Privacy Vulnerabilities in Marginals-based Synthetic Data

Steven Golob

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2024

Committee:

Martine De Cock

Anderson Nascimento

Program Authorized to Offer Degree:
Computer Science and Systems

University of Washington

Abstract

Privacy Vulnerabilities in Marginals-based Synthetic Data

Steven Golob

Chair of the Supervisory Committee:
Martine De Cock
School of Engineering and Technology

Synthetic data generation (SDG) lauds the benefit of augmenting, enhancing, and safeguarding real data, which in many applications is scarce. When acting as a privacy-enhancing technology, SDG aims to exclude any personally identifiable information from the underlying real data, all while maintaining important statistical properties that keep it useful to data consumers. Many SDG algorithms provide robust differential privacy guarantees. However, we show that those that preserve *marginal probability* statistics of the underlying data leak more information about *individuals* than has been previously understood. We demonstrate this by conducting a novel membership inference attack, *MAMA-MIA*, on three state-of-the-art differentially private SDG algorithms: MST, PrivBayes, and RAP. We present the heuristic for our attack on marginals-based SDG algorithms here. It assumes knowledge of auxiliary "population" data, and also assumes knowledge of which SDG algorithm was used. We use this information to adapt the recent DOMIAS attack to MST, PrivBayes, and RAP. Our approach went on to win the international SNAKE challenge in November 2023.

TABLE OF CONTENTS

Page

List of Figures ii

List of Tables iv

Chapter 1: Introduction 1

Chapter 2: DOMIAS Overview 6

Chapter 3: Our Approach 8
3.1 Shadow Modelling 9
3.2 Estimating the Densities ζ on Marginals-based Synthetic Data 10

Chapter 4: Experimental Results 14
4.1 Setup 14
4.2 Evaluation 15
4.3 Results 16

Chapter 5: Conclusion and Discussion 22

Bibliography 25

Appendix A: Supplementary Materials 28
A.1 Datasets 28
A.2 Activation Function 29
A.3 Extended Results 30
A.4 Experimental Parameters 31
A.5 Computational Complexity and Runtime 32

LIST OF FIGURES

Figure Number Page

1.1 Membership Inference Overview. The SDG algorithm takes D_{train} as input, and produces D_{synth}. While an attacker has access to D_{aux} and D_{synth}, D_{train} remains hidden. The goal is to detect which records in D_{target} are included in D_{train}, i.e. D_{member}. 2

2.1 A simple visualization of how DOMIAS detects overfitting. S gives an estimation of the probability distributions of D_{synth} and D_{aux} (left). Normalizing $S(D_{synth})$ by $S(D_{aux})$ exposes overfitting on D_{train} (right). 7

4.1 MA scores for set MI averaged over 30 runs, evaluating our attack and the DOMIAS+KDE attack on synthetic data derived from the SNAKE dataset across different ε. These results are achieved with our smallest configuration, where $|D_{train}| = 100$. Values averaged over *all* size configurations, and on the California dataset are listed in Appendix Tables A.1 and A.2. 17

4.2 Average MA scores for MAMA-MIA and DOMIAS+KDE over all data sizes in Table 4.1. The left two graphs depict membership inference on SNAKE data. The right two show membership inference on California Housing data. The top two graphs present results obtained on set MI experiments. And the bottom two, single MI experiments. 18

4.3 MA results of MAMA-MIA when $|D_{train}|$ =1,000. The higher curve in each graph show the result of our attack when using FPs obtained during shadow modelling in our density estimation ζ. The lower curves show the degraded, yet still strong, performance when arbitrarily-chosen FPs are used in ζ. . . . 19

4.4 Amount of marginals that were chosen by **MST** at least x percentage of the time, with the tick '100%' representing the amount of FPs chosen 100% of the time during shadow modelling. This chart combines FP counts observed using the California and SNAKE data. 19

4.5 Frequencies at which parent sizes were chosen for **PrivBayes** conditionals, combining shadow modelling on the SNAKE and California data. 20

4.6 Average distances from D_{train} to D_{synth} generated by each SDG algorithm, using our summation of Wasserstein Distance for each binarized column. As the privacy-loss budget ε increases, these distances diminish, ostensibly showing improved qualities, with MST yielding the greatest qualities. 21

A.1 Runtime results for MAMA-MIA's density estimation step, over different size configurations of D_{train}. Both axes use a logarithmic scale. 32

LIST OF TABLES

Table Number Page

4.1 Experimental dataset size configurations. Values are spaced evenly on the logarithmic scale. $|D_{member}|$ is always half of $|D_{target}|$, and the ratio of $|D_{target}|$ to $|D_{train}|$ is varied. 15

4.2 Podium results for SNAKE Challenge on **MST** 21

4.3 Podium results for SNAKE Challenge on **PrivBayes** 21

A.1 MA scores for set MI using MAMA-MIA and DOMIAS+KDE on the **SNAKE data**, averaged over 30 runs, and over each size configuration in Table 4.1 (except that experiments on RAP were only run with $|D_{train}| \in \{100, 316\}$). 30

A.2 MA scores for set MI using MAMA-MIA and DOMIAS+KDE on the **California Housing Dataset**, averaged over 30 runs, and over each size configuration from dataset sizes in 4.1 excluding the last size configuration (since D_{aux} only contains 20,640 records) and experiments on RAP were only run with $|D_{train}| \in \{100, 316\}$. 30

ACKNOWLEDGMENTS

I could not have completed this work without the support of my amazing research team, the PPMLHuskies, who stayed attentive during my many redundant presentations. Sikha Pentyala helped me tremendously with experiments, writing, editing, as well as generally coping. Anuar Maratkhan paved the way to our solution by working out the kinks in our initial investigations. And Professor Anderson Nascimento has been extraordinarily generous in mentoring me on many of the concepts fundamental to this work.

I reserve a hearty thank you for Professor Martine De Cock, who, throughout this voyage, has offered me *invaluable* scholarly guidance. As well, her candor continues to inspire me, and her unreserved kindness has been at times the only beacon for my haggard soul.

The financial support I've received as an NSF CSGrad4US Fellow and as a Carwein-Andrews Distinguished Fellow has made this work possible. These benefactors truly embolden me to pursue research that is humanitarian and impactful.

Lastly, my family, Nick, Tiffany, Patty and Tim watered me, gave me nutritious soil, and bathed me in periodic light. Though implausible, I hope to do as much good for others as they've done for me.

Disclaimer This material is based upon work supported by the National Science Foundation CISE Graduate Fellowships under Grant No. CNF2243307. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Chapter 1

INTRODUCTION

Access to quality data is essential for machine learning applications. Yet much of the most useful data from a research perspective is oftentimes the most protected. First in Europe with the adoption of the GDPR[1] in 2016, and more recently in the United States with the inception of the AI Bill of Rights[2], and the Executive Order on AI issued in late 2023 [4], guidelines to protect data privacy have greatly strengthened, and will continue to do so.

Synthetic data is seen as a solution to many of these privacy concerns, keeping sensitive data hidden, while generating artificial data that retains utility similar to the original data for downstream science tasks. There are a whole host of new SDG techniques developed to balance these objectives. Many of them do this by maintaining statistical properties of the real data, while perturbing these statistics with randomness to sever any connection to personally identifiable information (PII).

SDG techniques are even being adopted in industry. For example, "SDG-as-a-service" offered by *Mostly AI*[3], and others, is used by major companies with privacy concerns. Similarly, the U.S. and U.K. governments recently generated synthetic financial and biomedical data for use in a prestigious data science competition[4].

