

Imperial College London
Department of Computing

**Using generative models to design more
realistic privacy adversaries against data
release mechanisms**

Guépin Florent

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of the Imperial College London,
January 2025

Abstract

The demand for sensitive and proprietary datasets has been rising as companies, researchers, and government agencies across the world increasingly rely on their collection. They are used in a diverse variety of applications, from training machine learning models to computing aggregate statistics. Hence, it is crucial to ensure that the privacy of the data is protected. However, ensuring in practice that a data release mechanism preserves the privacy of every record in its dataset, as well as providing an accurate assessment of the privacy risk, is difficult. In practice, this privacy evaluation is done through adversarial evaluation, but those privacy evaluation adversaries are often relying on access to an auxiliary dataset with the same underlying distribution as the dataset under attack. In this thesis, we aim to provide a consistent identification of the privacy risks of data release mechanisms and to develop more realistic privacy evaluation, in two major contributions.

In our first contribution, we (a) explore methods to more accurately, in the context of synthetic generators, identify at-risk records from a given dataset. And (b) we present why averaging out the reported risks over multiple test datasets to evaluate the privacy risk might give a wrong assessment of the privacy risk.

In our second contribution, we study the relaxation of the auxiliary assumption when performing privacy attacks by employing generative models. First, we study how to relax this assumption in the context of membership inference attacks against synthetic data generators. Second, in the context of membership inference attacks against aggregation techniques, commonly used for releasing spatio-temporal information such as trajectories. Third, we study a new type of leakage, correlation inference attacks. We show how correlations can be used as building blocks to relax the auxiliary assumption in the context of attribute inference attack against machine learning models.

Acknowledgements

I would like to take this opportunity to express my utmost gratitude to Pr Yves Alexandre de Montjoye for this amazing PhD opportunity at Imperial College London. I learned a lot alongside him on this journey, especially critical thinking and scientific communication skills.

I would like to also take the time to thank all my Imperial collaborators: Ana-Maria Crețu, Matthieu Meeus, Vince Guan and Nataša Krčo. They made me grow as a researcher, and I am really grateful for that. I am especially immensely grateful toward Ana-Maria Crețu, for all the support, the time and the work we did together. I would not forget to mention all my colleagues for the insightful discussion in the office, and the time spent together that made my time at Imperial College feels great, even through the most stressful times: Shubham Jain, Florimond Houssiau, Andrea Gadotti, Luc Rocher, Arnaud Tournier, Bozhidar Stevanoski, Yuhan Liu, Yifeng Mao, Yao Zexi. I am also grateful for the opportunity to co-supervised the master thesis of the following students: Pablo de Dios Aguilar and Yolanda Yang.

To pave my way toward thinking about doing a PhD, I would like to thank all the people that believed in me and pushed me during my studies. From high school, where I met dedicated mentors, thank you Alexandre Picaud, Samuel Laurent, Jérôme Colinet, Caroline Touquet and Jean-Pierre Schmitt. From French preparatory classes, I would like to thank my teachers, especially my professor of computer science Serge Grunenberger. I would like to emphasize the gratitude I have toward Thierry Guérillot, who always believed in me. From my graduate years, I would like to special thanks Damien Sthelé for his support in my admission at Imperial, and Guillaume Hanrot for being my mentor during those years.

None of this would have been possible without the incredible support from my whole family and friends. Mehmed, Ersan, Irfan, Ersin, Emilien, Hugo, Clément L., Célia, Clément V., Rodolphe, Martin, Dino, Quentin, Paul, Jean-Paul, Simon and Yannis, thank you for all the great time we had. I'd like to give a special thanks to Florian, thank you for all the insightful discussions, and for the numerous games and project we shared, which allowed me to escape from the pressure. To my mother in law Fatima, thank you for the constant encouragement, it helped me a lot to stay on course with the PhD. To my godfather Frederic and god mother Jeannette, thank you for the incredible support during my academic journey. Thank you Claude and Marie-Odile for your perpetual backing, the calls that helped me stay on my PhD course. I couldn't have

accomplished what I did without the kindest help and the most compassionate assistance of Françoise and Didier. There are no words that can describe how much I am grateful to all the assistance you provided me, tirelessly - even at 5AM, and continually over all those years. I'd not forget to give a special thank to Salem, for the incredible –thus unknowingly– support and attention.

To my brother, Aloïs, from whom the calm strength and the many plays of strategic games provided me great comfort. To my sister, Suzanne, from whom I learned being pragmatic, and supported me while being miles away. To my parents, Stéphane and Marie-Desle, who were there, always, to stand by my decisions and to support me every step of the way. You both are models for me, and I would not have made it this far without you. I want to say thank you to you all for being there for me continuously.

I cannot emphasize enough how crucial the support from the love of my life, Sabrina, has been during this journey. Through my darkest time you have been there, giving me support when I needed it the most. This PhD was an odyssey from which you were my compass.

Dedication

This thesis is dedicated to my grandparents, my parents, my brother and sister, my fiancée Sabrina and my son Ilyès.

‘Unfortunately what is little recognized is that the most worthwhile scientific books are those in which the author clearly indicates what he does not know; for an author most hurts his readers by concealing difficulties.’

‘Malheureusement, ce qui est peu reconnu, c’est que les livres scientifiques les plus valables sont ceux où l’auteur indique clairement ce qu’il ne sait pas. Car un auteur blesse le plus ses lecteurs en dissimulant des difficultés.’

Evariste Galois

Statement of Originality

I declare that this thesis is composed by myself and that all the work presented in this thesis is mine, except where otherwise stated. I certify that all material that is not my own work, or part of a collaboration, has been properly acknowledged.

