with around 350 datasets each.

The major drawback of this augmentation in the availability of data is that it is accompanied by the increase in privacy risks for the individuals whose records are contained in the shared data. Such privacy risks have been demonstrated many times in the literature, even in the situation in which the data was supposed to be anonymized. For instance, in the context of mobility data, researchers have shown that any individual is unique in a population with respect to his mobility behaviour simply given three to five points of interests (*i.e.*, frequently visited locations) [17]. Following the same approach but in the context of financial data, another study has demonstrated that five credit card transactions are sufficient to uniquely pinpoint an individual [18]. Finally, even data that appears to be “harmless” at first sight such as movie ratings could lead to privacy risks as illustrated by Narayanan and Shmatikov [48, 59].

To address these issues, multiple anonymization techniques have been developed in the last two decades. Among these, there has recently been a growing interest in generative models and synthetic data. A key property of generative methods is their ability to mince even more the link between identities and the data shared in comparison with other anonymization techniques. This property is due to the summarization of the data to probabilistic representations that can be sampled to produce new profiles that are not associated with a particular identity. Nonetheless, generative models as well as the synthetic data they produce, can still leak information about the dataset used to train the generative model and thus lead to privacy breaches. In particular, a recent study by Stadler, Oprisanu and Troncoso [62] have demonstrated that generative models trained in

Sébastien Gambs: UQAM, E-mail: gambs.sebastien@uqam.ca

Frédéric Ladouceur: Ericsson Montréal, E-mail: ladouceu@stanford.edu

Antoine Laurent: UQAM, E-mail: laurent.antoine@courrier.uqam.ca

***Corresponding Author: Alexandre Roy-Gaumond:** UQAM, E-mail: roy-gaumond.alexandre@courrier.uqam.ca

¹ <https://www.kaggle.com/datasets>

² <https://archive.ics.uci.edu/ml/datasets.php>

³ <https://open.canada.ca/en/open-data>

⁴ <https://www.data.gov/>

⁵ <https://donnees.ville.montreal.qc.ca/>

⁶ <https://open.toronto.ca/>

a non-private manner provide little protection against inference attacks compared to simply publishing the original data. They also showed that generative models trained in a differentially-private manner did not improve the protection.

Thus, in addition to being based on a formal model such as differential privacy, we believe that data synthesis methods should be assessed with a privacy evaluation based on inference attacks to further quantify the privacy protection provided by the synthesis method. In this paper, we propose a copula-based approach to differentially-private data synthesis. Our approach aims at providing an answer to the following conundrum: *How to generate synthetic data that is private yet realistic with respect to the original data?*

The proposed generative model is called COPULA-SHIRLEY, which stands for COPULA-based generation of SyntHetIc diffeRentialL(E)Y-private data. In a nutshell, COPULA-SHIRLEY constructs a differentially-private vine copulas model that privately summarizes the training data and can later be used to generate synthetic data. We believe that our method has the following strengths:

- *Simple to use and deploy* - due to the nature of the mathematical objects upon which it is based, which are mainly density functions.
- *Flexible* - due to the highly customizable framework of copulas and their inherent ability to model even complex distributions by decomposing them into a combination of bivariate copulas.
- *Private* - due to the use of differential privacy to train the generative model as well as the privacy test for membership inference.

The outline of the paper is as follows. First, we review the literature on differentially-private data synthesis methods in Section 2 before presenting the background notions of differential privacy and copulas theory in Section 3. Afterwards, we describe in detail in Section 4 the synthesis based vine copulas that we propose COPULA-SHIRLEY, followed by an experimental evaluation of the utility-privacy trade-off that it can achieve on three real datasets in Section 5 before concluding.

2 Related work on private data synthesis

Although the literature on anonymization methods is huge, hereafter we focus on the methods for synthesiz-

ing data in a private manner. Most of these methods rely on generative models that condense the information in probabilistic models, thus severing the link between identity and data through randomness and abstraction. Common generative models include Bayesian networks [32], Markov models [63] and neural networks [4]. It is possible to roughly distinguish between two types of synthesis methods: partial and complete.

Partial data synthesis. Partial synthesis implies that for a given profile, some attributes are fixed while the rest is sampled by using the generative model. As a consequence partial synthesis requires both the access to part of the original profiles *and* the trained generative model.

A seminal work on partial data synthesis is done by Bindschadler, Shokri and Gunter in [11], in which they develop a framework based on plausible deniability and differential privacy. The generative model used is based on differentially private Bayesian networks capturing the joint probability distribution of the attributes. In a nutshell, the “seed-based” synthesis of new records takes a record from the original dataset and “updates” a random subset of its attributes through the model. Their approach releases a record only if it passes the privacy test of being similar to at least k records from the original dataset, hence the plausible deniability. A notable drawback of this approach is the high computational time and the large addition of noise, which makes it difficult to scale to high-dimensional datasets as pointed out in [15].

In contrast, complete synthesis methods output new profiles directly from the generative model. Hereafter, we focus on complete data synthesis methods.

Statistical-based generation. The PrivBayes approach [77] has become one of the standards in the literature for complete data synthesis through Bayesian networks. PrivBayes integrates differential privacy in each step of the construction of the Bayesian network while optimizing the privacy budget. More precisely, the construction applies the Exponential mechanism on the mutual information between pairs of attributes and the Laplace mechanism to protect the conditional distributions [22]. Moreover, the privacy of the method relies solely on the use of differential privacy and no inference attacks are provided. In addition, the greedy construction of the Bayesian network is time consuming, which makes it impractical for high-dimensional data.

Gaussian models are widely used for data modelling, independently of the properties of the data itself, as the structure of the data can be accurately estimated with only the mean and covariance of the

data. Recently Chanyaswad, Liu and Mittal have proposed a differentially-private method to generate high-dimensional private data through the use of random orthonormal projection (RON) and Gaussian models named RON-Gauss [15]. RON is a dimensionality reduction technique that lowers the amount of noise added to achieve differential privacy and achieves the Diaconis-Freedman-Meckes (DFM) effect stating that “*under suitable conditions, most projections are approximately Gaussian*”. With the DFM effect, Gaussian generative models are then well suited to model the projected data and are easily made differentially private. The authors have implemented the membership inference attack described in [58] as a privacy test and shown that with a privacy budget $\epsilon < 1.5$ the success of the attack is no better than a random guess.

Copula-based generation. Comparable works to our use copula functions as generative models [6, 36]. To the best of our knowledge, the first differentially-private synthesis method based on copulas is by Li, Xiong and Jiang and is called DP-COPULA [36]. The authors have used copulas to model the joint distributions based on marginal distributions. They use differentially-private histograms as marginal distributions and the Gaussian copula to estimate the joint distribution. No privacy tests on their differentially-private synthetic data was considered by the authors and much like RON-Gauss [15], they simplify the structure to a Gaussian model. However, some authors have shown that some tail dependencies are not fully captured by Gaussian models, which can lead to an important loss of information [38]. In contrast, as described in Section 3.2, vine copulas, which are the basis of COPULA-SHIRLEY, split the multivariate function into multiple bivariate functions enabling the bivariate families of copulas to successfully capture the dependence structure.

Tree-based generation. A vine copula is a structure with nested trees, which can thus be viewed as a tree-based generation model (TGM). Other PGMs for synthetic data generation such as Reiter’s work [55] and [14] rely on trees to model the conditional probabilities between a variable Y and some other predictor variables X_i , in which i varies from 1 to n , the number of attributes. More precisely, each leaf corresponds to a conditional distribution of Y given $a_i \leq X_i \leq b_i$. Such an approach can be classified as partial data synthesis due to the use of the predictor variables to form the imputed synthetic Y . In comparison, the nested trees of vine copulas have the advantages to provide fully synthetic data as well as deeper modelling of conditional distributions due to a more flexible representation.

GAN-based generation. Since the influential work of Goodfellow and collaborators [28] on Generative Adversarial Networks (GANs), this approach has been leveraged on for private data generation [24, 34, 67, 70, 75] to name a few. For instance, the work of Jordan, Yoon and van der Schaar has introduced PATE-GAN [34], which combines two previous differentially-private machine learning techniques of machine learning, namely Private Aggregation of Teacher Ensembles (PATE) [50] and Differentially Private Generative Adversarial Network (DPGAN) [74]. The combination of these two is done by using a “student” layer that learns from the differentially-private classifier PATE as the discriminator in the DPGAN framework, thus enabling the training of both a differentially-private generator and discriminator. The work of [34] advocates the privacy of their synthetic records because of the use of differential privacy but do not use any privacy tests to validate it. The main disadvantages of using techniques such as GANs are that they need a large quantity of data, a fine-grained tuning of their hyper parameters and they have a high complexity, which makes them inappropriate for a number of real-life applications.

Privacy evaluation. The amount of data synthesis methods based on generative models trained in a differentially private manner as grown significantly in recent years. However, a fundamental question to solve when deploying these models in practice is what value of ϵ to choose to provide an efficient protection. While there is currently no consensus on the “right value” of ϵ , often researchers set $\epsilon \in (0, 2]$ but it can sometimes be greater than $\epsilon > 10^5$ [37, 46, 50]. Moreover, a recent study has shown the importance of providing a privacy assessment even when the model is trained in a differential privacy manner [62]. In particular, the authors have demonstrated that differential privacy in model training does not provide uniform privacy protection and that some models fail to provide any protection at all. Thus, while achieving differential privacy is a first step for data synthesis, we believe that it is also important to complement them with methods that can be used to evaluate the remaining privacy risks of sharing the synthetic data.

To the best of our knowledge, very few works on data synthesis achieving differential privacy have also proposed methods to assess the remaining privacy risks related to synthetic data with the exception of [11], which assesses the plausible deniability level of the generated data. Privacy can be assessed in multiple ways. A common approach when using differential privacy, which mitigates the risk of membership disclosure risk

by diminishing the contribution of any individual in the model learned, is testing via membership inference attacks. A recent paper [30] proposed a model-agnostic method to assess the membership disclosure risk by quantifying the ease of distinguishing training records from other records drawn from a disjoint set with the same distribution.

In the machine learning context, the approach recently developed by Meehan, Chaudhuri and Dasgupta [43] tackled the data-copying issue that can arise from generative models. Data-copying is defined as a form of overfitting in which trained models output identical or very similar profiles to the training samples. The proposed framework focuses on global and local detection of data-copying by measuring the distribution of distances between the synthetic profiles and their nearest neighbours in the training set as well as in a test set. The main idea here is that a smaller distance between the synthetic profiles and the training ones compared to the synthetic to test ones is an indication that the data generation process has a tendency to exhibit data-copying.

3 Preliminaries

In this section, we review the background notions necessary to the understanding of our work, namely differential privacy as the privacy model upon which our method is built as well as copulas and vine copulas as the underlying generative model.