On the other hand, synthetic data with privacy assurances has been shown to leak private

---

[1]European General Data Protection Regulation
`https://gdpr-info.eu/`

[2]The AI Bill of Rights, unveiled by President Joe Biden in October 2022, may become a law in the future.
`https://www.whitehouse.gov/ostp/ai-bill-of-rights/`

[3]`https://mostly.ai/`

[4]`https://petsprizechallenges.com/`

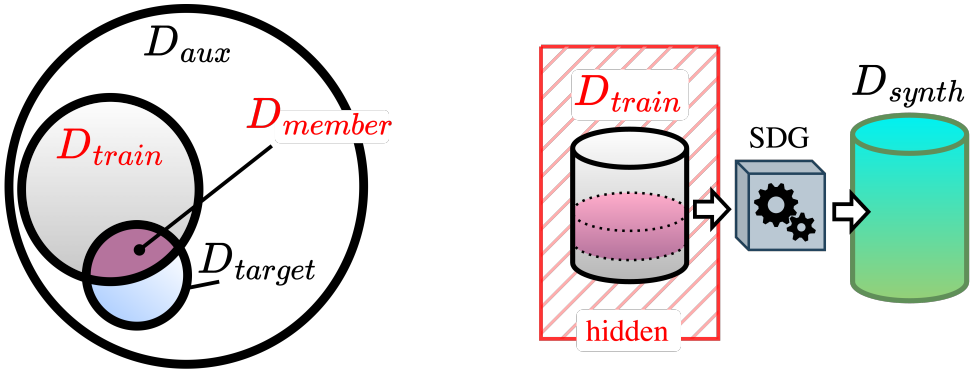

Figure 1.1: Membership Inference Overview. The SDG algorithm takes D_{train} as input, and produces D_{synth}. While an attacker has access to D_{aux} and D_{synth}, D_{train} remains hidden. The goal is to detect which records in D_{target} are included in D_{train}, i.e. D_{member}.

information from the training data, depending on the generation mechanism and the "privacy budget" used [25, 5, 11, 16]. For this reason, academically-motivated attacks have been developed to empirically show which generation techniques protect privacy best.

One fundamental attack is the membership inference attack (MIA). The goal of membership inference is to determine which real data examples were used to train the generator that produced the synthetic records. For example, synthetic data generated from a database of cancer patients may theoretically remove PII. But an MIA might make observations on the synthetic data that give confidence as to whether an individual was present in the training set, allowing the attacker to infer that the individual has cancer. MIAs can be conducted in other contexts too, such as on the output of a machine learning model instead of a generated dataset [22]. But in this thesis, we focus on the former, known as "Inference-on-Synthetic" attacks [15], depicted in Figure 1.1.

Different MIAs can rely on a different set of assumptions for the threat model; some MIAs use a threat model where an adversary knows little. Others assume a generous amount of knowledge in the hands an the adversary (the motivation being to build stronger privacy protections against a stronger opponent). A threat model with "the auxiliary data assumption" admits knowledge of the population data D_{aux} to the adversary [15]. A threat model

with "black-box knowledge of the generator" admits knowledge of the SDG algorithm used to generate the synthetic data D_{synth}, as well as its hyperparameters. We omit a discussion on the "white-box" threat model, where an adversary knows the internal weights and randomness used by the generator. Always, D_{synth} is presumed published and the training data D_{train} is hidden.

A prominent class of SDG algorithms are so-called marginals-based algorithms that preserve estimated marginal probabilities of the training data as their primary way of retaining the resemblance and utility of that data. Three state-of-the-art marginals-based SDG algorithms, MST [18], PrivBayes [27], and RAP [2] have been shown to be strong against MIAs, while providing superior utility [23, 19]. Importantly, unlike several other popular SDG algorithms, these three each provide robust mathematical definitions of their privacy using *differential privacy* (DP) [8]. The view that proper application of DP will quell many of the legality concerns of privacy is becoming a mainstay [3].

The well-known definition of differential privacy states that a randomized algorithm $\mathcal{A}$ with range O is ε-DP if for any two adjacent datasets D_1 and D_2 (datasets that differ in only one entry)

$$Pr[\mathcal{A}(D_1) \in O] \leq e^{\varepsilon} Pr[\mathcal{A}(D_2) \in O] \tag{1.1}$$

In the context of this thesis, $\mathcal{A}$ is a SDG algorithm, and $\mathcal{A}(D_1)$ and $\mathcal{A}(D_2)$ are the synthetic datasets obtained when applying $\mathcal{A}$ to real datasets D_1 and D_2 respectively. DP ensures that the inclusion or exclusion of any entry in the real dataset is obscured, in the sense that any output (synthetic dataset) obtained from computations over the real dataset would have been similarly likely to be reached whether the entry was present in the dataset or not. Differentially private SDG algorithms are, in other words, designed to withstand membership inference attacks. The privacy guarantees of course depend on the privacy budget $\varepsilon \geq 0$, with smaller values indicating stronger privacy guarantees. An ε-DP algorithm $\mathcal{A}$ is usually created out of an algorithm $\mathcal{A}^*$ by adding noise that is inversely proportional to ε. A high-level explanation of how each of these SDG algorithms apply DP noise, and how they work

in general, is given in Chapter 3.

Our main contribution in this thesis is a novel attack *MAMA-MIA* (MArginal Measurement Aggregation based Membership Inference Attack) on synthetic data. We modify the MAMA-MIA slightly for each SDG algorithm discussed here, and we make the case that it can be extended to others that are also based on marginal measurements. It works by exploiting simple, easily-reproducible behaviors of the SDG algorithms. We show that our attack is highly effective at detecting overfitting by the generator, allowing us to learn about individuals in the hidden training data.

As expected, our approach is very successful for high values of ε, and diminishes in efficacy as the privacy-loss parameter is reduced. The threat model we adhere to makes the two aforementioned assumptions, i.e. knowledge of D_{aux} and knowledge of the SDG algorithm used, and we discuss the merit in doing so. Our attacks are enhancements of the recently-proposed DOMIAS MIA [25]. DOMIAS makes use of the auxiliary data assumption to great success, but assumes no knowledge of the SDG algorithm. Our work shows that, with this second assumption, substantially more information about the hidden data can be learned than without it.

Our attacks are simple and efficient. While some MIAs require extensive computation of "shadow" synthetic datasets as a way of understanding the generator's behavior, MAMA-MIA performs *minimal* shadow modelling, discussed in Chapter 3. This is significant because burgeoning privacy laws are more protective against attacks with greater practical feasibility [9, 10].

Lastly, our membership inference heuristic and experiments are motivated by our participation in the international SNAKE (SaNitization Algorithm under attacK ...ε) Challenge [1], where our approach won first place. In this competition, we carried out an MIA on datasets generated by MST and PrivBayes, using a demographic auxiliary dataset. The MAMA-MIA used to win the competition is detailed in this thesis. Additionally, we extend our experiments to a third marginals-based SDG algorithm, RAP, and to a second dataset, which were not a part of the competition, in order to demonstrate MAMA-MIA's generalizability.

We presented a preliminary version of our research at a workshop collocated with the 38th Annual AAAI Conference on Artificial Intelligence:

> High Epsilon Synthetic Data Vulnerabilities in MST and PrivBayes
>
> S. Golob, S. Pentyala, A. Maratkhan, M. De Cock
>
> in: AAAI-24 Workshop on Privacy-Preserving Artificial Intelligence, 2024

Chapter 2

DOMIAS OVERVIEW

The novel MIA proposed recently, DOMIAS [25], outperforms many other approaches by leveraging the auxiliary data assumption. Conceptually, DOMIAS estimates the probability distribution of D_{synth} (using some *density* estimation S), and also of D_{aux}. With these, it simply divides the probability estimation of D_{synth} by that of D_{aux}.

$$\Lambda = \frac{S(D_{synth})}{S(D_{aux})} \tag{2.1}$$

The idea is that, if the generator trained by the SDG algorithm was at all overfit to its training data, then $S(D_{synth})$ should more closely resemble $S(D_{train})$ than $S(D_{aux})$. This discrepancy becomes pronounced in Λ. A member's presence may be inferred when $S(D_{synth})$ at that member's value is high relative to its probability at that value in $S(D_{aux})$.

To demonstrate this visually, without loss of generality, consider a density estimation of one numerical feature for two datasets D_{synth} and D_{aux}, depicted in Figure 2.1 (left). If the synthetic data was overfit to the training data, then normalizing the curve of D_{synth} with D_{aux} (right) makes clear where there may have been a concentration of training data used by the SDG algorithm. In the figure, a target record with value somewhere in the red shaded region would be classified as a member by DOMIAS[1]. Without the auxiliary data assumption, it is more difficult to contextualize observations on D_{synth} and detect overfitting.

[1]In this example the threshold for membership is when $\Lambda = 1.0$. But it needn't be. Normalization allows for any threshold to identify overfitting meaningfully.