Florent Guépin

Copyright Declaration

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC). Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose. When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Contents

Abstract	i
Acknowledgements	ii
1 Introduction	1
1.1 Motivation	1
1.2 Contributions	5
1.3 Thesis outline	7
1.4 Publications	8
2 Background theory	10
2.1 Synthetic distributions	10
2.1.1 Synthetic datasets	10
2.1.2 Generative models	11
2.1.3 Outliers	12
2.2 Privacy attacks	13
2.2.1 Shadow modeling	13
2.2.2 Membership inference attack	13

2.2.3	Attribute inference attack	14
2.3	Differential privacy	16
2.3.1	Definition	16
2.3.2	Differentially private generator	16
3	Vulnerable record identification in synthetic data publishing	18
3.1	Introduction	19
3.2	Related work	21
3.3	Identifying vulnerable records	23
3.4	Experimental setup	25
3.4.1	State-of-the-art MIA against synthetic tabular data	25
3.4.2	A new MIA against synthetic tabular data	26
3.4.3	Baselines for identifying vulnerable records	28
3.4.4	Parameters of the attack	29
3.5	Results	31
3.5.1	Performance of vulnerable record identification methods	31
3.5.2	Applicability of the method to the target attention attack	31
3.5.3	Robustness analysis for k	32
3.5.4	Robustness analysis for the cosine distance	33
3.5.5	Vulnerable records and differentially private generators	34
3.6	Discussion	35
3.7	Conclusion	36

4	A better evaluation of the vulnerability score for machine learning models	38
4.1	Introduction	38
4.2	Related work	40
4.3	MIA evaluation setups	42
4.3.1	Notation	42
4.3.2	Average evaluation setup	42
4.3.3	Specific evaluation setup	43
4.3.4	Comparing setups	44
4.4	MIAs against generative models	46
4.4.1	Adult and $\mathcal{M}_{target}$ Synthpop	48
4.4.2	Other generators and datasets	49
4.5	MIAs against machine learning classifiers	50
4.6	Theoretical results	51
4.7	MIA by a strong adversary	54
4.8	Conclusion and limitations	55
5	Removing the auxiliary assumption for MIAs against synthetic generators	58
5.1	Introduction	59
5.2	Attack scenarios	61
5.2.1	(S0) Auxiliary	62
5.2.2	(S1) Black box	62
5.2.3	(S2) Published	63
5.2.4	(S3) Upper bound	64

5.3	Experimental setup	65
5.3.1	Meta-classifier methods	66
5.3.2	Parameters of the attack	66
5.4	Results	67
5.4.1	Query based attack	67
5.4.2	Target attention attack	69
5.4.3	Robustness analysis for number of synthetic records m	71
5.5	Future work	71
5.5.1	Impact of releasing less synthetic records	71
5.5.2	Differentially private synthetic generation methods	72
5.5.3	Bridging the gap with the upper bound	73
5.6	Conclusion	73
6	A Zero Auxiliary Knowledge Membership Inference Attack on Aggregate Location Data.	75
6.1	Introduction	75
6.2	Definitions and threat model	79
6.2.1	Location Traces and Aggregates	79
6.2.2	Privacy measures on location aggregates	80
6.2.3	Problem formulation	83
6.2.4	Membership classifier	83
6.2.5	Threat model	84
6.3	Related work	85

6.4	Methodology	87
6.4.1	Zero auxiliary knowledge MIA framework	87
6.4.2	Generating synthetic traces from aggregate location data	88
6.4.3	Obtaining accurate marginals	89
6.4.4	Paired sampling for training	101
6.5	Experimental setup	104
6.5.1	MIA implementation	104
6.5.2	Evaluation	105
6.6	Experimental results	106
6.6.1	Suppression of Small Counts	106
6.6.2	ε -DP noise addition	109
6.6.3	Partial knowledge of the target trace	111
6.6.4	Paired Sampling vs. Independent Sampling	113
6.6.5	Removing the time knowledge over traces	114
6.7	Discussion	116
6.7.1	Analysis of results: practical risk of MIA	116
6.7.2	Proposed mitigations	119
6.7.3	Limitations	120
6.7.4	Generalizations to MIAs in other settings	121
6.8	Conclusion	122
7	Correlation inference attacks against machine learning models.	124
7.1	Introduction	124

7.2	Results	128
7.2.1	Correlation inference threat model	128
7.2.2	Attack methodology	131
7.2.3	Empirical evaluation	140
7.2.4	Attribute inference attack	151
7.3	Discussion	157
7.4	Materials and methods	161
7.4.1	Correlation inference attack	161
7.4.2	Attribute inference attack	166
7.5	Related work	167
8	Conclusion	171
8.1	Summary of thesis achievements	171
8.2	Limitations and future work	173
	Bibliography	175
A	Algorithms	App. 1
A.1	Algorithm in Chapter 7	App. 1
B	Presentation of datasets	App. 2
B.1	Presentation of the datasets used in Chapter 3,4 and 5	App. 2
B.2	Presentation of the datasets used in Chapter 6	App. 3
B.3	Presentation of the datasets used in Chapter 7	App. 4
C	Parameters and notations used throughout experiments	App. 5
C.1	Parameters used in experiments of Chapter 3	App. 5

C.2	Parameters used in experiments of Chapter 4	App. 6
C.3	Parameters used in experiments of Chapter 5	App. 6
C.4	Notations used in Chapter 6	App. 6
C.5	Parameters used in experiments of Chapter 7	App. 7
D	Additional results	App. 8
D.1	Additional results for Chapter 7	App. 8

List of Tables

3.1	Mean and standard deviation for AUC for the query-based attack method	30
3.2	Mean and standard deviation for AUC for the target attention method	32
4.1	CPU time to run each of our experiments in Chapter 4 on a single record	48
4.2	Miss rate across different datasets and synthetic data generators for AUC and TPR	50
4.3	Miss rate and RMSE across different datasets for $\mathcal{M}_{target}$ ML classifier for AUC and TPR	51
4.4	$\rho(R_{avg}^{\phi}, R_{sp}^{\phi})$ and $\mathcal{V}(R_{avg}^{\phi}, R_{sp}^{\phi})$ values. Values are presented across different datasets, ML models and synthetic data generators for AUC and TPR	53
5.1	MIA accuracy results (mean and standard deviation for 10 target records). This results are presented across datasets and generators, for the query based attack.	68
5.2	MIA accuracy results (mean and standard deviation for 10 target records). This results are presented across datasets and generators, for the target attention attack.	71
6.1	Mean accuracy scores with standard error for KK and ZK over various suppression thresholds. Results are computed on size 1000 aggregates from the CDR and Milan datasets across various suppression thresholds k	107