3.1 Differential privacy

Differential Privacy (DP) is a privacy model that aims at providing strong privacy guarantees with respect to the amount of information that can leak about individuals that are part of a database [22]. In a nutshell, DP ensures that the contribution of a particular profile will have a limited impact on the outcome of a computation run on the input database. This property is achieved by ensuring (*e.g.*, through the addition of noise to the output or by randomizing the computation) that the distribution of outputs of the computation or the query is “almost the same” whether or not the profile of an individual was in the input database.

For our framework, we use the bounded definition of differential privacy. The following definitions are

based on the notation used in the book of Dwork and Roth [22].

Definition 3.1 (Differential privacy (bounded)). *Let x and y two databases of equal size such that they differ at most by one record. A randomized mechanism \mathcal{M} is ϵ -differentially private such that for the space of possible outputs $\mathcal{S} \subseteq \mathfrak{S}(\mathcal{M})$ we have:*

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\epsilon) \cdot \Pr[\mathcal{M}(y) \in \mathcal{S}].$$

This property ensures that the difference in the outputs of the randomized mechanism is indistinguishable up to $\exp(\epsilon)$ for two databases that differ in at most one profile. Here, ϵ is the parameter defining the level of privacy. In particular, the smaller is the value of ϵ , the higher is the protection offered due to the increase in indistinguishability.

The (global) *sensitivity* Δ of a function f is an important notion in differential privacy formalizing the contribution of a particular individual on the output of this function. More precisely, the sensibility characterizes the answer to the following question: “What is the maximal contribution that an individual can have on the outcome of a particular function f ?”.

For an arbitrary function f , there exists many ways to make it ϵ -differentially private. For instance, the Laplacian mechanism [22] can be used to make a numerical function returning a real or a set of reals $f : \mathcal{D} \rightarrow \mathbb{R}^k$ ϵ -differentially private by drawing randomly from the Laplacian distribution $\text{Lap}(\frac{\Delta f}{\epsilon})$.

Another important concept in differential privacy is the notion of the *privacy budget*, which bounds the total amount of information leaked about the input database when applying several mechanisms on this database. The application of several mechanisms do not always have the same impact on the privacy budget depending on whether these mechanisms are applied in a sequential or parallel manner. The following theorems are inherent properties of differential privacy and help to adequately manage the privacy budget.

In the following, \mathcal{R} and \mathcal{R}' are arbitrary sets, which refer to the image of the functions defined in the theorems.

Theorem 3.1 (Closure under post-processing [22]).

Let $\mathcal{M} : \mathcal{D} \rightarrow \mathcal{R}$ be a ϵ -differentially private randomized mechanism and $f : \mathcal{R} \rightarrow \mathcal{R}'$ be an arbitrary function, independent of \mathcal{D} , then the composition $f \circ \mathcal{M} : \mathcal{D} \rightarrow \mathcal{R}'$ is also ϵ -differentially private.

A subtle but important point of Theorem 3.1 of the closure under post-processing is that the function f has to *independent of \mathcal{D}* , which means that it should not have access to or use any information about \mathcal{D} . If this is not the case, the theorem does not apply anymore.

Theorem 3.2 (Sequential composition [22]). *Let $\mathcal{M}_1 : \mathcal{D} \rightarrow \mathcal{R}$ and $\mathcal{M}_2 : \mathcal{D} \rightarrow \mathcal{R}'$ be two randomized mechanisms that are respectively ϵ_1 and ϵ_2 -differentially private, then the composition $(\mathcal{M}_1, \mathcal{M}_2)(x) : \mathcal{D} \rightarrow \mathcal{R} \times \mathcal{R}'$ is $(\epsilon_1 + \epsilon_2)$ -differentially private.*

Theorem 3.3 (Parallel composition [22]). *Let $\mathcal{M}_1 : \mathcal{D}_1 \rightarrow \mathcal{R}$ and $\mathcal{M}_2 : \mathcal{D}_2 \rightarrow \mathcal{R}'$ be two randomized mechanisms that are respectively ϵ_1 and ϵ_2 -differentially private, such that $\mathcal{D}_1, \mathcal{D}_2 \subset \mathcal{D}$ and $\mathcal{D}_1 \cap \mathcal{D}_2 = \emptyset$, then the composition $(\mathcal{M}_1, \mathcal{M}_2)(x) : \mathcal{D} \rightarrow \mathcal{R} \times \mathcal{R}'$ is $\max(\epsilon_1, \epsilon_2)$ -differentially private.*

Theorems 3.2 and 3.3 can easily generalized to k sequential mechanisms or disjoint subsets.

3.2 Copulas and vine copulas

Copulas. Historically, copulas date back from 1959 with Abe Sklar that has introduced this notion in his seminal paper [60] at which time they were mainly theoretical statistical tools. However, in the last decade they have experienced a significant growth in many domains including earth science for modelling atmospheric precipitation [8], in health for diagnostic tests [31], in finances for the study of financial time series [51], in social sciences for modelling the different usages of a technology [35], in genetics for the study of phenotypes [29] and even more recently in privacy for estimating the risk of re-identification [56].

In a nutshell, a copula is a multivariate Cumulative Density Function (CDF) on the unit cube with the marginal distributions (or simply marginals) on its axes modelling the dependence between the said marginals. The formal definition of a copula is as follows:

Definition 3.2 (Copula [49]). *Let (X_1, X_2, \dots, X_n) be a random vector such as the associated cumulative density functions F_{X_i} are continuous. The copula of (X_1, X_2, \dots, X_n) , denoted by $C(U_1, U_2, \dots, U_n)$, is defined as the multivariate cumulative density function of (U_1, U_2, \dots, U_n) in which $U_i = F_{X_i}(X_i)$ are the standard uniform marginal distributions of (X_1, X_2, \dots, X_n) .*

Estimating the copula function with a known distribution family results in modelling the dependence structure of the data. Assuming a variable of interest of dimension n , the modelling by copula is generally done by first choosing a family of copulas and then verifying the “goodness-of-fit” of this modelling, in the same manner that a known distribution can be used to estimate an arbitrary distribution.

Figure 1 [27] illustrates simulations from five common bivariate copulas and the tail dependence they help to model. We refer the interested reader to [33, 49] for further reading about the many existing families of copulas and their tail dependence modelling (each of these books defines and discusses more than 20 families of copulas).

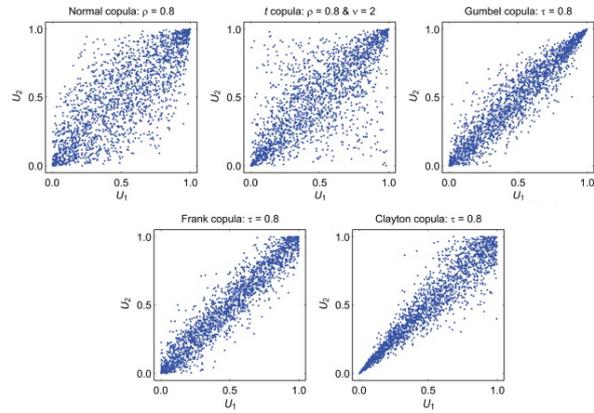


Fig. 1. Simulations from common copulas family.

Sklar’s theorem is a fundamental tool in copula theory that binds together the marginals and the copula, which means that we only need the marginals of the data to model the dependence structure.

Theorem 3.4 (Sklar’s Theorem [60]). *Let F be a multivariate cumulative distribution function with marginals F_{X_i} with $i \in \{1, 2, \dots, n\}$. Then, there exists a copula C such that $F(X_1, X_2, \dots, X_n) = C(F_{X_1}, F_{X_2}, \dots, F_{X_n})$. If all the marginals are continuous, then C is unique. Conversely, if F_{X_i} are univariate distribution functions and C is a copula, then $C(u_1, u_2, \dots, u_n) = F(F_{X_1}^{-1}(u_1), F_{X_2}^{-1}(u_2), \dots, F_{X_n}^{-1}(u_n))$ with $F_{X_i}^{-1}$ being the inverse CDF of F_{X_i} .*

Finally, a last fundamental theorem of the domain of copulas is the Probability Integral Transform (PIT) as well as the Inverse PIT.

Theorem 3.5 (PIT [19]). *Let X be a random variable with F_X as its cumulative density function (CDF). The variable $Y = F_X(X)$ is uniform standard.*

Theorem 3.6 (Inverse PIT [19]). *Let U be a uniform standard random variable and F an arbitrary CDF, then $Y = F^{-1}(U)$ is a random variable with F as its CDF.*

The PIT and its inverse are used to transform data into uniform standard distribution for the copulas to model and back into the original domain after the generation of new observations.

When applied on experimental data, uniform standard marginals take the form of pseudo-observations (*i.e.*, normalized ranked data) that are computed via the PIT. The pseudo-observations are then used to find the best fitted copula model. Figure 1 illustrates such an example in which data points (*i.e.*, pseudo-observations) are shown to best fit various copula models. The vine copulas described hereafter offer a more refined manner to describe the dependency structure.

Vine copulas. Vine copulas were introduced by Bedford and Cooke in the early 2000s to refine the modelling capabilities of copulas [9, 10]. The principle behind them is simple: they decompose the multivariate functions of the copula into multiple “couples” of copulas. A vine on n variables is a set of nested connected trees (T_1, T_2, \dots, T_n) such that edges of a tree T_i corresponds to vertexes in the following tree T_{i+1} . *Regular vines* refer to a subset of vines constructed in a more constrained way. As we only use regular vines in this paper, the term “vines” will refer to regular vines hereafter.

Definition 3.3 (Regular vine [9]). *A regular vine on n variables contains $n - 1$ connected nested trees denoted T_1, T_2, \dots, T_{n-1} . Let N_i and E_i be the set of nodes and edges of the tree T_i . These trees satisfy the following properties:*

1. T_1 is the tree that contains n nodes corresponding to the n variables.
2. For $i \in \{2, 3, \dots, n - 1\}$, the nodes set of T_i is such as $N_i = E_{i-1}$.
3. For $i \in \{2, 3, \dots, n - 1\}$, $\{a, b\} \in E_i$ with $a = (a_1, a_2)$ and $b = (b_1, b_2)$, it must hold that $|a \cap b| = 1$.

Condition 3. is called the *proximity condition* and can be translated to two nodes in tree T_i are connected by an edge if and only if these nodes, as edges, share a common node in tree T_{i-1} .

With this definition, each edge $e \in E_i$ is associated to (the density of) a bivariate copula $c_{j_e, k_e | D_e}$, in which j_e, k_e are the conditioned variables over the conditioning variables set D_e . Thus, the conditioned variables and the conditioning set form the conditional variables $U_{j_e | D_e}$ and $U_{k_e | D_e}$. The existence of such conditional variables is guaranteed by the conditions on the trees $T_i, i \in \{1, 2, \dots, n - 1\}$ [9, 20].

To help visualize what is a vine, Figure 2 [1] shows an example of a vine on 5 variables. In this example, on the tree T_4 , the conditioned variables are 1 and 5 and the conditioning set is $\{2, 3, 4\}$ forming the variables $U_{1|2,3,4}$ and $U_{5|2,3,4}$.