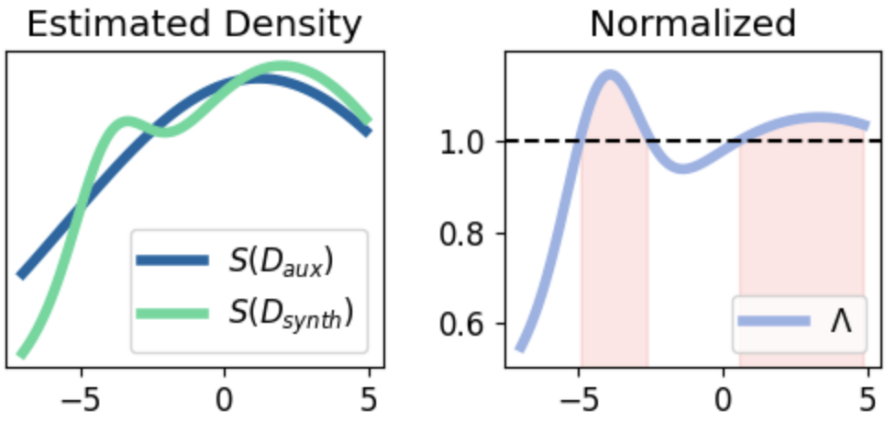

Figure 2.1: A simple visualization of how DOMIAS detects overfitting. S gives an estimation of the probability distributions of D_{synth} and D_{aux} (left). Normalizing $S(D_{synth})$ by $S(D_{aux})$ exposes overfitting on D_{train} (right).

Chapter 3
OUR APPROACH

The success of DOMIAS largely depends on the quality of the density estimation. If poor, the estimated probability distributions won't distinguish D_{synth} from D_{aux}. And in practice, when the datasets are large, or have many dimensions, key discrepancies in their density estimations become overshadowed by noise. Since the DOMIAS attack assumes no knowledge of the SDG algorithm, it uses generic density estimation techniques, such as KDE [21] and BNAF [7].

Our approach assumes black-box knowledge of the SDG algorithm. From this, we leverage basic knowledge of *how* MST, PrivBayes, and RAP preserve marginals in order to design a stellar density estimator for each. We use these instead of generic estimators, and therefor have a much stronger ability to detect overfitting. Applicable to any marginals-based SDG algorithm, our approach is to:

1. *identify* the way in which marginal measurements taken on D_{train} are chosen (we will refer to these measurements as "focal-points")

2. *simulate* the SDG algorithm several times (known as "shadow modelling"), recording the frequency at which focal-points are chosen

3. *aggregate* the frequently-chosen focal-points into a density estimation, ζ, a replacement of Λ in DOMIAS.

Algorithm 1 MAMA-MIA

Input: D_{synth} the synthetic dataset, D_{aux} the auxiliary dataset, F a list of h focal-points, a corresponding list of weights W for focal-points in F, D_{target} the target records.
Output: ζ the list of density estimations of (D_{synth}/D_{aux}) for each $t \in D_{target}$

Let $C_s = \{c_{s,1}, c_{s,2}...c_{s,h}\}$ be the marginals of focal-points in F measured on D_{synth}
Let $C_a = \{c_{a,1}, c_{a,2}...c_{a,h}\}$ be the marginals of focal-points in F measured on D_{aux}
Let $\zeta = [\,]$
for $t \in D_{target}$ **do**
    Let $\zeta[t] = \sum_{i=1}^{h} w_i \cdot c_{s,i}(t)/c_{a,i}(t)$
return ζ

To expound on the first step, MST, PrivBayes, and RAP approximate the joint probability distribution of D_{train} using marginal probability measurements. They select a subset of all marginals because maintaining higher specificity through all marginals is computationally infeasible when D_{train} is even moderately sized. They select which marginal measurements of D_{train}, which we call "focal-points" (FPs), to retain in D_{synth}. They are chosen and measured differently in MST, PrivBayes, and RAP, requiring our density estimation to be implemented slightly differently for each.

These focal-points are highly relevant to an attacker because they show what the generators measured on the hidden D_{train}! The success of our density estimation depends on choosing the same focal-points that the generator chose during its training because attaining such a measurement will give us the clearest possible picture of D_{train}. So determining how MST, PrivBayes, and RAP choose the focal-points is the first step.

All three algorithms are designed to choose focal-points that yield the most amount of of mutual information (or some variant thereof). But in order to adhere to ε-DP, they make these choices indeterminately. This is where we make use of shadow modelling.

3.1 Shadow Modelling

In the second step, we simulate the creation of D_{synth} by running the same SDG algorithm several times, using the same ε and training conditions. Since we do not know the true

D_{train}, we using random samples of D_{aux} as $\hat{D}_{train}$. From this process, we can record which focal-points are chosen, and with what frequency. This step involves obtaining an accurate implementation of the SDG algorithm, and modifying it to record the focal-points. No computation is necessary on any $\hat{D}_{synth}$, which are discarded.

Since we are using all the same parameters, using random samples of D_{aux}, we can develop a confidence of which focal-points were chosen, and then measured by the SDG on the hidden D_{train}. For example, if we notice that MST chose to measure the marginal probability of feature-pair ('age', 'income') 48 times out of 50 runs, we can say that the generator measured this marginal from D_{train} and maintained it in D_{synth} with high likelihood.

3.2 Estimating the Densities ζ on Marginals-based Synthetic Data

Once the oft-chosen FPs are known, we can build for any dataset a density estimation that closely resembles the generator's approximation of the same data. This custom density estimation, ζ, is depicted in Algorithm 1. Lines 1–2 are unique to each SDG algorithm. The overall aggregation of focal-point measurements into ζ is the same for all.

How this aggregation is done is discretionary. Our approach achieves success by summing focal-point measurements from D_{synth} for a particular target's value, divided by the same measurement on D_{aux}. Since the true focal-points are chosen stochastically, we weight each focal-point measurement by its frequency chosen in shadow modelling w_i; if one focal-point is chosen only half the time, we don't exclude it in the density metric, but rather cut its influence in half. This weighting is less important when ε is large and the generator's focal-point choices become more deterministic.

A more theoretically-sound aggregation might be to replace line 5 of Algorithm 1 with

$$g_t = \sum_{i=1}^{h} \begin{cases} \log(c_{s,i}(t)/c_{a,i}(t)) & \text{if } w_i \geq \omega \\ 0 & \text{if } w_i < \omega \end{cases}$$

$$\zeta[t] = e^{g_t} \tag{3.1}$$

where ω is some threshold for how many times the FP f_i was chosen out of 50 runs ($\omega = 40$, for instance). However, we chose our heuristic in Algorithm 1 because the resulting density estimation achieved slightly better attack accuracy.

Once ζ is measured for the targets, we convert it to a probability of membership, $P \in [0, 1]$. Like DOMIAS and other works on MIAs, we omit detailing this activation here, since this step is also discretionary, and may be impractical when the amount of members in $D_{targets}$ is unknown. However we provide a description of the activation we designed in Appendix A.2, since it achieved strong results in the SNAKE Competition.

This density estimation approach can be tailored to similar SDG algorithms that estimate the probability distribution of a training dataset via marginals. It is highly efficient, and operates in linear time with respect to the size of D_{synth} and D_{aux} (we offer an asymptotic analysis in Appendix A.5). We now describe how the focal-points manifest in MST, PrivBayes, and RAP.

3.2.1 MAMA-MIA on MST

MST [18] builds a graphical approximation of the joint probability distribution of D_{train} where the nodes are the features of D_{train}, the edges are the two-way marginal probabilities between two features, and the graph is an undirected tree. During synthesis, MST creates data samples in proportion with the probabilities of all of the marginals measured in the graph.

The edges chosen by MST (i.e. marginals) are the focal-points in our density estimation, so how the graph is constructed is most interesting to us. MST attempts to draw edges that create a maximum spanning tree (hence "MST") based on the mutual information of each feature-pair. But the attempt is inexact because of the differential privacy mechanisms applied to this decision process. Half of the privacy budget is spent on selecting the optimal edges, and half is spent on calculating the marginal measurements themselves.

During shadow modelling, we will observe that the amount of FPs chosen is $h = d - 1$, where d is the number of features in D_{train}, since that's how many edges are in a tree of d

nodes. Algorithm 1, we input the counts as weights W for all focal-points F observed during shadow modelling.