6.2	Mean accuracy scores with standard error for KK and ZK for event level DP. Results are computed on size 1000 aggregates from the CDR and Milan datasets across various privacy budgets ε for event level DP.	110
6.3	Mean accuracy scores with standard error for KK and ZK for user-day level DP. Results are computed on size 1000 aggregates from the CDR and Milan datasets across various privacy budgets ε for user-day level DP.	110
6.4	Mean AUC scores with standard error for KK and ZK without time knowledge and greedy approach. This results are computed on raw aggregates of size $m = 1000$ when the adversary only knows some of the target's visited ROIs and employs the greedy approach of assuming that the target visits each known ROI during every epoch.	115
6.5	Mean AUC scores with standard error for KK and ZK without time knowledge and random sampling approach. This results are computed on raw aggregates of size $m = 1000$ when the adversary only knows some of the target's visited ROIs and employs the random sampling approach of distributing random visits uniformly across each known ROI.	116
6.6	Mean AUCs of KK and ZK MIAs on the CDR dataset with varying m.	118
6.7	Mean AUCs of KK and ZK MIAs on the Milan dataset with varying m.	118
7.1	Results of our correlation inference attacks on three real-world datasets.	151
7.2	Comparison between CI-AIA and other attribute inference attacks on Fifa19. We report the attack accuracy averaged over 1000 runs (with 95% confidence interval).	155
7.3	Comparison between our CI-AIA and attacks obtained by substituting Jayaraman and Evans's [JE22] prediction rule to step 3 of CI-AIA. We report the attack accuracy averaged over 1000 runs (with 95% confidence interval) on the Fifa19 dataset.	157

C 1	Parameters used throughout experiments of Chapter 3.	App. 5
C 2	Parameters used throughout experiments of Chapter 4. N_{shadow} represents the number of shadow models used to train the MIA, and N_{eval} represents the number of evaluation models used.	App. 6
C 3	Parameters used throughout experiments of Chapter 5.	App. 6
C 4	Glossary of notations used in Chapter 6.	App. 7

List of Figures

1.1	Focus of each chapter in the generic privacy attack pipeline. In this figure, we chose to present an adversary using the shadow modeling technique, as it is typically done in the literature [AMS ⁺ 15, SSSS17].	5
3.1	Illustration of our target attention architecture.	28
3.2	AUC of MIAs across record identification techniques for the query-based attack , for datasets UK Census and Adult and synthetic data generators BayNet and Synthpop.	31
3.3	AUC of MIAs across record identification techniques for the target attention attack , for datasets UK Census and Adult and synthetic data generators BayNet and Synthpop.	32
3.4	Mean and standard deviation of the AUC of MIAs for 10 records selected by the vulnerability score in various settings. We vary in (a) the number of neighbors considered and in (b) the distance metrics considered. Results on Adult, Synthpop for the query-based attack.	33
3.5	Mean and standard deviation of the AUC of the MIAs for varying values of ϵ in DP-based generator PrivBayes. Results on UK Census (left) and Adult (right) for the query-based attack and for the 10 points for each record identification method.	35

4.1	Presentation of the evaluation setups. On the left: the average evaluation setup, sampling multiple datasets from the evaluation pool D_{eval} and training an evaluation model on each. On the right: the specific evaluation setup: the only evaluation datasets used are the full D_{target} , in which x_{target} is included, and D_{target} in which x_{target} is replaced by a reference record x_R sampled from the evaluation pool D_{eval}	44
4.2	Standard deviation of the AUC as a function of the number of evaluation models used to calculate AUC Adult dataset.	47
4.3	Presentation of the privacy risks per record. The privacy risk is computed for each record in D_{target} sampled from the Adult dataset, $\mathcal{M}_{target}$ Synthpop, and $\phi = AUC$. (a) per-record specific and average risks. The dashed grey lines mark the high-risk threshold $t = 0.8$. The shaded grey area marks all the high-risk records missed in the average setup. (b) histogram of per-record absolute differences between the specific and average risks.	49
4.4	AUC results for the classic and stronger attack for 500 points randomly sampled from D_{target}, for both the average and the specific setup. . . .	55
5.1	Comparison of MIA accuracy for the query based attack method across 4 different scenarios. It present results accross S0, S1, S2 and S3, for both generators Synthpop and BayNet. Figure (a) shows results for UK Census, while figure (b) displays results for Adult.	69
5.2	Comparison of MIA accuracy for the target attention attack method across 4 different scenarios. We consider all scenarios S0, S1, S2 and S3, for both generators Synthpop and BayNet. Figure (a) shows results for UK Census, while figure (b) displays results for Adult.	70
5.3	Mean and standard deviation of MIA accuracy in scenarios S1. This scenario give a black-box access for varying number m , the number of synthetic records available to the adversary. Results for BayNet and the query-based attack using (a) UK Census (b) Adult.	72