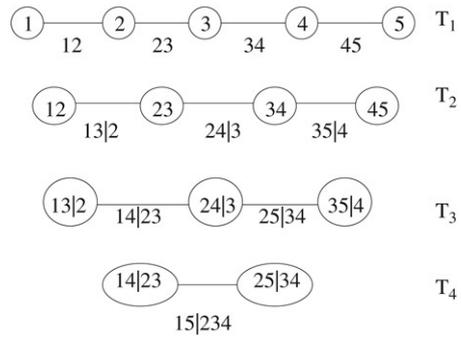


Fig. 2. Example of a vine on 5 variables.

From a vine V , it is possible to describe a multivariate density function as the product of bivariate copulas. The following theorem implies that the joint density of a multivariate random variables can always be estimated via vine copulas.

Theorem 3.7 (Vine density decomposition [9]). *Let V be a vine on n variables, then there is a unique density function f such as*

$$f(X_1, X_2, \dots, X_n) = \prod_{i=1}^n f_i(X_i) \prod_{m=1}^{n-1} \prod_{e \in E_m} c_{j_e, k_e | D_e}(U_{j_e | D_e}, U_{k_e | D_e})$$

in which $U_{j_e | D_e} = F_{j_e | D_e}(X_{j_e | D_e})$ and $U_{k_e | D_e} = F_{k_e | D_e}(X_{k_e | D_e})$.

Formal and thorough definitions of vine copulas are available in [9, 10, 20].

The construction of a vine is non-trivial as there are approximately $\frac{n!}{2} \cdot 2^{\binom{n-2}{2}}$ different vines on n variables. The current state-of-the-art technique of construction

uses a top-down greedy heuristic known as the Dissmann’s algorithm [20].

Dissmann’s algorithm for vine copula selection. The technique known as the Dissmann’s algorithm [20] is based on the hypothesis that the top of the tree is more important in modelling the data than the lower branches as they contain raw information about the correlation between attributes. Dissmann’s algorithm can be summarized as:

1. Choose the optimal tree that maximizes the sum of correlation coefficients between variables.
2. Choose the optimal copula family for each pair of attributes (imposed by the tree) that minimizes the information loss.
3. Construct all the following trees (from the previous optimal tree) that respect the regular vine definition 3.3 and return to Step 1. or stop if no tree can be constructed.

Step 1 is done by computing all the possible linear trees (*i.e.*, paths) and assigning to each edge the correlation coefficients between the two variables represented by the vertices. The path with the maximal sum of the correlation coefficients is chosen. For Step 2, each pair of attributes linked in the path are fitted on bivariate copula functions and the best fitted one is chosen using a model selection criterion (such as Akaike Information Criterion [3] or Bayesian Information Criterion [57]).

Note that due to the sequential aspect of the algorithm, it can be stopped at any level [12]. A truncated vine copula will have independent bivariate copulas to model the conditional variables for all the trees below the threshold. We will refer to the truncation level as Ψ .

The complexity of this heuristic lies on the fitting of n bivariate copulas on the first tree T_1 , $n - 1$ copulas on the second tree T_2 and so on until the last bivariate copula on the last tree T_{n-1} , in which n is the number of attributes. If $e(m)$ is the complexity of estimating a bivariate copula given m records, then the complexity of the Dissmann’s algorithm is given by $O(e(m) \times n \times \Psi)$.

The critical drawback of a sequential heuristic such as Dissmann’s algorithm is that reaching the optimal solution is not guaranteed, in particular for high dimensions. Moreover, the greedy algorithm needs to compute the spanning tree and to choose a copula family for every pair of nodes at each level of the vine leading to important computational time with the increase in dimensionality.

Recent advances regarding vine copulas construction such as [45] and [65] propose respectively the use

of the Lasso [69] method and the use of Reinforcement Learning (RL) with Long Short Term Memory networks (LSTM) for near-optimal vine selection. Another approach found in the literature uses auto-encoders on the data (here it could be used on the private pseudo-observations) to mitigate the curse of dimensionality [66]. A recent paper proposed a differentially private graphical model for low-dimensional marginals [42] that could also be beneficial to our work. This work defines an ad hoc framework for accurately and privately estimate marginals and improve existing generative models such as PrivBayes. We leave the investigation of these approaches as future works.

4 COPULA-SHIRLEY

COPULA-SHIRLEY is a simple and flexible framework for private data synthesis through vine copulas. In this section, we will first provide an overview of the algorithm, before describing in detail each of its subcomponents. Finally, we present our privacy analysis as well as the membership inference attack that we have used to evaluate the privacy level provided by our method.

4.1 Overview

Algorithm 1 outlines the framework of our method. The COPULA-SHIRLEY algorithm takes as input a dataset D as well as a parameter ϵ , representing the privacy budget. The last two input parameters of COPULA-SHIRLEY, $nGen$ and $EncodingMethod$ correspond respectively to the number of synthetic data points to generate and which method of encoding for the categorical attributes to use.

Note that copulas are originally mathematical objects used for modelling *continuous* numerical variables. This means that the application of copulas to categorical attributes requires an adequate preprocessing of the data, which is done by the *Preprocessing* function. Using the preprocessed data, the algorithm can tackle the task of building a vine copula for modelling the data in a differentially-private manner. Afterwards, this vine copula can be sampled for producing new synthetic data.

4.2 Preprocessing

As directly cited from [36]: “*Although the data should be continuous to guarantee the continuity of marginals,*

Algorithm 1: COPULA-SHIRLEY

Input: Dataset: D , global privacy budget: ϵ ,
 number of records to generate: $nGen$,
 encoding method : $EncodingMethod$

Output: Differentially-private synthetic
 records: D_{syn}

- 1 $(pseudoObs, dpCDFs) \leftarrow$
 $Preprocess(D, \epsilon, EncodingMethod)$
 (Section 4.2)
- 2 $vineModel \leftarrow SelectVineCopula(pseudoObs)$
 (Section 4.3)
- 3 $D_{syn} \leftarrow$
 $GenerateData(vineModel, nGen, dpCDFs)$
 (Section 4.4)
- 4 **return** D_{syn}

discrete data in a large domain can still be considered as approximately continuous as their cumulative density functions do not have jumps, which ensures the continuity of margins.” This implies that if treated as continuous, discrete data can be modelled with copulas. Thus, copulas can be used to model a wide range of data without the need of much preprocessing. However, an important issue arises when copulas are applied on categorical data as such data do not have an ordinal scale and copulas mostly use rank correlation for modelling. In this situation, one common trick is to view categorical data simply as discrete data in which the order is chosen arbitrarily (*e.g.*, by alphabetical order), which is known as *ordinal encoding*. This trick has been proven to be a downfall for vine copula modelling for some metrics, as it will be discussed in Section 5.

Another technique to deal with categorical attributes is the use of dummy variables, in which categorical values are transformed into binary indicator variables [64] (this technique is also known as *one-hot encoding*). The use of dummy variables helps preserve the rank correlation between the attributes used by the copulas (pseudo-observations).

Two other *supervised* encoding techniques known as the Weight of Evidence (WOE) encoding [72] and the Generalized Linear Mixed Model (GLMM) encoding [13, 71] have been evaluated. Both need a reference attribute (called a predictor) and encode the categorical attributes in a way that maximizes the correlation between the pair of encoded and reference attributes. The WOE encoder can only be used with a binary reference attribute and is computed using the natural log of the number of 1’s over the number of 0’s of the ref-

erence attribute given the (encoded) attribute’s value. The GLMM can be seen as an extension of logistic regression in which the encoding of an attribute is computed by the expected value of an event (*i.e.*, encoded attribute) given the values of the predictor.

The preprocessing method shown in Algorithm 2 converts categorical values into dummy variables (if necessary), computes the differentially-private CDFs from the noisy histograms and outputs the noisy pseudo-observations needed for the vine copula model construction. In the following, we describe in more details these different processes.

Split training. Since our framework needs two sequential training processes: learning differentially-private cumulative functions and learning the vine-copula model, we rely on the parallel composition (Theorem 3.3) of differential privacy by splitting the dataset D into two subsets instead of using the *sequential composition* (Theorem 3.2) and splitting the global budget ϵ . In this manner, we efficiently use the modelling strengths of copulas functions by sacrificing data points to reduce the noise added via the differentially-private mechanism (such as the Laplacian mechanism) while preserving much of the data’s utility. Algorithm 2, line 3 shows the process of splitting the dataset into two subsets.

Algorithm 2: Preprocessing

Input: Dataset: D , global privacy budget: ϵ ,
 encoding method : $EncodingMethod$

Output: Differentially-private
 pseudo-observations: $pseudoObs$,
 differentially-private CDFs: $dpCDFs$

- 1 $D \leftarrow EncodingMethod(D)$
- 2 $(dpTrainSet, modelTrainSet) \leftarrow$
 $SplitDataset(D)$
- 3 $dpHistograms \leftarrow$
 $ComputeDPHistograms(dpTrainSet, \epsilon)$
- 4 **foreach** $hist$ in $dpHistograms$ **do**
- 5 $cdf \leftarrow \frac{CumulSum(hist[binCounts])}{Sum(hist[binCounts])}$
- 6 $dpCDFs[col] \leftarrow cdf$
- 7 **foreach** col in $modelTrainSet$ **do**
- 8 $cdf \leftarrow dpCDFs[col]$
- 9 $pseudoObs[col] \leftarrow cdf(modelTrainSet[col])$
- 10 **return** $(pseudoObs, dpCDFs)$

Computation of DP histograms. The estimation of differentially-private histograms is the key pro-

cess of COPULA-SHIRLEY. The current implementation computes naively a differentially-private histogram by adding Laplacian noise of mean 0 and scale $\frac{\Delta}{\epsilon}$ with $\Delta = 2$ to each bin count of every histogram (in the bounded setting of DP, the global sensitivity Δ of histogram computation is 2). A more sophisticated method, such as the ones defined in [2, 73, 76], could possibly be used to improve on the utility of the synthetic data.

Computations of CDFs. Cumulative and probability density functions are intrinsically linked together and it is almost trivial to go from one to another. With continuous functions, CDFs are estimated via integration of the PDF curves. In a discrete environment like ours, as we use histograms to estimate PDFs, a simple normalized cumulative sum over the bin counts provides a good enough estimation of the CDFs, which is shown on line 6 of Algorithm 2. This is similar to the approach proposed by the Harvard University Privacy Tools Project in the lecture notes about DP-CDFs [44], in which they add noise to the cumulative sum counts and then normalize. Our method always produces a strictly monotonically increasing function, whereas their approach tends to produce non-monotonic jagged CDFs. A strictly monotonic increasing function is desirable as it means that the theoretical CDF always has an inverse. Non-monotonic jagged CDFs can produce erratic behaviours when transforming data to pseudo-observations and especially when mapping back to the natural scale with the inverse CDFs.