3.2.2 MAMA-MIA on PrivBayes

Our tailored MIA on PrivBayes synthetic data follows a similar approach. PrivBayes [27] also estimates the probability distribution of D_{train} by constructing a graph. Except that, while MST constructs an undirected tree, with exactly $d-1$ edges by default, PrivBayes constructs a *directed* graph, where edges represent important *conditional* probabilities between "child" features and sets of "parent" features.

Like MST, it uses the conditionals-based graphical approximation in order to generate D_{synth}[1], making this approximation highly relevant to an attacker. Edges that yield high mutual information are preferred by PrivBayes, but how many parents are allowed in each conditional depends on ε.

Specifically, when ε is small, PrivBayes reduces k, the maximum number of parents in its graphical approximation. Intuitively, setting k to be large can allow conditionals to more closely approximate the true probability distribution. However, it also means that the high-specificity conditionals are more susceptible to the DP-noise. Throttling back k mitigates this. So entirely different conditionals are tended towards for different ε. We use these conditionals as our focal-points for PrivBayes, and determine which are chosen during shadow modelling, using the correct ε.

3.2.3 MAMA-MIA on RAP

RAP (Relaxed Adaptive Projection) [2], in contrast, does not build a graphical model to estimate the joint probability distribution. Instead, it encodes D_{train}'s features into binary form, then initializes an arbitrary dataset D' of the same dimension. RAP then updates the values in D' until the focal-point measurements resemble those taken on D_{train}.

[1]This is done by sampling values directly from the conditionals, and then following the corresponding path along the directed graph.

RAP's focal-points are called "queries", which are simply k-way marginals on the binarized features, where $k = 3$ by default. Over several iterations, it measures new queries, and re-updates D' using differentiable learning with respect to the errors of those queries. This update happens using Sparsemax (a variant of softmax) to achieve gradient descent [17]. Once RAP is finished updating D', it is projected from a table of floating point values into the binarized domain. This one-hot encoding is then decoded into D_{synth}.

The queries that yield the greatest differences between D_{train} and D' are favored by RAP, in an effort to reduce the maximum error across all queries. The default number q of queries is 50. Note that $q = 50$ is quite small, relative to the $d - 1$ marginals measured by MST, since there are far more possible queries over the binarized domain. Consequentially, the amount of information contained in a marginal on binary features is much less.

So RAP expects the scientist to specify a "workload" – that is, hand-select a subset of features to be considered in the query selection process – using domain- and task-specific knowledge. This greatly narrows the amount of possible focal-points considered, and so poses a challenge for us during shadow-modelling; since we only have access to default information, the FPs we observe will likely be wildly different from those actually measured on D_{train}. Even when default behavior is used to train the generator, as is done in our experiments, the shear amount of FPs considered causes the fitting of D_{synth} to be highly volatile, even when ε is large. This notwithstanding, our density estimation ζ of RAP synthetic data uses the frequencies of queries chosen during shadow modelling, and we use those as the weighted focal-points in Algorithm 1.

Chapter 4

EXPERIMENTAL RESULTS

4.1 Setup

Our experiments are motivated by our participation in the SNAKE Challenge [1], which is framed as follows, and depicted in Figure 1.1. Provided is an auxiliary tabular dataset D_{aux} (the "SNAKE data" is described in Appendix A.1). The SNAKE data consists of twelve categorical features, and three discrete, numerical features, all depicting socio-economic data. Several D_{synth} are generated using the Reprosyn[1] implementations of MST and PrivBayes, using all default parameters. The synthetic datasets are generated for values of $\varepsilon \in \{1, 10, 100, 1000\}$, each using random samples $D_{train} \subset D_{aux}$ as training data. Both D_{synth} and D_{train} contain 10,000 records.

Our goal is to perform membership inference on 100 "targets" $D_{target} \subset D_{aux}$. 50 targets are members of the hidden training data, $D_{member} = D_{target} \cap D_{train}$ and $D_{target} \not\subset D_{train}$, diagrammed in Figure 1.1.

The records can also be grouped into record sets (for example, a record set could be a family of individuals in census data). Record sets have between one and ten records in the SNAKE dataset. Of the 201,279 auxiliary records in the SNAKE data, there are 77,111 sets (families). Membership inference is conducted both on individuals as targets, known as "single MI", and on entire record sets as targets, known as "set MI", where all records in a set are members of D_{train} [13].

We also extend our experiments to continuously-valued data, the California Housing Dataset (see Section A.1) with similar results, and apply our MIA to the authors' imple-

[1] `https://reprosyn.readthedocs.io`

$\lvert D_{train}\rvert$	$\lvert D_{synth}\rvert$	$\lvert D_{target}\rvert$	$\lvert D_{member}\rvert$
100	100	10	5
316	316	26	13
1,000	1,000	64	32
3,162	3,162	158	79
10,000	10,000	398	199
31,623	31,623	1,000	500

Table 4.1: Experimental dataset size configurations. Values are spaced evenly on the logarithmic scale. $|D_{member}|$ is always half of $|D_{target}|$, and the ratio of $|D_{target}|$ to $|D_{train}|$ is varied.

mentation of RAP[2] to legitimize our findings. We augment our experiments with thirteen privacy-loss parameters, $\varepsilon \in \{10^{i/3} | -3 \leq i \leq 9\}$, and with six dataset sizes, listed in Table 4.1.

For each experiment, we run 30 times and average our results. This entails generating our own D_{synth}, with different ε values, from random samples of the auxiliary data. To maintain consistency, we use the same D_{train} for each of the three SDG algorithms in a run. Appendix A.4 provides a more thorough appraisal of our setup. And lastly, we score our membership predictions for D_{target} using "membership advantage", described in Section 4.2.

4.2 Evaluation

In alignment with the SNAKE challenge, we evaluate our predictions P of the targets against the ground truth "membership advantage" (MA) [26], defined as:

$$MA = (tpr - fpr + 1)/2 \tag{4.1}$$

where *tpr*, *fpr* are the true positive rate and false positive rate, computed by weighting individual predictions by their distance from a 0.5 threshold: $2 \cdot |0.5 - p_i|, p_i \in P$. An MA score of 0.5 is as good as random guessing.

[2] `https://github.com/amazon-science/relaxed-adaptive-projection`

For set MI, MA is evaluated on inferences for entire record sets, rather than inferences for individual records. We approach set MI by first making inferences for each individual record in the set, then by simply taking the average. This scored better than other approaches, such as trusting the most confident individual inference of the set, or other trivial weightings.

In order to contextualize our accuracy, we compare all results from using our custom density estimator ζ against the DOMIAS attack using KDE as its density estimator. Since KDE is an estimator of numerical values we encode categorical values from the SNAKE data ordinally.

Additionally, for every run, we compute the distance between D_{train} and D_{synth} generated by the SDG algorithms in order to evaluate the trade-off between the quality of the data and the level of privacy preserved for each algorithm separately. We calculate distance over the binarized form of D_{train} and D_{synth} with a summation of the Wasserstein Distances of each corresponding column in lieu of other distance metrics[3] [20]. By measuring individual columns separately, we render the calculation deterministic.

4.3 Results

Our attacks on MST, PrivBayes, and RAP, revealed remarkable increases in privacy leakage when $\varepsilon > 10$. As shown in Figure 4.1, our attack on synthetic data generated by MST identifies members with almost perfect accuracy when $\varepsilon \geq 100$, and far outperforms DOMIAS+KDE. Our attack on PrivBayes fared somewhat less well, but still showed a clear improvement. However, if ε is set low enough, then all of our attacks, along with our config-

[3]Note that related works suggest that quantifying quality of generative models is inherently application-specific [24, 6, 12]. While other works too evaluate the quality of synthetic data using Wasserstein distance, training machine learning models on both D_{train} and D_{synth}, then comparing their predictive success, has become commonplace [14, 19]. So has comparing arbitrary k-way marginals between the two. For our purposes, training models would have been excessive for the scope of this work, and was more likely to introduce bias. Our aim is simply to provide a modest, reliable distance metric with which to weigh against empirical privacy loss. Furthermore, we deliberately abstained from measuring marginals in our distance computation; since we already know that these SDG algorithms maintain marginal consistency, and since relying fully on marginals is contrary to the argument we are making, we opted to show the relative distances from a different angle.