6.1	Adversary's prior knowledge presented for both scenario of Chapter 6. This figure present adversary's prior knowledge in the previous work, Knock-Knock MIA [PTDC17] (left), and our work, Zero Auxiliary Knowledge MIA (right). The ZK adversary does not require knowledge of location traces of real people to run the MIA.	79
6.2	Application of aggregation for location data with various protections. This is an example of how suppression of small counts and differential privacy may be applied to an aggregate with 3 ROIs (rows) and 3 epochs (columns). . .	82
6.3	Architecture of our Zero Auxiliary Knowledge MIA attack. <i>Adv</i> first creates synthetic traces, then uses them with the partial target trace to train the membership classifier, before predicting membership.	87
6.4	Statistical parameters needed in entry to generate synthetic location traces. The four statistical parameters for the unicity model by Farzanehfar et al. [FHdM21] include marginal distributions in Figures 6.4(a)-6.4(c) (the dataset's true marginal distributions are shown in red) and the Delaunay triangulation of ROIs in Figure 6.4(d).	88
6.5	Log compression for SSC aggregates. SSC biases the estimate obtained from the aggregate by creating more extreme values. The true time marginal from the CDR dataset (plotted for the first week) is better approximated after the empirical estimate from the aggregate ($m = 1000$, $k = 1$) undergoes log compression $\log(1 + \gamma x)$ with γ chosen as in (8).	91
6.6	Time marginal approximation. Time marginal approximation from a raw aggregate over $m = 1000$ users from the CDR dataset.	92
6.7	Power transformation for DP aggregates. DP noise compresses the estimate obtained from the aggregate. The true space marginal from the CDR dataset (organized by popularity) is better approximated after the empirical estimate from the aggregate ($m = 1000$, $\frac{\Delta}{\epsilon} = 1$) undergoes power transformation x^p with p selected according to Algorithm 7.	93

- 6.8 **Representation of the space and time marginals obtained from ε -DP aggregates over $m = 1000$ users from the Milan dataset.** They are plotted for different noise scales $\frac{\Delta}{\varepsilon}$. Interestingly, the distribution does not converge to a uniform distribution as the noise scale increases, due to the increasing variance of $Lap(\frac{\Delta}{\varepsilon})$ 99
- 6.9 **Exponential estimate for activity marginal.** Once the mean number of visits is approximated, the activity marginal from the CDR dataset is best approximated by a lognormal distribution with an optimal skew parameter. However, it can also be approximated without additional parameters by an exponential distribution. 100
- 6.10 **AUC of the Zero Auxiliary Knowledge MIA for different number of synthetic traces.** This results are generated in the setting $m = 1000$, $k = 1$, and $\varepsilon = 1$ -DP at the event level. 105
- 6.11 **Results comparison between ZK and KK while varying suppression thresholds.** Mean AUC scores with standard error for ZK and KK on size 1000 aggregates with various suppression thresholds 107
- 6.12 **The percentage of nonzero entries that are suppressed in a size $m = 1000$ aggregate.** We plot this percentage after undergoing SSC with threshold k . 108
- 6.13 **Result comparison between our ZK-MIA and KK-MIA against DP-aggregates.** Mean AUC scores with standard error for ZK and KK on size 1000 aggregates with various privacy units and budgets ε 111
- 6.14 **Result comparison between our ZK-MIA and KK-MIA with only the knowledge of a fraction of the trace.** Mean AUC scores with standard error for ZK and KK on size 1000 aggregates with $\varepsilon = 1$ event DP protection and $k = 1$ suppression for varying fractions of the known target trace. 113
- 6.15 **Study of the impact of paired sampling on both attack settings ZK and KK.** Mean AUC scores with standard error for ZK and KK on size 1000 aggregates from the Milan dataset across different privacy budgets ε under user-day level ε -DP, for MIAs using independent sampling vs. paired sampling. . . . 114

6.16	Mean AUC scores with standard error for ZK and various aggregates sizes. Aggregates from the Milan dataset are computed with event level ε -DP for varying privacy budgets ε and for varying sizes of aggregates.	117
7.1	Illustration of correlation constraints $(\rho(X_1, Y), \rho(X_2, Y))$ that lead to only one possible bin for the unknown correlation $\rho(X_1, X_2)$. Here, we show results for $N_B = 3$ possible bins.	135
7.2	Impact of the number of samples K on the model-less attack performance. We report the accuracy (with 95% confidence interval) over 1000 target models.	137
7.3	Accuracy of the correlation inference attack on $n = 3$ variables as a function of the correlation constraints $\rho(X_1, Y)$ and $\rho(X_2, Y)$. We show results of (A) the model-less attack and of the model-based against two different target models: (B) logistic regression and (C) multilayer perceptron. The color of each cell indicates the accuracy of our correlation inference attack applied to models trained on synthetic datasets whose correlations $\rho(X_1, Y)$ and $\rho(X_2, Y)$ belong to the region defined by the cell. There are 200×200 cells in total. . . .	143
7.4	Logistic regression: Comparison between black-box and white-box correlation inference attacks. We report results for our model-less attack (A), our black-box model-based attack using as features the confidence scores (B. 1), and our white-box model-based attacks using as features the model weights (B. 2) or the combined weights and confidence scores (B. 3).	144
7.5	Multilayer perceptron (MLP): Comparison between black-box and white-box correlation inference attacks. We report results for our model-less attack (A), our black-box model-based attack using as features the confidence scores (C. 1), and our white-box model-less attacks using as features the raw model weights (C. 2), the canonical weights (i.e., the raw weights mapped to a canonical representation [GWY ⁺ 18], C. 3), or the combined canonical weights and confidence scores (C. 4).	144