Computation of pseudo-observations. As copulas can only model uniform standard marginals (*i.e.*, pseudo-observations, we need to transform our data. Easily enough, the PIT states that for a random variable X and its CDF F_X , we have that $F_X(X)$ is uniform standard. Algorithm 2 does this at lines 8-10 by first extracting the corresponding CDF (line 9) before applying the said CDF onto the data (disjoint from the set used for computing the CDFs) (line 10).

4.3 Construction of the vine copula

The two existing previous works at the intersection of differential privacy and copulas are restricted to Gaussian copulas [6, 36] because it makes it easier to build them in a differentially private manner. Indeed, it is possible to represent a Gaussian copula by the combination of a correlation matrix and the marginal densities. In such case, it is enough to introduce appropriate noise in these two mathematical structures to obtain a copula model that is differentially-private by design. The

use of vine copulas in such context is more complex, as for each pair of copulas, a noisy estimation of its parameters is required, which results in an overall huge injection of noise. Such a framework would also need a differentially-private implementation of the Dissmann’s algorithm, which is a complex task.

In contrast, COPULA-SHIRLEY reduces considerably the noise added and the complexity of the implementation by computing the dependencies on the noisy marginal densities rather than on the original data, thus enabling the use of vine copulas in a private manner. COPULA-SHIRLEY is generic in the sense that it can be used with any tree or copula selection criterion, any algorithm for vine construction and any method for differentially-private histograms or PDFs. The complexity of COPULA-SHIRLEY is mainly impacted by the algorithm for selecting the vine copula as the complexity of `ComputeDPHistograms`, `ComputeCDFs` and `ComputePseudoObs` is $O(nd)$, in which n is the number of profiles and d is the number of attributes (*i.e.*, dimensions) describing each profile. This latest cost is negligible compared to the cost of Dissmann’s algorithm.

Algorithm 3: SelectVineCopula

Input: Pseudo-observations: *pseudoObs*

Output: Vine Copula Model: *vineModel*

```

1 vineModel ←
  DissmannVineSelection(pseudoObs)
  (Section 3.2)
2 return (vineModel)

```

Algorithm 3 outlines the general process for the vine selection. As stated earlier, our method only needs the differentially-private uniform standard pseudo-observations in order to select a vine. Dissmann’s algorithm is then used with the noisy observations, and only these observations, to fit the best vine copula. The current implementation of COPULA-SHIRLEY uses the implementation of Dissmann’s algorithm from the R library `rvinecopulib` [47]. It offers a complete and highly configurable method for vine selection as well as some optimization techniques for reducing the computational time for building the vine.

4.4 Generation of synthetic private data

The last step of our framework is the generation of synthetic data. To realize this, uniform standard

observations must be sampled from the vine copula model selected via the previous method. As we use the `rvinecopulib` implementation of the vine selection algorithm, we naturally use their implementation of the observation sampling from vines. Line 1 of Algorithm 4 refers to the implementation of this sampling. We refer the interested reader to [20] for more details about how to sample from nested conditional probabilities as defined by the vine copula structure.

To map the sampled observations back to their original values, we use the Inverse Probability Integral Transform (Inverse PIT), which only requires the inverse CDFs of the attributes. This process is shown on lines 2 to 5 of Algorithm 4. This last step concludes the framework of differentially-private data generation with COPULA-SHIRLEY.

Algorithm 4: GenerateData

Input: Vine Copula Model: $vineModel$,
 Number of records to generate: $nGen$,
 Differentially-private CDFs: $dpCDFs$

Output: Synthetic records: D_{syn}

```

1  $synthObs \leftarrow$ 
    $SampleFromVine(vineModel, nGen)$ 
2 foreach  $col$  in  $synthObs$  do
3    $cdf \leftarrow dpCDFs[col]$ 
4    $invcdf \leftarrow InverseFunction(cdf)$ 
5    $D_{syn}[col] \leftarrow invcdf(synthObs[col])$ 
6 return  $D_{syn}$ 
    
```

4.5 Differential privacy analysis

It should be clear by now that COPULA-SHIRLEY relies solely on differentially-private histograms, which makes the whole framework differentially-private by design.

Theorem 4.1 (DP character of COPULA-SHIRLEY).
Algorithm 1 is ϵ -differentially-private.

Proof. The method `ComputeDPHistograms` provides ϵ -differentially-private histograms by the Laplacian mechanism theorem [22]. The computation of $dpCDFs$ only uses the previous differentially-private histograms to compute the CDFs, in a parallel fashion; thus these density functions are ϵ -differentially-private as per the *Closure Under Post-Processing* and *Parallel Composition* properties of differential privacy. To obtain the

$pseudoObs$, we simply apply the differentially-private CDFs to the held-to model training set. The resulting $pseudoObs$ data is ϵ -differentially-private due to the application of a ϵ -differentially-private mechanism and by the *Parallel Composition* theorem.

`DissmannVineSelection` only uses the ϵ -differentially-private set of data $pseudoObs$ for its selection, resulting in a ϵ -differentially-private vine structure by the *Closure Under Post-Processing* property. Both, `SampleFromVine` and `InverseFunction` only use private data and therefore do not violate the *Closure Under Post-Processing* property. Finally, the whole process is closed and never violates the *Closure Under Post-Processing* property of differential privacy, as the algorithm only operates independently for each attribute and therefore makes use of the *Parallel Composition* property; thus Algorithm 1 is ϵ -differentially-private. \square

4.6 Privacy evaluation through membership inference

As stated earlier, we believe that one should not rely solely on the respect of a formal model such as differential privacy but also that synthetic data should be assessed with respect to a privacy test based on inference attacks to quantify the privacy protection provided by the synthesis method.

In this work, we opted for the Monte Carlo membership inference attack (MCMIA) introduced by Hilprecht, Härterich and Bernau [30] to assess the privacy of our method. Simply put, MCMIA quantifies the risk of pinpointing training records from other records drawn from the same distribution given synthetic profiles. One of the benefits of MCMIA is that it is a non-parametric and model-agnostic attack. In addition, MCMIA provides high accuracy in situations of model overfitting in generative models and outperforms previous attacks based on shadow models.

The framework of the attack is as follows. Let \mathcal{S}_T be a subset of m records from the training set of the model and \mathcal{S}_C be a subset of m control records. Control records are defined as records from the same distribution as the training ones but never used in the training process. Let x be a record from the global data domain \mathcal{D} ; $U_r(x) = \{x' \in \mathcal{D} \mid d(x, x') \leq r\}$ is defined as the neighbourhood of x with respect to the distance d (*i.e.* the set of records x'_i close to x). A synthetic record g from a generative model G is more likely to be similar to a record x as the probability $P[g \in U_r(x)]$ increases. The probability $P[g \in U_r(x)]$ is estimated via the Monte

Carlo integration: $P[g \in U_r(x)] \approx \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{g_i \in U_r(x)}$, in which g_i are synthetic samples from G .

To further refine the attack, the authors propose an alternative estimation of $P[g \in U_r(x)]$ based on the distance between x and its neighbours :

$$P[g \in U_\epsilon(x)] \approx \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{g_i \in U_r(x)} \log(d(x, g_i) + \eta)$$

in which η is an arbitrary small value set to avoid $\log 0$ (we used $\eta = 10^{-12}$). The function

$$\hat{f}_{MC}(x) := \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{g_i \in U_r(x)} \log(d(x, g_i) + \eta)$$

will be the record-scoring function. From this definition, if the record x obtains a high score, the synthetic samples g_i are very likely to be close to x and implies an overfitted model over x .

To compute the privacy score of a synthetic dataset, given \mathcal{S}_G a set of n synthetic samples from a generative model G , we first compute $\hat{f}_{MC}(x)$ over \mathcal{S}_G for all $x \in \mathcal{S}_T \cup \mathcal{S}_C$. Afterwards, we take the m records from the union set $\mathcal{S}_T \cup \mathcal{S}_C$ with the highest \hat{f}_{MC} scores to form the set \mathcal{I} . By computing the ratio of records that are both in \mathcal{S}_T and \mathcal{I} , we obtain the Privacy Score of the model over the set \mathcal{S}_T :

$$PrivacyScore(\mathcal{S}_T) := \frac{\text{card}(\mathcal{S}_T \cap \mathcal{I})}{\text{card}(\mathcal{S}_T)}$$

$PrivacyScore(\mathcal{S}_T)$ can be interpreted as the ratio of training records successfully distinguished from control records. From this definition, a score of 1 means that the full training set has been successfully recovered, thus implying a privacy breach while a score of 0.5 means that the records from the training and control sets are indistinguishable among the synthetic profiles.

In our experiments, we found out that using a coarser distance like the Hamming distance provide higher membership inference scores than the Euclidean distance. To set the value of r , the size of the neighbourhood, we used the *median heuristic* as defined by the authors as it was the one providing the highest accuracy:

$$r = \text{median}_{1 \leq i \leq 2m} \left(\min_{1 \leq j \leq n} d(x_i, g_j) \right)$$

in which $x_i \in \mathcal{S}_T \cup \mathcal{S}_C$ and $g_j \in \mathcal{S}_G$.

5 Experiments

In this section, we investigate experimentally the privacy-utility trade-off of the synthetic data gener-

ated by COPULA-SHIRLEY. In addition, we compare it with two ϵ -differentially private data synthesis methods presented in Section 2, namely PrivBayes [77] and DP-Copula [36]. In McKay’s and Snoke’s article about the NIST challenge on differentially-private data synthesis [41], the authors ranked PrivBayes in the top 5, supporting its usage in our comparative work. As COPULA-SHIRLEY can be seen as a refinement of the DP-Copula model with the introduction of vines, we wanted to compare our method to a differentially-private Gaussian copula approach. Our experimental setup is available as a Python script at: <https://github.com/alxxrg/copula-shirley>.

5.1 Experimental setting

Datasets. Three datasets of various dimensions have been used in our experiments. The first one is the UCI Adult dataset [21], which contains 32 561 profiles. Each profile is described by 14 attributes such as gender, age, marital status and native country. The attributes are mostly categorical (8), the rest is discrete (6).

The second dataset used is COMPAS [5], which consists of records from criminal offenders in Florida during 2013 and 2014. It is the smallest set of the three and contains 10 568 profiles, each described with 13 attributes quite similar to Adult that are either discrete or categorical with the same proportion.

The third dataset used is Texas Hospital [68] from which we uniformly sampled 150 000 records from a set of 636 140 records and selected 17 attributes, 11 of which are categorical. We sample down this dataset to reduce the burden of the computational task, mainly for the PrivBayes algorithm.

Parameters for data synthesis. To evaluate the data synthesis, we have run a k -fold cross-validation technique [25] with $k = 5$. For each fold, all generative models are learned on the training set of this fold and then synthesized the same number of records. All evaluation metrics are measured by using the fold’s left-out test set and the newly generated synthetic data. For the privacy budget, we have tried various values for this parameter in the range $\epsilon \in [0.0, 8.0, 1.0, 0.1, 0.01]$ (here $\epsilon = 0$ means that the DP is deactivated), similar to the parameters used in the comparative study of the NIST competition for evaluating differentially-private synthesis method [41].