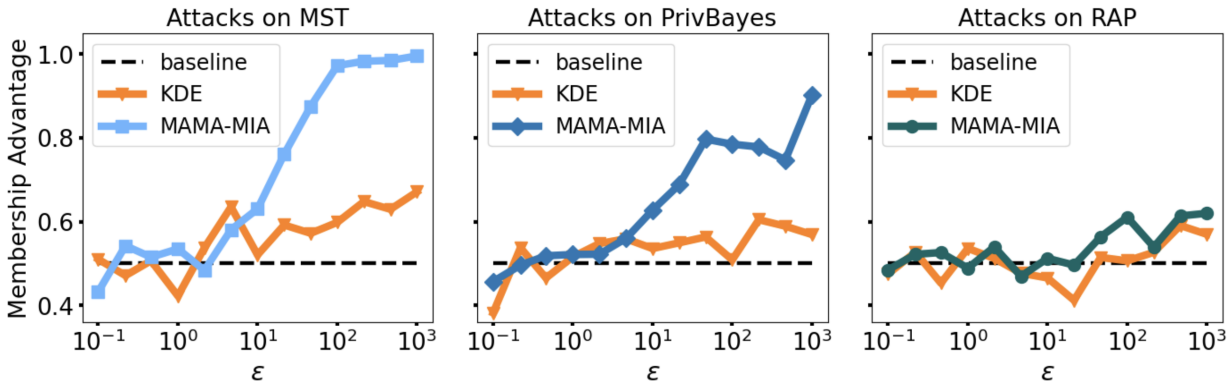

Figure 4.1: MA scores for set MI averaged over 30 runs, evaluating our attack and the DOMIAS+KDE attack on synthetic data derived from the SNAKE dataset across different ε. These results are achieved with our smallest configuration, where $|D_{train}| = 100$. Values averaged over *all* size configurations, and on the California dataset are listed in Appendix Tables A.1 and A.2.

uration of DOMIAS, perform close to random guessing. Our attack on RAP was the least successful, but on average, still outperformed the attack not based on predicting internal marginal measurements.

Our strong performance on MST and PrivBayes is consistent for both set MI and single MI, though set MI yielded greater privacy leakage, shown in Figure 4.2. These graphs and Appendix Tables A.1 and A.2 show how MAMA-MIA generalizes to the California dataset, and to several different sizes of D_{train}, D_{synth}, and D_{target}. But we weren't able to improve results on RAP for larger dataset sizes, or for the California dataset. Fitting D_{synth} to D_{train} without defining a workload is highly volatile, due to limitations discussed in Section 3.2.3. In practice, the FPs chosen during shadow modelling were highly variable, even for large ε. So the following findings will largely omit RAP results.

Our results for MST and PrivBayes were better than anticipated, which we credit in part to our confident identifications of focal-points during shadow modelling. In our investigation, it became apparent that measuring the correct focal-points in our density estimation ζ was critical our the attack's success. See, for example, how much our MA scores degrade when

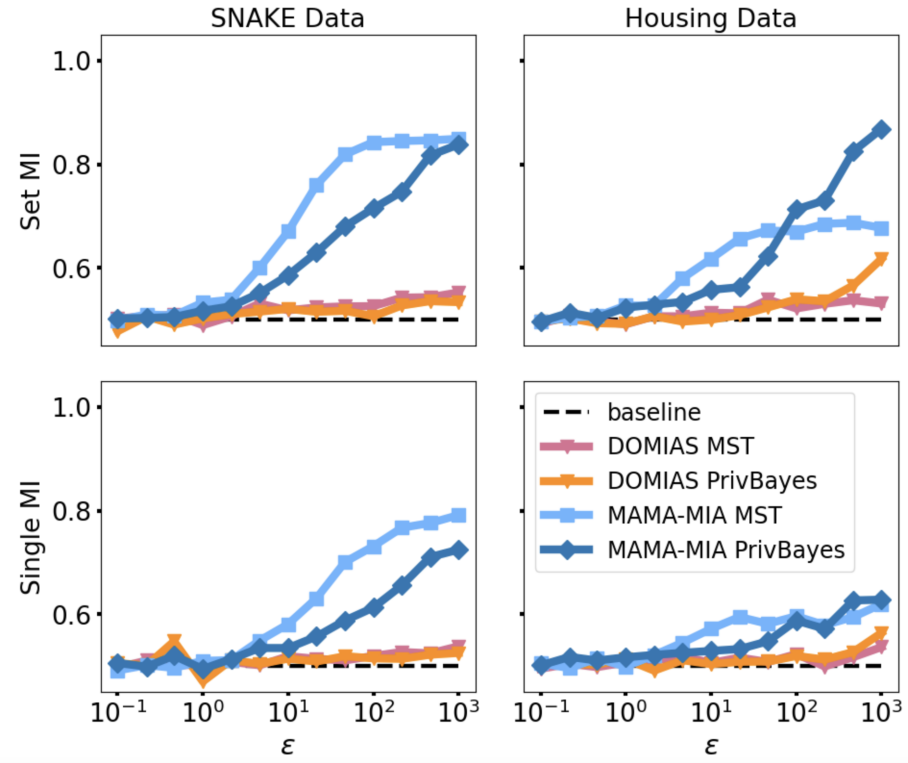

Figure 4.2: Average MA scores for MAMA-MIA and DOMIAS+KDE over all data sizes in Table 4.1. The left two graphs depict membership inference on SNAKE data. The right two show membership inference on California Housing data. The top two graphs present results obtained on set MI experiments. And the bottom two, single MI experiments.

we use *arbitrary* focal-points in ζ, depicted in Figure 4.3. Our density estimation is still more accurate than DOMIAS, but it is substantially less accurate than when we use focal-points predicted in our shadow modelling step, which lends weight to its effect.

As expected, we observed that the variability of FP selection increased as the privacy-loss budget ε decreased. For MST, we visualize these findings as a bar graph in Figure 4.4, where each bar represents a percentage of shadow runs FPs were reselected, across different values of ε. Notice that, when $\varepsilon = 0.1$, most of the marginals selected by MST were only chosen less than 50% of the time, boding poorly for our confidence. On the other hand, with higher ε, a vast majority of marginals were chosen more than 75% of the time during shadow modelling, which allows our density estimation of D_{synth} to be closer to that of the hidden D_{train}.

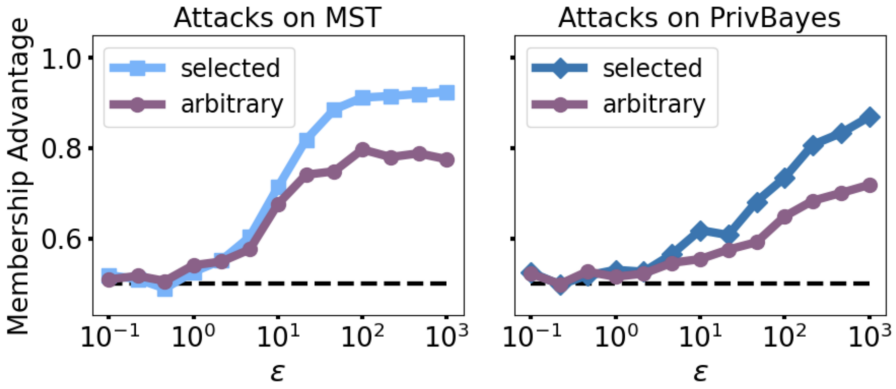

Figure 4.3: MA results of MAMA-MIA when $|D_{train}|$ =1,000. The higher curve in each graph show the result of our attack when using FPs obtained during shadow modelling in our density estimation ζ. The lower curves show the degraded, yet still strong, performance when arbitrarily-chosen FPs are used in ζ.