- 7.6 Impact of the largest constraint $\max(|\rho(X_1, Y)|, |\rho(X_2, Y)|)$ on the attack accuracy for different number of variables n .** We compare the results of our model-based and model-less attacks on logistic regression (left) and MLP models (right). 146
- 7.7 Attack accuracy under scenario S2 and $N_B = 5$ correlation bins for different number of variables in the dataset n .** We report the accuracy (with 95% confidence interval) over 1000 target models. 146
- 7.8 Attack accuracy for different scenarios and number of variables in the dataset n .** We consider three different attack scenarios: **(S1)** the adversary knows the correlations between the target variables and the output variable $\rho(X_1, Y)$ and $\rho(X_2, Y)$, **(S2)** the adversary knows the correlations between all the input variables and the output variable $\rho(X_i, Y)_{i=1, \dots, n-1}$, and **(S3)** the adversary knows all the correlations between the variables except for the target correlation $\rho(X_1, X_2)$. We compute the accuracy (with 95% confidence interval) over 1000 target models. 148
- 7.9 Impact of mitigations on the accuracy of our attack against Logistic regression models.** We report results for two mitigations: limiting the number of black-box queries (A) and lowering the precision of confidence scores (B). We report the accuracy (with 95% confidence interval) of our model-less and model-based attacks over 1000 targets models for $n \in \{3, 4, 5\}$ variables. 149
- 7.10 Trade-off between the vulnerability of DP Logistic regression models to our attack (A) and their utility (B).** We compute the metrics (with 95% confidence interval) over 1000 target models for $n \in \{3, 4, 5\}$ variables. 150
- 7.11 Correlation constraint extraction from LR models for different number of queries $\tilde{Q}$.** We compute the $\mathbb{L}_1$ error between the inferred and the ground-truth values of the correlations: $\frac{1}{n} \sum_{i=1}^n |\hat{\rho}(X_i, Y) - \rho(X_i, Y)|$ (mean with 95% confidence interval). 160

7.12	Comparison between black-box and white-box CIAs against MLPs when the adversary uses different seeds (left) or the same seed (right) as the target model for training the models used to generate the meta-classifier’s training dataset. We report the mean attack accuracy (with 95% confidence interval).	161
B 1	Number of visits for both CDR and Milan datasets. The number of visits is computed per target over the 50 targets.	App. 4
D 1	Impact of the average constraint $(\rho(X_1, Y) + \rho(X_2, Y))/2$ on the attack accuracy for different number of variables n. We compare the results of our model-based and model-less attacks on logistic regression (left) and MLP models (right).	App. 8
D 2	Impact of the smallest constraint $\min(\rho(X_1, Y) , \rho(X_2, Y))$ on the attack accuracy for different number of variables n. We compare the results of our model-based and model-less attacks on logistic regression (left) and MLP models (right).	App. 9
D 3	Impact of mitigations on the accuracy of our attack against MLP models. We report results for two mitigations: limiting the number of black-box queries (A) and lowering the precision of confidence scores (B). We report the accuracy (with 95% confidence interval) of our model-less and model-based attacks over 1000 targets models for $n \in \{3, 4, 5\}$ variables.	App. 9
D 4	Fifa19: Impact of the granularity of one-way marginals available to the adversary. We report the attack accuracy (mean and standard deviation) on logistic regression models as we vary the number of discretization bins applied to the one-way marginals.	App. 10

Chapter 1

Introduction

1.1 Motivation

Data usage is essential for various mechanisms, such as machine learning systems, statistical modeling and decision-making mechanisms. Data collection has expanded continuously during the past years. Datasets are collected, traded, bought and are now a whole segment of the market. However, those datasets may directly –with information such as location, medical records and financial transaction– or indirectly –with information such as bias or global information about an area–, relate to private personal information. As the mechanisms trained using those datasets can leak this information, there is an increasing concern about safe-guarding it, while continuing to train accurate, usable systems.

Alongside the broader collection of datasets, a growing number of privacy regulations have been introduced by legal authorities worldwide [Gre21]. The name of the prominent privacy regulation in Europe is the General Data Protection Regulation (GDPR) [Eur16]. This regulation considers data to be anonymous if it cannot be linked back to an individual by any means ‘reasonably likely to be used’. This emphasizes the adversary-based definition of privacy: to ensure a dataset is private, one must take an adversarial point of view and answer the following question: Is the data leakage from the data release mechanism realistic or not?

Researchers in the field of data privacy, through the scope of adversarial evaluation, have proposed various methods to quantify the privacy leakage of data release mechanisms. Those adversarial methods consist of attacking the privacy of a given mechanism, through the lens of an adversary, having access solely to a predefined amount of knowledge. This evaluation uses a variety of techniques that often rely on very strong theoretical assumptions. This often results in privacy evaluations being dismissed by data practitioners, as the adversary is seen as too unrealistic of a threat. To address this issue, this thesis focuses on two main tasks. First, to provide a consistent identification of the privacy risks of data release mechanisms, through a better evaluation of the privacy leakage in data release mechanisms and identification of at-risk individuals in synthetic datasets. Second, to weaken the assumptions of attacks to make privacy adversaries more realistic. This task is divided into three different sections to cover different data release mechanisms, providing three different types of access to the data released: (a) synthetic data publishing (b) aggregation mechanisms and (c) machine learning models.

Shadow modeling [AMS⁺15, SSSS17] is a technique often used to construct privacy attacks against machine learning models. This technique first creates shadow models that mimic the behavior of the target model, and second trains a *meta*-model to extract information about the target record’s influence on the model. This approach is very costly in practice due to the high number of trained models, thus it is important to identify records that would be more at risk beforehand instead of running the attack on every record within the dataset. That is why there has been a growing research interest in vulnerable record identification in the machine learning community [CJZ⁺22, MS20]. But despite the interest for privacy evaluation, previous work has relied on various mechanism to identify at-risk records in the context of synthetic datasets, thus motivating the need for a consistent vulnerable record identification strategy. Indeed, [SOT22] based their identification strategy on records with rare values, and [HJC⁺22] based their identification strategy on low log-likelihood ratios, assuming independence between the different attributes. However in high dimensions, there can be a large group of records in minority groups and attributes are highly unlikely to be independent.