For the categorical encoders, we have used the Python library `category_encoders` [40]. For the supervised encoders (WOE & GLMM), to avoid leaking in-

formation about the training set, we train the encoders on a disjoint set that is the fold’s left-out test set. By default, we used the WOE encoder as it was shown to provide slightly better results (see Figure 4). For the membership inference attack implementation discussed in Section 4.6, the control set used is the fold’s left-out set.

Implementation details. As previously mentioned, COPULA-SHIRLEY uses the implementation of Dissmann’s algorithm from the R library `rvinecopulib` [47]. In our tests, we used all the default parameters of the vine selection method, which can be found in the reference section [47], except for two parameters: the truncation level Ψ and the measure of correlation for the optimal tree selection. We opted to stop at the second level of the tree ($\Psi = 2$). The truncation level has been thoroughly studied and we show that a deeper vine model does not drastically improve the model as shown in Appendix C. We also opted for the Spearman’s ρ rank correlation as it is the preferred statistic when ties occur in the data [53].

As stated earlier, we have split the training set into two disjoint subsets with a ratio of 50/50, respectively for the differentially-private histograms and the pseudo-observations. The impact of different ratios is illustrated in Figure 3. Finally, for a more refined representation, we choose to use as many bins for our histograms as there are unique values in the input data. By default, we use the Laplacian mechanism for computing the DP-histograms (see, however, Appendix B for the impact on the utility of other DP mechanisms).

PrivBayes. We use the implementation of PrivBayes referenced by [41] called DataSynthesizer [52]. Apart from the privacy budget ϵ , the implementation of PrivBayes we have run has only one parameter, which is the maximal number of parents nodes in the Bayesian network. We use the default parameter which is 3.

DP-Copula. While we did not find an official implementation of Li, Xiong and Jiang [36], we discovered and used an open-source implementation available on GitHub [54]. In addition to the privacy budget, the only parameter of DP-Copula is used to tune how this budget is divided between the computation of the marginal densities and the correlation matrix in a differentially-private manner. We choose to set the value of this parameter so that half of the privacy budget is dedicated to the computation of the marginal densities and half to the computation of the correlation matrix.

5.2 Utility testing

Statistical tests. Statistical tests can be used to quantify the similarity between the original dataset and the synthetic one. We use the Kolmogorov-Smirnov (KS) distance to estimate the fidelity of the distributions of the synthetic data compared to the original data. The KS distance is computed per attributes, the reported scores in the results section representing the average score over all attributes. We also evaluate the variation in the correlation matrices between the raw and synthetic data using the Spearman’s ρ rank correlation. The score represent the mean absolute difference of the correlation coefficients between the two matrices. In the following section, we refer to this score as the correlation delta metric. See Appendix A for more details on these tests.

Classification tasks. Classification tasks are complementary to statistical tests in the sense that they simulate a specific use case, which is the prediction of a specific attribute, thus helping to evaluate if the correlations between attributes are preserved with respect to the classification task considered. In this setting, the utility is measured by training two classifiers. The first classifier is learned on a training set (*i.e.*, the same training set used by the generative models), drawn from the original data while the second one is trained on the synthetic data produced by a generative model. Afterwards, the classifiers are tested on the same test set, which is drawn from the original data but disjoint from the training set. Ideally, a classifier trained on the synthetic data would have a classification performance similar to the one trained on the original data. The comparison between classifiers is done through the use of the Matthews Correlation Coefficient (MCC) [39], which is a measure of the quality of a classification (see Appendix A). Note that the MCC of a random classifier is close to ≈ 0.0 while the score of perfect classifier would be 1.

In our experiments, we evaluated the synthetic data on two classification tasks: on a binary classification problem and a multi-class classification problem. We opted for *gradient boosting* [23] classifier as this algorithm is known for its robustness to overfitting and its performance that is often close to the state-of-the-art methods, such as deep neural networks, on many classification tasks. We use the XGBoost [16] implementation of the gradient boosting algorithm.

To further deepen our analysis, we also evaluated the synthetic data over a simple linear regression task. To evaluate its success, we computed the *root mean*

square error (RMSE):

$$\text{RMSE}(Y, \tilde{Y}) = \sqrt{\frac{\sum_{i=0}^n (\tilde{y}_i - y_i)^2}{n}}$$

in which Y are the true values and \tilde{Y} are the linear model’s outputs.

For the Adult dataset, the binary classification problem is over the attribute salary, the multi-class classification problem over the attribute relationship and the linear regression over age. On COMPAS, the binary classification task is over the attribute `is_violent_recid`, the multi-class problem on race and the linear regression on the attribute `decile_score`. For the classification and regression tasks on Texas, the attributes `ETHNICITY`, `TYPE_OF_ADMISSION` and `TOTAL_CHARGES` are used respectively for the binary, multi-class and regression problems.

5.3 Results

Splitting ratios for the vine copula model. Recall that in our preprocessing step (Algorithm 2, the training dataset is split in two sets, one for learning the DP-CDFs and the other for computing pseudo-observations. We first evaluate the splitting ratios between the two sets. As shown in Figure 3, most metrics are stable across the different ratios. One exception is the KS distance, which exhibits an increase when the ratio given for the pseudo-observations is higher, giving the importance of the DP-CDFs. As there is no clear consensus in the data, we opted for a ratio of 50/50 for the other experiments.

Encoding of the categorical attributes. The proper encoding of the categorical attributes is crucial for our approach. As shown in Figure 4, both classification tasks display lower performances with the ordinal encoding. In addition, the one-hot encoding could not be tested on Texas as the number of attributes augment from 17 to 5375, due to the increase in the computational burden as our approach scales linearly with the number of attributes. The one-hot encoding also perform badly for the KS distance and the regression task in addition of considerably increasing the execution time. We opted for the WOE encoder as it performed best for both classification tasks compared to the GLMM encoder.

Comparative study of the models. Figure 5 illustrates the scores for each privacy budget over the three datasets and four models separately while Figure 6 displays the score over all the values of the privacy budget combined. One of the trends that we observed is

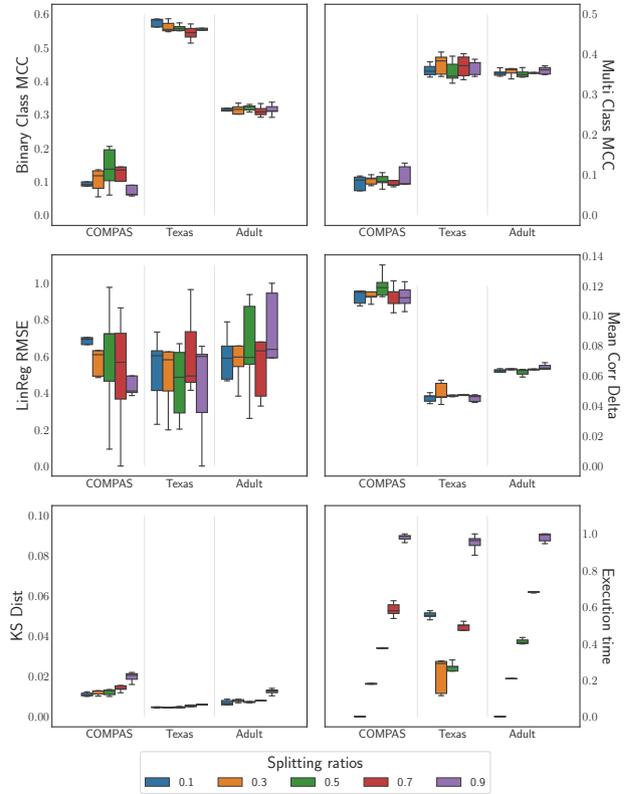


Fig. 3. Impact of the splitting ratios for the vine copula model with $\epsilon = 1.0$. The linear regression RMSE and the execution time are normalized. For the MCC, higher is better, while the other metrics, lower is better. Measures are averaged over a 5-fold cross-validation run.

that PrivBayes provides better score than COPULA-SHIRLEY for most of the other classification tasks. In addition, DP-Copula and DP-Histograms failed completely at the two classification tasks. Our approach and PrivBayes performed well on the regression task compared to DP-Copula. PrivBayes generated the data that with the smallest correlation coefficients difference from the original datasets except for the Texas dataset in which COPULA-SHIRLEY provides the best results overall. In addition, COPULA-SHIRLEY is always the preferred one for generating faithful distributions. The privacy evaluation demonstrates that a smaller privacy budget does not necessarily mean a lower risk of membership inference. Furthermore, all models do not provide a consistent protection over all the datasets. While PrivBayes offers the best protection on the smallest dataset (*i.e.*, COMPAS), COPULA-SHIRLEY is the best on the biggest dataset (*i.e.*, Texas).

Figure 6 exhibits the global scores for each generative model’s synthetic data. PrivBayes produced the best results overall for the classification tasks and some

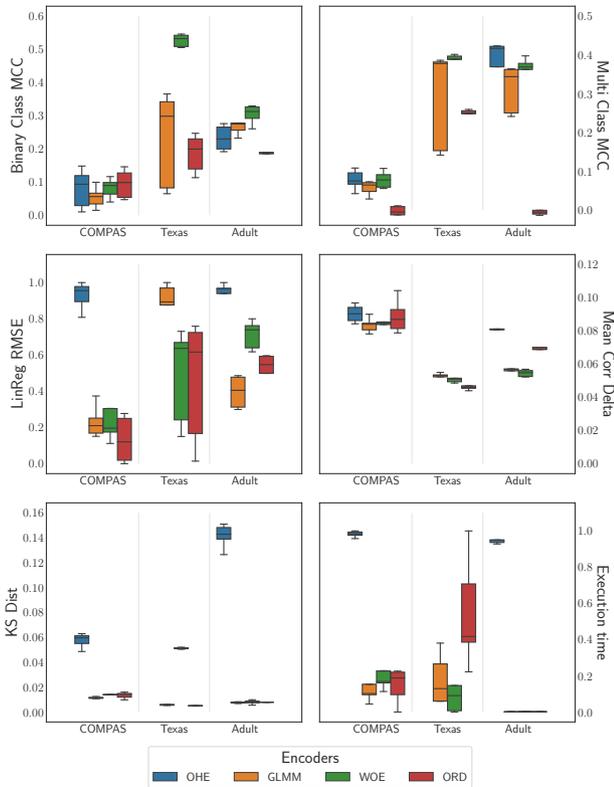


Fig. 4. Impact of the categorical attributes encoding with $\epsilon = 1.0$. The linear regression RMSE and the execution time are normalized. For the MCC, higher is better, while for the other metrics, lower is better. Measures are averaged over a 5-fold cross-validation run.

decent results for the linear regression task and the correlation delta metric. From these results, it seems the best over the four models for capturing the attributes interdependence. From the heights of the bars, it is possible to say that PrivBayes is second to being the most inconsistent generative model. DP-Histogram produced some of the worst data for classification and some of the worst scores for the correlation metric which is not surprising giving that no dependence structure learned. COPULA-SHIRLEY is the runner-up for most of the tests as it produced some of the most faithful distributions along DP-Histogram. Vine-copula models also show a more stable fit to the training data than PrivBayes and DP-Copula. The DP-Copula model produced the most unreliable distributions as well as failing completely at the multi-label classification task and the linear regression task. For the membership inference attack test, all models provided a decent protection but our method provided the best overall protection with the largest dataset of the three, PrivBayes offering the worst. From Figures 5 and 6, COPULA-SHIRLEY seems to be the best

at modelling and protecting Texas, the biggest dataset.