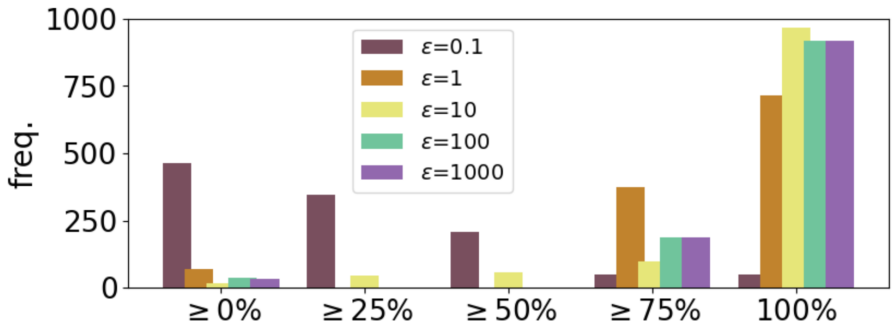

Figure 4.4: Amount of marginals that were chosen by **MST** at least x percentage of the time, with the tick '100%' representing the amount of FPs chosen 100% of the time during shadow modelling. This chart combines FP counts observed using the California and SNAKE data.

Moreover, PrivBayes *preferred* different FPs for different ε. Figure 4.5 shows a histogram of how frequent conditionals' parents sizes were selected, and the trends for different ε. When $\varepsilon = 0.1$, PrivBayes only ever allows for conditionals to have one or zero parents. But when $\varepsilon = 1000$, the graphical estimation is constructed by favoring conditionals with two, three, or four parents. This was expected because of the way PrivBayes changes its maximum parent size k to make the most effective use of its privacy budget.

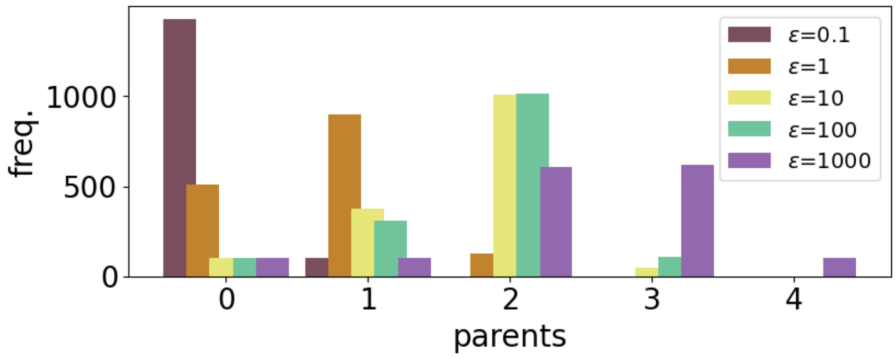

Figure 4.5: Frequencies at which parent sizes were chosen for **PrivBayes** conditionals, combining shadow modelling on the SNAKE and California data.

Next, our difficulty in launching a successful attack on RAP is corroborated by our measures of the high average distances from D_{train} to RAP D_{synth}, which are considerably greater than distances to D_{synth} generated by MST and PrivBayes (shown in Figure 4.6). This distance decreases for all of them when privacy is relaxed. Of the three, MST's D_{synth} had the smallest distance to the training data without fail, which is consistent with other works' conclusions that the utility of MST outperforms PrivBayes, but only slightly [19, 23]. This also aligns with our findings of greater overfitting in MST. The high distance of RAP is caused by using default parameters, rather than hand-selecting a workload, from which RAP would more consistently select FPs as discussed in Section 3.2.3.

For the SNAKE Challenge, we were tasked with conducting set MI on eights sets of targets and synthetic datasets; one for each $\varepsilon \in \{1, 10, 100, 1000\}$, for both MST and PrivBayes. D_{train} and D_{synth} contained 10,000 records, while D_{target} contained 100 *sets* of records, 50 of which were members of D_{train}. Our predictions using this novel attack achieved the highest MA scores for all eight tasks (shown in Tables 4.2 and 4.3), resulting in our team winning the final phase of the competition.

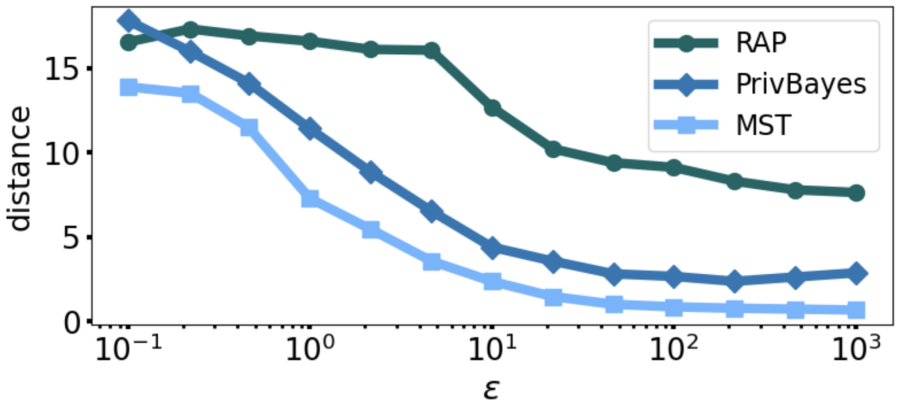

Figure 4.6: Average distances from D_{train} to D_{synth} generated by each SDG algorithm, using our summation of Wasserstein Distance for each binarized column. As the privacy-loss budget ε increases, these distances diminish, ostensibly showing improved qualities, with MST yielding the greatest qualities.

ε	1	10	100	1000
MAMA-MIA	**0.61**	**0.79**	**0.76**	**0.81**
(participant #2)	0.60	0.60	0.55	0.69
(participant #3)	0.60	0.53	0.56	0.54

Table 4.2: Podium results for SNAKE Challenge on **MST**

ε	1	10	100	1000
MAMA-MIA	**0.65**	**0.71**	**0.83**	**0.94**
(participant #2)	0.53	0.62	0.69	0.51
(participant #3)	0.57	0.59	0.61	0.55

Table 4.3: Podium results for SNAKE Challenge on **PrivBayes**

Chapter 5

CONCLUSION AND DISCUSSION

In this thesis, we proposed a new membership inference attack, MAMA-MIA, that is tailored to synthetic data generators that employ marginals. These types of generators are numerous and are generally found to produce the highest quality tabular data. However, this thesis shows that it comes with a previously unseen cost to privacy.

Our attack shows that a substantial amount of *individual* privacy leakage can be retrieved from synthetic datasets when ε is large. Shadow modelling was essential to this end, by bringing the true probability of a generator's focal-point selection into focus. We use this result to build the case that marginals-based SDG algorithms should be re-examined to strengthen privacy protections. We compare our results against the recently proposed DOMIAS MIA, off of which our attack is based. We apply our approach to three SDG algorithms, MST, PrivBayes, and RAP, and verify our results by using two datasets, thirteen values of ε, and six dataset sizes.

The question of choosing ε is certainly application-specific – dependent on a multitude of factors such as the desired utility and characteristics of the dataset. But care must be taken when the application's privacy concerns are significant. Our work also shows how an adversary can use black-box knowledge of the SDG algorithm and which ε used in an attack, if this information is published. Our attack heuristic is general enough that it is likely to do well on similar, marginals-focused algorithms that offer high utility.

Curiously, the accuracy curve of our approach against PrivBayes increases more sharply than for MST, and in some cases, outperforms it. As Figure 4.5 shows, PrivBayes with $\varepsilon = 1000$ chooses mostly multiple-parent conditionals in its graphical estimation of D_{train}. This means that measurements are *highly specific*, and that an attacker's measurement of the

same conditionals may gain this specific knowledge of the hidden data, minus the unknown DP noise added.

On the other hand, MST only ever selects two-way marginals by default, which can only approximate the distribution of D_{train} so well, and explains why our curves for MST on larger datasets mostly level off with increasing ε. Only by constructing a perfect probability distribution could MST and an attacker capture perfect information of D_{train}, which is computationally infeasible when D_{train} is even moderately sized.

To return to our results on RAP, we are even more limited by our threat model. We can't know any specifications in creating D_{synth} by the algorithms that aren't default. In generating quality synthetic data, RAP expects the owners of D_{train} to use domain knowledge to specify a workload, i.e. which *subset* of k-way marginals to consider during FP selection. Without this information, our search is without a compass, and choosing the correct FPs during shadow-modelling is close to random guessing. Insight gained on which FPs would be selected out of*all choices* is both computationally exhaustive and unrepresentative of those apt to be selected from a much smaller, hand-selected subset. But, to some degree, when $|D_{train}| = 100$, MAMA-MIA is still able to do better than random guessing.