Synthetic data generators are a notable solution [BDR19] to limit privacy risks and share dataset information. A synthetic data generator is trained on a private and sensitive dataset,

and produces synthetic datasets that preserve the statistical properties of the real training dataset while producing realistic synthetic records. They are originally thought safe, because the original individual information in the training dataset was believed to be uninferable from the released information [NRD16, XSCIV19a, ZCP⁺17]. In that regard, a synthetic dataset in theory can then be used freely because it does not fall under the scope of the GDPR anymore. This particular interest emphasizes the need for good and consistent privacy evaluations to validate the safety of the synthetic data, leading some researchers to show that this promising solution was in fact, not without privacy risks [SOT22, HJC⁺22]. Yet, the identification of at-risk records remains inconsistent between papers, highlighting the need for a consistent identification mechanism.

Privacy risks, or quantification of the risk of information leakage about individuals, are then derived from the performances of the privacy attacks. The privacy risks evaluation the release of ML models poses is done through traditional ML evaluation: Given a dataset disjoint from the available information to the attacker, we sample multiple instances called *test datasets*, and we evaluate the privacy risk by averaging the success of the privacy adversary over this test pool. We here call this technique dataset sampling. Dataset sampling is a natural extension of the ML evaluation pipeline to the quantification of the privacy risk. However, it has been shown in previous research that the success of privacy adversaries often relies on the memorization of one or multiple, specific records by the model [AZT24, CCN⁺22]. One specific dataset may have very different at-risk individuals due to the uniqueness of its distribution, such as *outliers*. Moreover, outliers have been shown to present higher privacy risks [HJC⁺22, MGCdM23]. Therefore, from one dataset to another, outliers may be very different, and so will be their privacy risk evaluation. Thus, we here argue that averaging the privacy risk over multiple test datasets does not reflect the real risk of the *actual* dataset used.

Finally, most privacy adversaries rely on an artificial and hard-to-meet assumption in practice [SZH⁺18]: the use of an auxiliary dataset, with the same underlying distribution as the target dataset used to train the data release mechanism. While it is possible to find datasets of images freely available online which ultimately might be reassembled to follow the same underlying distribution as the target dataset, it is not the same for medical information, location

information or financial records. Due to the specificity of this data, it is hard to find a publicly available dataset following the same underlying distribution as one given target dataset. Finding such a dataset has for a long time not been considered *reasonably likely* to be found in practice, and left privacy attacks to remain a theoretical threat.

Each data release mechanism possesses its own specificity that leaks to some extent some information about the original training dataset. Indeed, each data release mechanism has been trained using the original training dataset, and thus contains some information stored in the mechanism about the dataset. Because of how this information is released, this allows for different types of attacks. We here consider three different types of data release mechanisms, for three different kinds of data release, namely direct access to individual synthetic records, access to aggregated real records, and indirect access through the release of a ML model. First, we consider a *synthetic generator*. It will produce synthetic datasets that will map the same dataset level information contained in their training dataset. This first data release mechanism is made of synthetic individual records. Second, we consider a data release mechanism that gives access to an aggregate of the individual information. Third, we now consider there is no direct release of any kind of the individual information, and that we only give a black-box access to a machine learning model. A machine learning model will contain in its weights, biases or outputs some information about the distribution of its training dataset, but this information is not directly accessible. Each data release mechanism, directly or indirectly, contains private information from its training dataset, motivating the need to lift the unrealistic assumption of a privacy adversary to realistically evaluate their privacy.

This thesis studies the *realisticness* of privacy adversaries by providing a better vulnerable record identification method, creating a specific evaluation setup to better understand the privacy risk of a specific dataset, and by weakening hard-to-meet assumptions at the core of privacy attacks.

1.2 Contributions

In this thesis, we make the following three contributions. Taken together, these contributions show that even while removing core assumptions of privacy adversaries, we were able to build convincing attacks by using synthetic models to generate the necessary information. In Figure 1.1, we illustrate how the various chapters of this thesis are integrated into our three contributions, highlighted in red, for the generic pipeline of adversarial privacy evaluation.

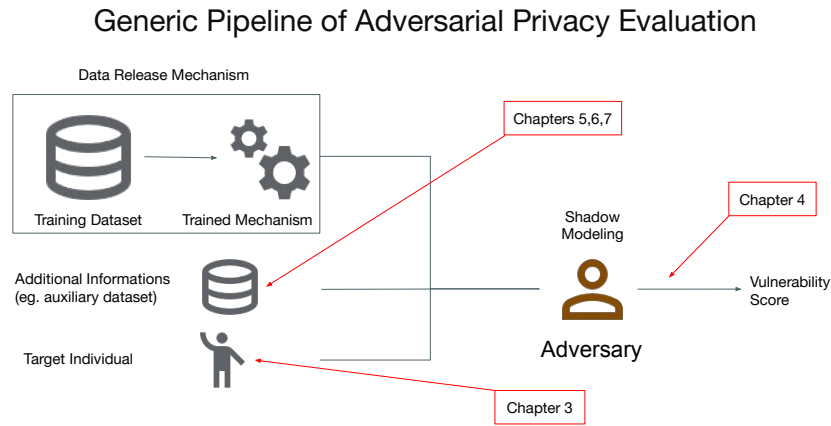


Figure 1.1: **Focus of each chapter in the generic privacy attack pipeline.** In this figure, we chose to present an adversary using the shadow modeling technique, as it is typically done in the literature [AMS⁺15, SSSS17].

In our first contribution in Chapter 3, we propose the first vulnerable record identification strategy in the context of synthetic datasets. In comparison with machine learning, where this is a well-known and studied problem [CJZ⁺22, MS20], in the field of generative models, there is no commonly agreed-upon vulnerable record identification mechanism. E.g. in their state-of-the-art approach, Houssiau et al. [HJC⁺22] use lowest log-likelihood-based record identification, while Stadler et al. [SOT22] hand-pick vulnerable records. Our results show that our strategy correctly evaluates the most vulnerable records, and proposes records with more vulnerability than the previous techniques used in the literature.