Running time. All experiments were conducted on an Intel core i5-6600k with 16 GB of flash memory. The running times given in Table 1 strengthen the observation that Bayesian approaches can be extremely time-consuming when the dimensionality of data increases. The complexity of using vine copulas (COPULA-SHIRLEY) is also significantly higher than using simple multivariate copulas (DP-Copula).

| Dataset | COPULA-SHIRLEY | PrivBayes | DP-Copula |
|----------------------|----------------|-----------|-----------|
| Adult (32 561 × 14) | 33.452 | 67.211 | 2.241 |
| COMPAS (10 568 × 13) | 1.592 | 3.184 | 0.695 |
| Texas (150 000 × 17) | 34.055 | 123.157 | 17.945 |

Table 1. Average running times in minutes of each method on the three datasets.

Additional analysis on multivariate correlations. We created a small synthetic dataset composed of 5000 records and 6 attributes (named from A to F) with various correlation coefficients between the attributes to show the models strength in capturing the dependence structure in the data. The dataset was curated so that the attributes A and F are highly positively correlated when the values of A are below zero but only slightly negatively correlated when the values of A are above zero. This is shown in Table 2 at columns ‘A<0 - F’ and ‘A>0 - F’. Our method offers the closest correlation coefficients to the original ones four times out of seven. PrivBayes is better at capturing the dependence when it varies in different parts of the data. These also emphasize the fact that the vine copula approach is superior at modelling the dependence structure than DP-Copula and the naive approach of simply sampling from DP-Histograms. See Appendix D for the scatter plot of the curated synthetic data as well as the scatter plots of the observations sampled from the models.

| | A - B | A - C | A - D | A - E | A - F | A<0 - F | A>0 - F |
|------------|---------------|----------------|---------|---------|---------|---------|---------|
| Synth Data | 0.9150 | -0.9565 | 0.2319 | -0.1701 | 0.4018 | 0.7881 | -0.1137 |
| COP-SHIRL | 0.9161 | -0.9555 | 0.2763 | -0.0477 | 0.4059 | 0.5484 | 0.0216 |
| PrivBayes | 0.8794 | -0.9076 | 0.0899 | -0.1860 | 0.3529 | 0.655 | -0.1014 |
| DP-Cop | 0.7933 | -0.1284 | 0.1507 | -0.0552 | 0.3417 | 0.1829 | 0.1472 |
| DP-Hist | 0.0220 | 0.0127 | -0.0268 | -0.0367 | -0.0191 | -0.0186 | -0.0002 |

Table 2. The Spearman’s correlation coefficients between the pair of attributes. Best scores are bold.

Summary of results. Our method COPULA-SHIRLEY displays the highest statistical similarity be-

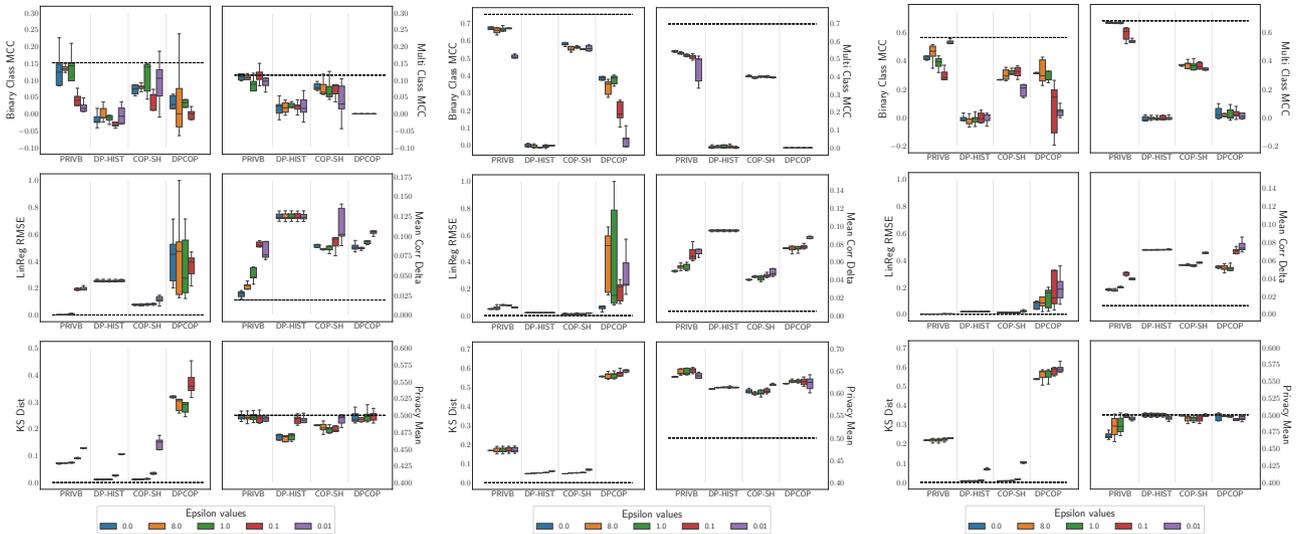


Fig. 5. From left to right: results on COMPAS, Texas and Adult datasets with various ϵ privacy budget. The linear regression RMSE is normalized. Dashed lines represent the scores over raw data and measures are aggregated over a 5-fold cross-validation run.

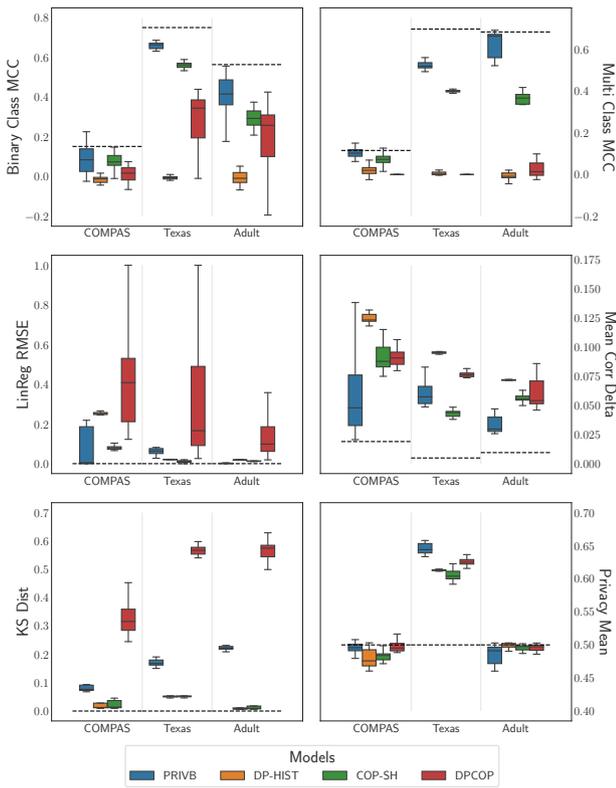


Fig. 6. Global average results of the generative models over all the ϵ dp-budget values combined. The linear regression RMSE is normalized. Dashed lines represent the scores over raw data and measures are aggregated over a 5-fold cross-validation run.

tween raw and synthetic data. While our vine-copula approach outperforms DP-Copula on the classification tasks most of the time, it did not do the same

for PrivBayes. COPULA-SHIRLEY was on par with PrivBayes a few times, mostly for the linear regression task. PrivBayes generated decent synthetic distributions with outstanding inter-attributes dependence, which is why PrivBayes always achieve the best score for the classification tasks. The significant drawback of PrivBayes is the execution time for training the model. Globally, COPULA-SHIRLEY offered decent data quality, a more stable fitting and a stronger protection than PrivBayes on the largest dataset. Compared to its copula-based counterpart DP-Copula, COPULA-SHIRLEY globally produced far better data, in addition of being a clear improvement over multivariate copula and the naive DP-Histograms sampling. In addition, we believe the performances of COPULA-SHIRLEY could be boosted with a few optimizations like a more sophisticated vine-copula selection algorithm or a thorough preprocess for better rank correlation preservation.

6 Conclusion

Data synthesis methods that have been trained to also provide strong privacy guarantees are considered to be a promising approach in data anonymization. In this work, we have proposed COPULA-SHIRLEY, a new data synthesis method based on differentially-private vine copulas. Our method benefits from the simplicity and flexibility of the modelling power of vine copulas, which reproduces distributions faithfully while ensuring privacy through the use of differential privacy as well as

a privacy test based on membership inference. However a fundamental open question is to characterize what it means for synthetic data to be private yet realistic. Future work will include the development of a more complete framework to evaluate the privacy risks of releasing synthetic data through the use of diverse inference attacks.

Acknowledgements

Sébastien Gambs is supported by the Canada Research Chair program, a Discovery Grant (NSERC), the Legalia project (FQRNT) as well as a grant for the Office of the Privacy Commissioner (OPC) of Canada.