So then, off the cuff, this reveals one defense MST can take; MST also provides the option for the scientist to manually select k-way marginals for its estimation of the training data's probability distribution. These hand-picked marginals obviously cannot be determined through shadow-modelling, which only simulate default behavior, and so would weaken an attacker's ability to reconstruct an estimation of the hidden data.

Broader questions of our work can also be raised. Whether or not the black-box knowledge assumption is reasonable is a well-founded one. However, given the open-source nature of these algorithms, and the ease of shadow-modelling, this assumption is well within reason. It is also our belief that opposing a strong adversary motivates the effort to build strong defenses – an effort in which we hope to partake.

Further, the high accuracies we achieved might be questioned if $\varepsilon = 1000$ were considered excessive, or impractical. Given how how noisy these datasets are, we don't consider using

$\varepsilon = 1000$ out of the realm of possibility. But more to the point, the purpose of our work is not to build a practical attack, but rather to pronounce the disparity between successes of prior attacks and what can be achieved using a stronger threat model. A broader investigation into practical SDG techniques will lead to more insight on high ε allowance.

Exciting possible directions to continue research on this topic include designing new density functions to exploit traits of other state-of-the-art synthetic generators, like MWEM-PGM. Or, these could include adapting our heuristic to GAN-based algorithms and algorithms that generate images and types of data other than tabular. An important direction would be to analyze the disparities in privacy leakage for groups underrepresented in the training data when using our approach. There also is the angle further analyzing ε's effect on each algorithm's synthetic data quality, and how our attack's success on each calls for a serious reevaluation of that trade-off. And most importantly, future work includes improving or developing new synthetic data generation techniques that are resistant to membership inference attacks that use this approach.

BIBLIOGRAPHY

[1] Tristan Allard, Louis Béziaud, and Sébastien Gambs. Snake challenge: Sanitization algorithms under attack. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 5010–5014, 2023.

[2] Sergul Aydore, William Brown, Michael Kearns, Krishnaram Kenthapadi, Luca Melis, Aaron Roth, and Ankit A Siva. Differentially private query release through adaptive projection. In *International Conference on Machine Learning*, pages 457–467. PMLR, 2021.

[3] Steven M Bellovin, Preetam K Dutta, and Nathan Reitinger. Privacy and synthetic datasets. *Stanford Technology Law Review*, 22:1, 2019.

[4] Joseph R Biden. Executive order on the safe, secure, and trustworthy development and use of artificial intelligence. *Federal Registrar*, 88:75191–75226, 2023.

[5] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*.

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

[7] Nicola De Cao, Wilker Aziz, and Ivan Titov. Block neural autoregressive flow. In *Uncertainty in Artificial Intelligence*, pages 1263–1273. PMLR, 2020.

[8] 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.

[9] European Parliament and Council of the European Union. Regulation (EU) 2016/679 of the European Parliament and of the Council.

[10] Florent Guépin, Matthieu Meeus, Ana-Maria Cretu, and Yves-Alexandre de Montjoye. Synthetic is all you need: removing the auxiliary data assumption for membership inference attacks against synthetic data. *arXiv preprint arXiv:2307.01701*, 2023.

[11] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019.

[12] 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.

[13] 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.

[14] 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.

[15] Florimond Houssiau, James Jordon, Samuel N Cohen, Owen Daniel, Andrew Elliott, James Geddes, Callum Mole, Camila Rangel-Smith, and Lukasz Szpruch. Tapas: A toolbox for adversarial privacy auditing of synthetic data. In *NeurIPS 2022 Workshop on Synthetic Data for Empowering ML Research*.

[16] James Jordon, Daniel Jarrett, Evgeny Saveliev, Jinsung Yoon, Paul Elbers, Patrick Thoral, Ari Ercole, Cheng Zhang, Danielle Belgrave, and Mihaela van der Schaar. Hide-and-seek privacy challenge: Synthetic data generation vs. patient re-identification. In *NeurIPS 2020 Competition and Demonstration Track*, pages 206–215. PMLR, 2021.

[17] Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*, pages 1614–1623. PMLR, 2016.

[18] Ryan McKenna, Gerome Miklau, and Daniel Sheldon. Winning the NIST contest: A scalable and general approach to differentially private synthetic data. *arXiv preprint arXiv:2108.04978*, 2021.

[19] Mayana Pereira, Meghana Kshirsagar, Sumit Mukherjee, Rahul Dodhia, Juan Lavista Ferres, and Rafael de Sousa. Assessment of differentially private synthetic data for utility and fairness in end-to-end machine learning pipelines for tabular data. *arXiv preprint arXiv:2310.19250*, 2023.

[20] Aaditya Ramdas, Nicolás García Trillos, and Marco Cuturi. On Wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, 19(2):47, 2017.

[21] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[22] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017.

[23] Yuchao Tao, Ryan McKenna, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. Benchmarking differentially private synthetic data generation algorithms. In *AAAI-22 Workshop on Privacy-Preserving Artificial Intelligence*, 2022.

[24] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations*, 2016.

[25] Boris van Breugel, Hao Sun, Zhaozhi Qian, and Mihaela van der Schaar. Membership inference attacks against synthetic data through overfitting detection. *AISTATS*, 2023.

[26] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282, 2018.

[27] 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.

Appendix A

SUPPLEMENTARY MATERIALS

A.1 Datasets

A.1.1 SNAKE Dataset

This is the dataset used in the 2023 SNAKE competition. It is comprised of three numerical features, two finite ordered features, and ten purely categorical features. It holds demographic information. To give a sense, the features include age, state of residency, number of children, marital status, ethnicity, gender, field of profession, weekly hours worked, etc. Each record represents an individual, and individuals are grouped by a household identifier, which we use as record sets during set MI. This dataset has 201,279 records, containing 77,111 households (i.e. record sets).

A.1.2 California Housing Dataset

We use *sklearn*'s sample California Housing Dataset[1], which has been used frequently in machine learning research. The data holds information on residential homes and households by district, collected as part of the 1990 U.S. Census. It contains nine continuously-valued features, and has a total of 20,640 records. It is useful for our purposes to compare against the more categorical-heavy SNAKE data. We create groupings of records arbitrarily to use as sets during set MI experiments.

---

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html`

A.2 Activation Function

Once we compute ζ for the set of targets D_{target}, we convert them to a probability of membership in D_{train}, $P(\zeta) \in [0, 1]$. We achieve this by designing an activation function,

$$P(\zeta) = \frac{1}{(1 + e^{-c(\log \zeta - m)})} \tag{A.1}$$

which maintains the monotonicity of ζ. This is simply a modified sigmoid function, where c is a confidence parameter, defining how far away from probability 0.5 we want our predictions.

This function also maintains that $P(\zeta) = 0.5$ when the densities for D_{synth} and D_{aux} are the same for a given target, which is consistent with the intuition behind how Λ is defined in DOMIAS, Chapter 2. m is the value of ζ for a certain percentile among all of the targets. This percentile is the expected proportion of targets that are actually members. For example, since in all of our experiments exactly half of the targets were members, we set m in $P(\zeta)$ to the median density estimation of the targets. This maps half of the targets to $P > 0.5$, and half of the targets to $P < 0.5$ (except for the median, which is predicted at $P = 0.5$).

Or, in the more realistic setting, when an attacker has no knowledge or expectation of how many targets are members, setting $P(\zeta) = \min(\sqrt[c]{\zeta}/2,\, 1)$ can be a useful mapping of ζ to probabilities, with c similarly acting as a confidence level. This function maintains that when $\zeta > 1$, probability of membership is $> 50\%$, which is still consistent with the intuition behind Λ.

DOMIAS and other works on MIAs stop short of defining an activation function, and score their results using AUC. We define ours here because of the good results it achieved in the SNAKE Challenge, and because our density estimation function ζ is a direct replacement for DOMIAS. But we also recognize that this activation step is discretionary, and may not be useful in real-world attack scenarios where the adversary does not know how many candidates are members to calibrate the function.