In our second contribution in Chapter 4, we propose a novel specific evaluation setup to evaluate the privacy risk of a target record. We identify a limitation in the current evaluation setup of Membership Inference Attacks (MIAs). Indeed, as a natural extension of the evaluation of

machine learning models, it is usually instantiated in practice by randomly sampling multiple test datasets from a larger test pool, non overlapping with the auxiliary dataset given to the adversary. Adding the target record into half of the test datasets results in an averaged evaluation of the privacy risk presented by the target record over the possible distributions. Recent work has shown how the privacy risk of a record is distribution dependent [CJZ⁺22, MGCdM23], suggesting that an outlier presents more risk than other records. We first evaluate the impact of randomness over the current evaluation, and we then introduce a new specific evaluation setup to better grasp the privacy risk of the target record with respect solely to the dataset released through the data release mechanism. Our results show how these results hold for both synthetic generators and machine learning models. We then instantiate a strong adversary with the knowledge of the specific dataset used to train the data release mechanism, to show that attacks could be adapted to a specific dataset.

MIAs rely on the access of an auxiliary dataset with the same underlying distribution of the target dataset (the need for additional information, as shown in Figure 1.1). In practice, access to such a dataset can be very complex, and often seen as an hard-to-meet hypothesis. Our third contribution answers this need for auxiliary information by removing access to additional datasets, and using generative models to provide the adversary with the necessary information. We divided the contribution over three different chapters, focusing in each chapter on a different type of data release mechanism. Each data release mechanism studied present a different access to the data, to cover as much as possible the spectrum of potential releases.

In the first part of our third contribution in chapter 5, we show how we can remove this hypothesis by using generative models. In this chapter, we first focus our analysis on synthetic data generators as data release mechanisms. We consider three different levels of access to the model for the adversary. First, the adversary can query the generator an infinite amount of times, second, the adversary only uses the released synthetic dataset, and third, the adversary has access to a secondary synthetic generator, explicitly trained without the target record. Our results show that an adversary can still be a serious threat to the privacy of target records even if we remove the hard-to-meet assumption.

In our second part of our third contribution in chapter 6, we show how we can remove the access to auxiliary information assumption against aggregation techniques. Aggregation techniques are well known to be at risk [DN03, PTDC17], however, adversaries mostly rely on access to past participation in aggregated groups [PTDC17]. We here provide an attack achieving similar performance, but without any access to auxiliary information. This attack is thus made very realistic, even when the aggregate is made differentially private before being released. Our results show that unless the aggregate is very noisy, MIA performance is always high, even if the attack only relies on the released aggregate.

Property Inference Attacks are a type of privacy attack that targets global properties of a dataset that often do not require the adversary to possess any additional information about the target dataset. Dataset-level characteristics could indeed reveal sensitive information about the private dataset [AMS⁺15, GWY⁺18, ZTO21], such as bias. Another dataset-level characteristic that can be learned through access to data release mechanisms is the distribution. In our third part of our third contribution in chapter 7, we show how we can use generative models to infer correlations of the training dataset of machine learning models, through our Correlation Inference Attack (CIA). We also show how correlation can be used as a building block to other record level privacy attacks. In this chapter, we use correlations to develop a better Attribute Inference Attack (AIA).

1.3 Thesis outline

The remainder of the thesis is organized as follows:

Chapter 2 Formalizes the key concepts used throughout the thesis.

Chapter 3 Presents an efficient identification of vulnerable individuals in the training data of synthetic data generators.

Chapter 4 Presents a specific evaluation of privacy risk, to provide a more reliable estimate of the dataset's privacy.

Chapter 5 Presents how a generative model can be used to remove the auxiliary information access in the context of membership inference attacks against synthetic datasets.

Chapter 6 Presents how a generative model can be used to remove the auxiliary information access in the context of membership inference attacks against aggregation mechanisms.

Chapter 7 Presents a new type of privacy attack, a correlation inference attack, to infer correlations between variables of a tabular dataset and how this can be used as a building block for attribute inference attacks, removing the auxiliary information access hypothesis.

Chapter 8 Concludes this thesis by summarizing our contribution and presenting ideas for future work related to this thesis.

1.4 Publications

The research outlined in this thesis comes from the following peer-reviewed articles with an asterisk (*) indicating the primary author(s) of each paper:

- Matthieu Meeus*, Florent Guépin*, Ana-Maria Crețu and Yves-Alexandre de Montjoye (2023, September). “Achilles’ heels: vulnerable record identification in synthetic data publishing”. In European Symposium on Research in Computer Security (pp. 380-399). Cham: Springer Nature Switzerland. (Chapter 3)
- Florent Guépin*, Matthieu Meeus*, Ana-Maria Crețu and Yves-Alexandre de Montjoye (2023, September). “Synthetic is all you need: removing the auxiliary data assumption for membership inference attacks against synthetic data”. In European Symposium on Research in Computer Security (pp. 182-198). Cham: Springer Nature Switzerland. (Chapter 5)

- Vince Guan*, Florent Guépin*, Ana-Maria Crețu and Yves-Alexandre de Montjoye (2024, July). “A Zero Auxiliary Knowledge Membership Inference Attack on Aggregate Location Data”. Proceedings on Privacy Enhancing Technologies. (Chapter 6)
- Ana-Maria Crețu*, Florent Guépin* and Yves-Alexandre de Montjoye (2024, July). “Correlation inference attacks against machine learning models”. In Science Advances, 10(28), eadj9260. (Chapter 7)

The following article was included but is still under review:

- Florent Guépin*, Nataša Krčo*, Matthieu Meeus and Yves-Alexandre de Montjoye (2024). “Lost in the Averages: A New Specific Setup to Evaluate Membership Inference Attacks Against Machine Learning Models.” arXiv preprint arXiv:2405.15423. (Chapter 4)

Chapter 2

Background theory

This chapter aims to provide the key definitions necessary to understand the theoretical notions used throughout the paper. Concepts specific to each chapter are left to be defined within those chapters.