References

- [1] Kjersti Aas, Claudia Czado, Arnoldo Frigessi, and Henrik Bakken. Pair-copula constructions of multiple dependence. *Insurance: Mathematics and economics*, 44(2):182–198, 2009.
- [2] Gergely Acs, Claude Castelluccia, and Rui Chen. Differentially private histogram publishing through lossy compression. In *2012 IEEE 12th International Conference on Data Mining*, pages 1–10, Brussels, Belgium, 2012. Institute of Electrical and Electronics Engineers (IEEE).
- [3] Hirotugu Akaike. Information theory and an extension of the maximum likelihood principle. In Hirotugu Akaike, editor, *Selected papers of hirotugu akaike*, pages (p. 199–213). New York: Springer, 1998.
- [4] James A. Anderson. *An Introduction to Neural Networks*. Cambridge: The MIT Press, 1997.
- [5] Julia Angwin, Jeff Larson, Lauren Kirchner, and Surya Mattu. Machine bias, Mar 2019.
- [6] Hassan Jameel Asghar, Ming Ding, Thierry Rakotoarivelo, Sirine Mrabet, and Mohamed Ali Kaafar. *Differentially Private Release of High-Dimensional Datasets using the Gaussian Copula*. *arXiv preprint*, 2019. (Preprint).
- [7] Borja Balle and Yu-Xiang Wang. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*, pages 394–403, 2018.
- [8] A Bárdossy and GGS Pegram. Copula based multisite model for daily precipitation simulation. *Hydrology & Earth System Sciences*, 13(12):2299—2314, 2009.
- [9] Tim Bedford and Roger M. Cooke. Probability density decomposition for conditionally dependent random variables modeled by vines. *The Annals of Mathematics and Artificial Intelligence*, 32(1-4):245–268, 2001.
- [10] Tim Bedford and Roger M. Cooke. Vines: A new graphical model for dependent random variables. *The Annals of Statistics*, 30(4):1031–1068, 2002.
- [11] Vincent Bindschaedler, Reza Shokri, and Carl A Gunter. *Plausible deniability for privacy-preserving data synthesis*. *arXiv preprint*, 2017. (Preprint).
- [12] Eike C Brechmann, Claudia Czado, and Kjersti Aas. Truncated regular vines in high dimensions with application to financial data. *Canadian Journal of Statistics*, 40(1):68–85, 2012.
- [13] Norman E Breslow and David G Clayton. Approximate inference in generalized linear mixed models. *Journal of the American statistical Association*, 88(421):9–25, 1993.
- [14] Lane F Burgette and Jerome P Reiter. Multiple imputation for missing data via sequential regression trees. *American journal of epidemiology*, 172(9):1070–1076, 2010.
- [15] Thee Chanyaswad, Changchang Liu, and Prateek Mittal. Ron-gauss: Enhancing utility in non-interactive private data release. *Proceedings on Privacy Enhancing Technologies*, 2019(1):26–46, 2019.
- [16] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, and Yuan Tang. Xgboost: extreme gradient boosting. *R package version 0.4-2*, pages 1–4, 2015.
- [17] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3:1376, 2013. doi: 10.1038/srep01376.
- [18] Yves-Alexandre De Montjoye, Laura Radaelli, Vivek Kumar Singh, et al. Unique in the shopping mall: On the reidentifiability of credit card metadata. *Science*, 347(6221):536–539, 2015.
- [19] Luc Devroye. *Non-Uniform Random Variate Generation*. New York: Springer-Verlag, 1986.
- [20] Jeffrey Dissmann, Eike C Brechmann, Claudia Czado, and Dorota Kurowicka. Selecting and estimating regular vine copulae and application to financial returns. *Computational Statistics & Data Analysis*, 59:52–69, 2013.
- [21] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
- [22] 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.
- [23] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.
- [24] Lorenzo Frigerio, Anderson Santana de Oliveira, Laurent Gomez, and Patrick Duverger. Differentially private generative adversarial networks for time series, continuous, and discrete open data. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 151–164, Lisbon, Portugal, 2019. Springer.
- [25] Seymour Geisser. The predictive sample reuse method with applications. *Journal of the American statistical Association*, 70(350):320–328, 1975.
- [26] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. *SIAM Journal on Computing*, 41(6):1673–1693, 2012.
- [27] Katsuichiro Goda and Solomon Tesfamariam. Multi-variate seismic demand modelling using copulas: Application to non-ductile reinforced concrete frame in victoria, canada. *Structural Safety*, 56:39–51, 2015.
- [28] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and

- Yoshua Bengio. Generative adversarial nets. In *28th Conference on Neural Information Processing Systems*, pages 2672–2680, Montréal, Canada, 2014. Advances in Neural Information Processing Systems.
- [29] Jing He, Hongzhe Li, Andrew C Edmondson, Daniel J Rader, and Mingyao Li. A gaussian copula approach for the analysis of secondary phenotypes in case–control genetic association studies. *Biostatistics*, 13(3):497–508, 2012.
- [30] Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. Monte carlo and reconstruction membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019(4):232–249, 2019.
- [31] A Hoyer and O Kuss. Meta-analysis of diagnostic tests accounting for disease prevalence: a new model using trivariate copulas. *Statistics in medicine*, 34(11):1912–1924, 2015.
- [32] Finn V. Jensen. *Introduction to Bayesian Networks*. New York: Springer, 1997.
- [33] Harry Joe. *Multivariate models and multivariate dependence concepts*. Londres: Chapman & Hall/CRC, 1997.
- [34] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations (ICLR)*, New Orleans, USA, 2019. ICLR.
- [35] David Lazer, D Brewer, N Christakis, J Fowler, and G King. Life in the network: the coming age of computational social. *Science*, 323(5915):721–723, 2009.
- [36] Haoran Li, Li Xiong, and Xiaoqian Jiang. Differentially private synthesization of multi-dimensional data using copula functions. In *Proceedings of the 17th International Conference on Extending Database Technology*, volume 2014, pages 475–486, Athens, Greece, 2014. Extending Database Technology (EDBT).
- [37] Ziqi Liu, Yu-Xiang Wang, and Alexander Smola. Fast differentially private matrix factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 171–178, Vienna, Austria, 2015.
- [38] Donald MacKenzie and Taylor Spears. ‘the formula that killed wall street’: The gaussian copula and modelling practices in investment banking. *Social Studies of Science*, 44(3):393–417, 2014.
- [39] Brian W Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.
- [40] William D McGinnis, Chapman Siu, S Andre, and Hanyu Huang. Category encoders: a scikit-learn-contrib package of transformers for encoding categorical data. *Journal of Open Source Software*, 3(21):501, 2018.
- [41] Claire McKay Bowen and Joshua Snoko. Comparative study of differentially private synthetic data algorithms and evaluation standards. *arXiv preprint*, 2019. (Preprint).
- [42] Ryan McKenna, Daniel Sheldon, and Gerome Miklau. Graphical-model based estimation and inference for differential privacy. In *International Conference on Machine Learning*, pages 4435–4444. PMLR, 2019.
- [43] Casey Meehan, Kamalika Chaudhuri, and Sanjoy Dasgupta. *A Non-Parametric Test to Detect Data-Copying in Generative Models*. *arXiv preprint*, 2020. (Preprint).
- [44] Daniel Muise and Kobbi Nissim. Notes on differential privacy in cdfs, https://privacytools.seas.harvard.edu/files/privacytools/files/dpcdf_usermanual_2016.pdf. *Harvard University Privacy Tools Project*, April 2016.
- [45] Dominik Müller and Claudia Czado. Selection of sparse vine copulas in high dimensions with the lasso. *Statistics and Computing*, 29(2):269–287, 2019.
- [46] Takao Murakami, Koki Hamada, Yusuke Kawamoto, and Takuma Hatano. Privacy-preserving multiple tensor factorization for synthesizing large-scale location traces. *arXiv preprint arXiv:1911.04226*, 2019.
- [47] Thomas Nagler and Thibault Vatter. R interface to the vinecopulib c++ library, <https://vinecopulib.github.io/rvinecopulib/>, 2017.
- [48] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy*, pages 111–125, Oakland, USA, 2008. Institute of Electrical and Electronics Engineers (IEEE).
- [49] Roger B. Nelsen. *An introduction to copulas (2nd ed.)*. New York: Springer Science & Business Media, 2007.
- [50] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. *Semi-supervised knowledge transfer for deep learning from private training data*. *arXiv preprint*, 2016. (Preprint).
- [51] Andrew J Patton. Copula-based models for financial time series. In *Handbook of financial time series*, pages (p. 767–785). Springer, 2009.
- [52] 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, Chicago, USA, 2017. Association for Computing Machinery (ACM).
- [53] Marie-Therese Puth, Markus Neuhäuser, and Graeme D Ruxton. Effective use of spearman’s and kendall’s correlation coefficients for association between two measured traits. *Animal Behaviour*, 102:77–84, 2015.
- [54] Thierry Rakotoarivelo. Dpcopula-kendall algorithm, 2019.
- [55] Jerome P Reiter. Using cart to generate partially synthetic public use microdata. *Journal of Official Statistics*, 21(3):441, 2005.
- [56] Luc Rocher, Julien M Hendrickx, and Yves-Alexandre De Montjoye. Estimating the success of re-identifications in incomplete datasets using generative models. *Nature communications*, 10(1):1–9, 2019.
- [57] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.
- [58] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy*, pages 3–18, San Jose, USA, 2017. Institute of Electrical and Electronics Engineers (IEEE).
- [59] Ryan Singel. Netflix cancels recommendation contest after privacy lawsuit. *Wired*, 12 février 2010.
- [60] Abe Sklar. Fonctions de répartition à n dimensions et leurs marges. *Publications de l’Institut de Statistique de l’Université de Paris*, 8:229–231, 1959.
- [61] Nikolai Vasil’evich Smirnov. Approximate laws of distribution of random variables from empirical data. *Uspekhi Matematicheskikh Nauk*, (10):179–206, 1944.
- [62] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. Synthetic data—a privacy mirage. *arXiv preprint*

arXiv:2011.07018, 2020.

- [63] William J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton: Princeton University Press, 1994.
- [64] Daniel B Suits. Use of dummy variables in regression equations. *Journal of the American Statistical Association*, 52(280):548–551, 1957.
- [65] Yi Sun, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Learning vine copula models for synthetic data generation. In *Proceedings of the 2019 AAAI Conference on Artificial Intelligence*, volume 33, pages 5049–5057, Honolulu, USA, 2019. Advancement of Artificial Intelligence (AAAI).
- [66] Natasa Tagasovska, Damien Ackerer, and Thibault Vatter. *Copulas as High-Dimensional Generative Models: Vine Copula Autoencoders*. *arXiv preprint*, 2019. (Preprint).
- [67] Uthaipon Tantipongpipat, Chris Waites, Digvijay Boob, Amaresh Ankit Siva, and Rachel Cummings. Differentially private mixed-type data generation for unsupervised learning. *arXiv preprint arXiv:1912.03250*, 2019.
- [68] Texas Department of State Health Services, Austin, Texas. *Texas Hospital Inpatient Discharge Public Use Data File 2013 Q1*, <https://www.dshs.texas.gov/THCIC/Hospitals/Download.shtm>. 2013.
- [69] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.
- [70] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. Dp-cgan: Differentially private synthetic data and label generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 98–104, Long Beach, USA, 2019.
- [71] David A Williams. Extra-binomial variation in logistic linear models. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 31(2):144–148, 1982.
- [72] IJ Wod. Weight of evidence: A brief survey. *Bayesian statistics*, 2:249–270, 1985.
- [73] Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. Differential privacy via wavelet transforms. *IEEE Transactions on knowledge and data engineering*, 23(8):1200–1214, 2010.
- [74] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. *Differentially private generative adversarial network*. *arXiv preprint*, 2018. (Preprint).
- [75] Chugui Xu, Ju Ren, Deyu Zhang, Yaoyue Zhang, Zhan Qin, and Kui Ren. Ganobfuscator: Mitigating information leakage under gan via differential privacy. *IEEE Transactions on Information Forensics and Security*, 14(9):2358–2371, 2019.
- [76] Jia Xu, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, Ge Yu, and Marianne Winslett. Differentially private histogram publication. *The VLDB Journal*, 22(6):797–822, 2013.
- [77] Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4):1–41, 2017.