A.3 Extended Results

	MST		PrivBayes		RAP	
ε	MAMA-MIA	KDE	MAMA-MIA	KDE	MAMA-MIA	KDE
.1	0.50	0.50	0.50	0.48	0.48	0.48
.22	0.51	0.51	0.50	0.51	0.54	0.54
.46	0.51	0.51	0.50	0.49	0.50	0.48
1	0.53	0.49	0.52	0.50	0.47	0.54
2.15	0.54	0.51	0.53	0.51	0.54	0.52
4.64	0.60	0.53	0.55	0.52	0.50	0.50
10	0.67	0.52	0.59	0.52	0.51	0.51
21.54	0.76	0.52	0.63	0.51	0.53	0.45
46.42	0.82	0.53	0.68	0.52	0.54	0.53
100	0.84	0.53	0.71	0.51	0.60	0.51
215.44	0.84	0.54	0.75	0.53	0.55	0.52
464.16	0.85	0.54	0.82	0.54	0.58	0.58
1000	0.85	0.55	0.84	0.53	0.59	0.54

Table A.1: MA scores for set MI using MAMA-MIA and DOMIAS+KDE on the **SNAKE data**, averaged over 30 runs, and over each size configuration in Table 4.1 (except that experiments on RAP were only run with $|D_{train}| \in \{100, 316\}$).

	MST		PrivBayes		RAP	
ε	MAMA-MIA	KDE	MAMA-MIA	KDE	MAMA-MIA	KDE
.1	0.50	0.49	0.50	0.50	0.52	0.53
.22	0.50	0.51	0.51	0.50	0.51	0.50
.46	0.51	0.50	0.50	0.49	0.49	0.50
1	0.53	0.49	0.52	0.49	0.48	0.48
2.15	0.53	0.51	0.53	0.51	0.52	0.56
4.64	0.58	0.51	0.53	0.50	0.46	0.49
10	0.62	0.51	0.56	0.50	0.48	0.54
21.54	0.66	0.51	0.56	0.51	0.48	0.52
46.42	0.67	0.54	0.62	0.52	0.47	0.52
100	0.67	0.52	0.71	0.54	0.52	0.48
215.44	0.68	0.53	0.73	0.54	0.53	0.51
464.16	0.69	0.54	0.82	0.57	0.56	0.54
1000	0.68	0.53	0.87	0.62	0.54	0.51

Table A.2: MA scores for set MI using MAMA-MIA and DOMIAS+KDE on the **California Housing Dataset**, averaged over 30 runs, and over each size configuration from dataset sizes in 4.1 excluding the last size configuration (since D_{aux} only contains 20,640 records) and experiments on RAP were only run with $|D_{train}| \in \{100, 316\}$.

A.4 Experimental Parameters

For each attack, shadow-modelling consisted of 50 runs. For each membership experiment described in this thesis, results were averaged over 30 runs. The results shown in Figure 4.1 are based on set MI, conducted on SNAKE data, with $|D_{train}| = 100, |D_{synth}| = 100$, with ten target record sets, five of which members. The results in Figure 4.2 are an average of results from experiments run with data size configurations listed in Table 4.1. All of these values are evenly spaced on the logarithmic scale, while the proportion of D_{target} to D_{train} changes, to stage different inference challenges. (We omit experiments on the California Housing Dataset using the last configuration because it only contains 20,640 records.)

Since MST, PrivBayes, and RAP operate on discretely valued data, we segment continuous values into ten buckets. This was a conservative choice, since finer granularity of buckets would allow the marginals to better approximate the joint probability distribution, and explains why our results were better on the SNAKE data, which has much more than ten values for most of its features.

During set MI experiments, sets in D_{target} contain at least four records. When we compare MA accuracies of using ζ with FPs found in shadow-modelling versus using ζ with arbitrary FPs, shown in Figure 4.3, we fairly select the same size and amount of focal-points as the generators do by default during synthesis. But otherwise the feature combinations are randomly selected, as if we hadn't shadow-modelled.

The results on RAP from Figure 4.1 deviate from the default parameters, in that, instead of selecting 50 queries per epoch, we select 70 queries per epoch, and increase the maximum possible updates per epoch to 2600 from 1000. We do this as a counter measure to induce fitting on the chosen focal-points, selected from a much larger domain of queries (773,239 possible for the binarized SNAKE data), rather than fitting to a specified workload as RAP intends. Otherwise, all default values for MST, PrivBayes, and RAP are used in our experiments.

All experiments are conducted on an Apple M2 Max chip with 64GB of memory.

A.5 Computational Complexity and Runtime

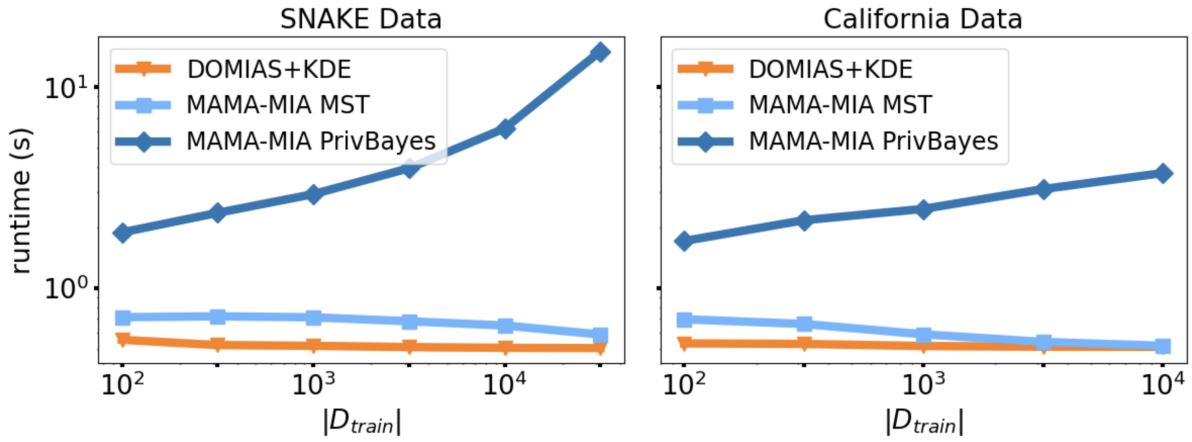

Figure A.1: Runtime results for MAMA-MIA's density estimation step, over different size configurations of D_{train}. Both axes use a logarithmic scale.

The computational costs of our approach can be broken down into two stages. The first stage is where we conduct shadow modelling to determine which focal-points to use in our density estimation. We omit an analysis of the complexity of the SDG algorithms themselves, but can think of them each as some function of the size of the sample training data $\mathcal{O}(f(|\hat{D}_{train}|))$. Therefore, if we simulate the algorithm u times, then the complexity of this stage is simply $\mathcal{O}(u \cdot f(|\hat{D}_{train}|))$, since the only computations we add are constant-time steps of recording which focal-points were chosen.

The second stage is where we perform the density estimation to calculate membership predictions for D_{target}. For each tuple in MST and RAP, we construct a marginal probability table for D_{aux}, D_{synth}. We make the reasonable assumption of the ability to use amortized constant-time hash tables during this construction, and so constructing these takes linear time with respect to the size of the datasets, at most size n. Then for each record in D_{target}, we look up the probability of its value. Together these steps for one tuple amount to $\mathcal{O}(|D_{aux}| + |D_{synth}| + 2|D_{target}|)$ time, which is just $\mathcal{O}(n)$, since $D_{target} \subseteq D_{aux}$, and assuming

$|D_{synth}| < |D_{aux}|$. This is also the case for PrivBayes, where we build conditional probability tables instead. Like marginal tables, constructing conditional tables with hash tables requires looking at each feature in the tuple for each record once, since the number of features in each conditional is practically small.

We note the worst case, if our hash tables fail, where runtime of building a probability table for a tuple as $\mathcal{O}(n \cdot \prod l)$, assuming for simplicity's sake that each feature has l possible values. This does manifest somewhat when attacking PrivBayes on SNAKE data, which has a larger domain than our discretized California Dataset, and so stresses hashing capability. These average runtime performances are shown in Figure A.1. The seemingly superlinear behavior of PrivBayes may be explained by the added layer of hashing required for computing conditionals. However, since both axes are scaled logarithmically, this actually misrepresents how linear the PrivBayes runtime results appear graphically. Measurements are taken in seconds (s), and taken only on the density estimation step. However, even for D_{aux} containing over 200,000 records, computation happens in a matter of seconds. We can also explain MST's seemingly constant runtime by the fact that the $\mathcal{O}(n)$ runtime is dominated by size of D_{aux}, which doesn't change between runs, and is far larger than every D_{synth} size processed.