7 Appendix

A Details on statistical measures

Kolmogorov-Smirnov distance. The Kolmogorov-Smirnov (KS) test is a classical statistical hypothesis test that can be used to test if two samples are drawn from the same distribution [61]. In this test, the hypothesis that the reference distribution and the experimental one follow the same law is considered to be valid only if the statistic of the test D_{KS} is below a threshold $\sigma(\alpha)$. To realize this test, a reference (or theoretical) distribution f_t is needed as well as an experimental distribution f_e and their respective cumulative density functions F_t and F_e . The statistic of the KS test is given by :

$$D_{KS}(F_t, F_e) = \sup_x |F_t(x) - F_e(x)|. \quad (1)$$

The KS statistic, also known as the Kolmogorov-Smirnov distance, is a distance between the empirical distribution function of the experimental sample and the cumulative distribution function of the reference distribution. The KS distance is between 0 and 1. The KS test can be used to assess the capacity of the generative model to recreate faithful distributions to the original data.

Mean Correlation Delta. The score represent the mean absolute difference of the correlation coefficients between the correlation matrices of the reference dataset and the synthetic dataset. If C_o represent the Spearman correlation matrix of the reference dataset and C_s represent the correlation matrix of the synthetic dataset, the Mean Correlation Delta is computed as follows:

$$\frac{\sum_{i,j=1}^n |C_o(i,j) - C_s(i,j)|}{\sum_{i,j} 1}$$

where $C(i, j)$ is the correlation coefficient between the i -th and j -th attributes. This metric is inspired by the one used in the PWSCUP 2020 “Anonymity against Membership Inference” Contest⁷.

Matthews Correlation Coefficient. The Matthews Correlation Coefficient [39] (MCC) is computed in the following manner:

$$MCC = \frac{\frac{TP}{N} - (S \cdot P)}{\sqrt{(S \cdot P)(1 - S)(1 - P)}}$$

⁷ https://www.iwsec.org/pws/2020/Images/PWSCUP2020_rule_20200826_e.pdf

where

$$\begin{aligned}
 N &= \text{Number of records} & TP &= \text{True positive rate} \\
 FN &= \text{False negative rate} & FP &= \text{False positive rate} \\
 S &= \frac{TP + FN}{N} & P &= \frac{TP + FP}{N}
 \end{aligned}$$

The MCC is preferable to the F1-measure because it is more robust in the situation of class imbalance. This measure is between -1 and 1. The MCC can be generalized to multi-class classification.

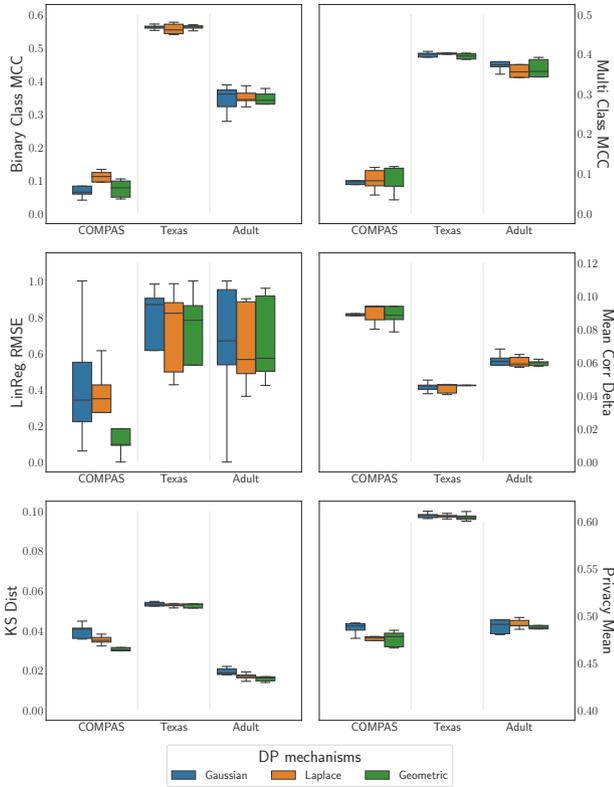


Fig. 7. Impact of the dp-mechanism on the metrics. Obtained with $\epsilon = 0.1$. The linear regression RMSE is normalized. For the MCC, higher is better, for the privacy score, closer to 0.5 is better and for the other metrics, lower is better. Measures are aggregated over a 5-fold cross-validation run.

B Impact of other differential privacy mechanisms.

The Laplacian mechanism is the classical approach to provide differentially-private outputs. However, this mechanism is optimized for continuous values while our implementation currently uses discrete histograms. We investigated two other differentially-private mechanisms

to possibly improve our approach. The first is the Geometric mechanism [26] which is the discrete counterpart of the Laplacian mechanism. The second is the Gaussian mechanism [22], which provides relaxed privacy protection but often a higher query accuracy. We used the implementations of the mechanism from the IBM Differential Privacy Library⁸. For the Gaussian mechanism, the optimized δ is used according to Balle and Wang [7]. Similarly to the study performed by McKay and Snoko [41], we set $\delta = 0.001$.

As shown in Figure 7, our method is generally stable across the three mechanisms. The three mechanisms also seems to offer a similar protection as illustrated by the privacy scores. The Geometric mechanism shows increased performance compared to the other two for the KS distance.

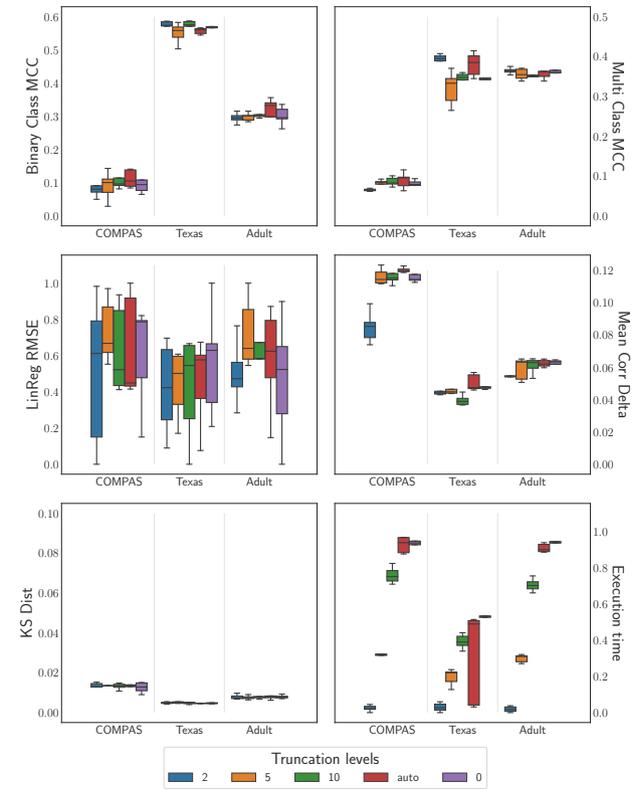


Fig. 8. Impact of the truncation level of the vine on the metrics. Obtained with $\epsilon = 1.0$. The linear regression RMSE and the execution time are normalized. For the MCC, higher is better for the other metrics, lower is better. Measures are aggregated over a 5-fold cross-validation run.

⁸ <https://diffprivlib.readthedocs.io/>

C Impact of truncation level of vine models.

In Figure 8, when $\Psi = \text{'auto'}$, the level of truncation uses the threshold method implemented in the R library that stops the Dissmann's algorithm when all the bivariate copulas are fitted to the independent copula. $\Psi = 0$ means that no truncation is done.

Figure 8 illustrates that a deeper vine model is not necessarily a better model. When truncated to the second tree, vine models exhibit good statistical measures as well as good regression performances. There is no consensus on the classification tasks other than all values of Ψ offer somewhat similar measures.

D Multivariate correlation analysis.

Here we illustrate the difference between the original data and the synthetic data generated from the models. Figure 9 shows the curated synthetic dataset observations used in our analysis. Figures 10, 11, 12 and 13 show respectively the sampled observations from COPULA-SHIRLEY, PrivBayes, DP-Copula and DP-Histograms.

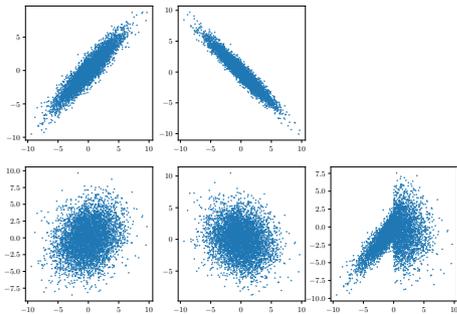


Fig. 9. Reference synthetic data used for the multivariate correlation analysis.

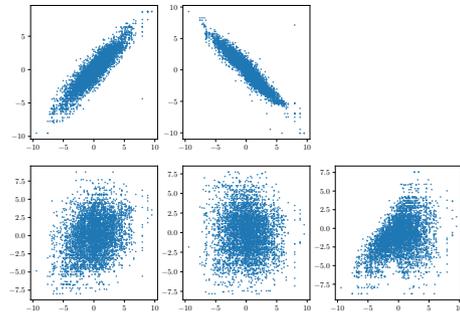


Fig. 10. Observations sampled from COPULA-SHIRLEY.

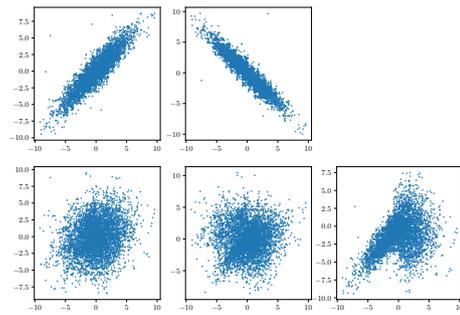


Fig. 11. Observations sampled from PrivBayes.

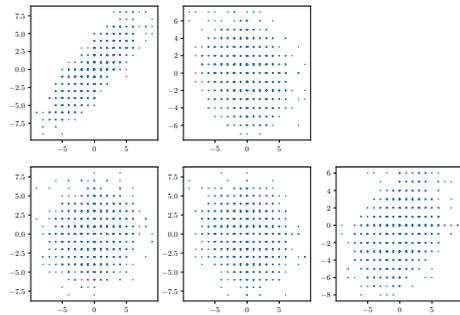


Fig. 12. Observations sampled from DP-Copula.

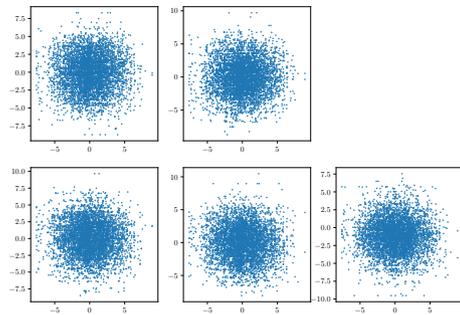


Fig. 13. Observations sampled from DP-Histograms.