2.1 Synthetic distributions

In this chapter, we aim to formally provide a definition of a synthetic distribution. To do it, we will first present the different dataset-level definitions used in this thesis, as well as the definition of a synthetic dataset and synthetic generator, the model used to generate synthetic datasets based on a learned distribution. Second, we present all the generators used in the thesis. We conclude this chapter by presenting the definition of an outlier with respect to a given distribution.

2.1.1 Synthetic datasets

We here first present the formal definition of both tabular and image datasets, that will then be used to train synthetic generators to synthesized datasets, similarly to [HJC⁺22].

Definition 2.1. Considering $\mathcal{F} = \mathcal{F}_1 \times \dots \times \mathcal{F}_n$, where each $\mathcal{F}_i$ for $i \in [1, n]$ defines a probability distribution over a finite space, a **tabular record** x is a set of n attributes such that $x = (x_1, \dots, x_n) \sim \mathcal{F}$.

Definition 2.2. A **tabular dataset** D is a finite collection of tabular records, formally $\exists m$ such that $D = \{x_t \mid \forall t \in [1, m] \ x_t \sim \mathcal{F}\}$.

Definition 2.3. An **image record** X is defined as a 3 dimensional matrix, $(X_{i,j}^r, X_{i,j}^g, X_{i,j}^b) \in [0, 255]^3$ and an **image dataset** D is a finite collection of image records.

When an entity (e.g., a company, a governmental institution or a research institute) wants to give third-party access to a sensitive, proprietary dataset D , one popular approach consists in training a *synthetic data generator* to produce a *synthetic dataset*. This *synthetic dataset* contains all the relevant statistical information about D , while trying to mitigate the privacy risks when released.

Definition 2.4. A **synthetic tabular generator** Φ is a randomized function ($\Phi : \mathcal{F} \rightarrow \mathcal{F}$) trained on a tabular dataset D , to produce its synthetic counterpart, D^s , drawn from the learned distribution of Φ . We write $\Phi = \mathcal{G}(D)$ to say that a certain fitting procedure $\mathcal{G}$ (e.g., parameter fitting of a Bayesian network [ZCP⁺17] or a Generative Adversarial Network (GAN) such as CTGAN [XSCIV19b].) was applied to the original dataset D to obtain the generator Φ .

Following this definition, we now denote by D^s the synthetic dataset obtain from training a synthetic generator Φ using the target tabular dataset D .

2.1.2 Generative models

We present here the two main statistical models that will be used in this thesis.

Synthpop is a model first introduced as an R package by Nowok et al. [NRD16]. The goal is to provide an estimation of the joint distribution of the training dataset ($\mathbb{P}[D_1, \dots, D_n]$ for each column D_i of the training dataset D) by looking at the series of conditional probabilities

of the training distributions ($\mathbb{P}[D_i|D_j]$ for each column D_i, D_j of the training dataset D). These conditional probabilities are computed through classification and regression trees fitted on this training dataset. For each record, it then selects a column and generates its value according to the learned distribution, and the next ones are estimated conditional on the generated ones.

BayNet uses Bayesian networks to generate synthetic datasets [ZCP⁺17]. To do so, the BayNet algorithm first creates a Directed Acyclic Graph from the training dataset, to capture the causal relationships between the features of the training dataset. To create the graph, the mechanism estimates $\mathbb{P}[X|\text{Parent}(X)]$ (the causal relationship) from the training dataset. Similarly to Synthpop, the synthetic dataset is then created by sampling from the learned joint distribution of the training dataset, which is computed over the series of different conditional distributions.

2.1.3 Outliers

Definition 2.5. For a given distance metric d over the space $\mathcal{D}$ ($\mathcal{D}$ is the space made of all x such that $x \sim \mathcal{F}$), an **outlier** is a data point x within a given dataset D , with respect to a certain threshold t if $\forall y \in D \setminus \{x\}, d(x, y) \geq t$.

Intuitively, outliers are records that are distant from the majority of the records in a dataset D . We provide a one dimensional example. Consider a gaussian distribution $\mathcal{G}$ of mean μ and standard deviation σ , we can compute that 99.73% of the data points sampled from a gaussian distribution falls within the interval $[\mu - 3\sigma, \mu + 3\sigma]$ [RCKW05]. If we define a dataset D , such that points inside D follow the gaussian distribution $\mathcal{G}$, data points outside of $[\mu - 3\sigma, \mu + 3\sigma]$, for the standard distance metric $\mathcal{L}_1$ over $\mathbb{R}$, are considered *outliers*. An outlier is more likely to exist in multi-dimensional dataset, as the space is sparser.

Within a multi-dimensional dataset, outliers often present some unique feature association, which makes it very hard to be learned by the synthetic generators. This uniqueness ultimately often makes synthetic generators either not able to provide accurate information about outliers

which produce synthetic datasets lacking representativity and not usable, or remember too much about them, which makes them highly vulnerable. In either case, the private information of outliers within a dataset D , due to their vulnerability, raises a particular attention when releasing the dataset.

2.2 Privacy attacks

In this section, we will first introduce the shadow modeling technique for privacy attacks, second, we will present the membership inference attack, then we will conclude with a presentation of the attribute inference attack.

2.2.1 Shadow modeling

Shadow modeling is a technique often used at the basis of state-of-the-art privacy attacks [SSSS17, SOT22, HJC⁺22, CCN⁺22, GMCdM23, AMS⁺15]. At a high level, the adversary trains shadow mechanisms on an auxiliary dataset to mimic the behavior of the target mechanism $\mathcal{M}_T$. Concretely, to be able to train meaningful shadow mechanisms, the adversary usually uses an auxiliary dataset with the same underlying distribution as the training dataset. From this auxiliary dataset, the adversary samples shadow training datasets, half of them containing the target record, the other half will not contain it. Labeling all the trained shadow mechanisms, we then use it to train a meta-classifier. At inference time, the adversary queries the meta-model on the target mechanism, and returns the prediction.

2.2.2 Membership inference attack

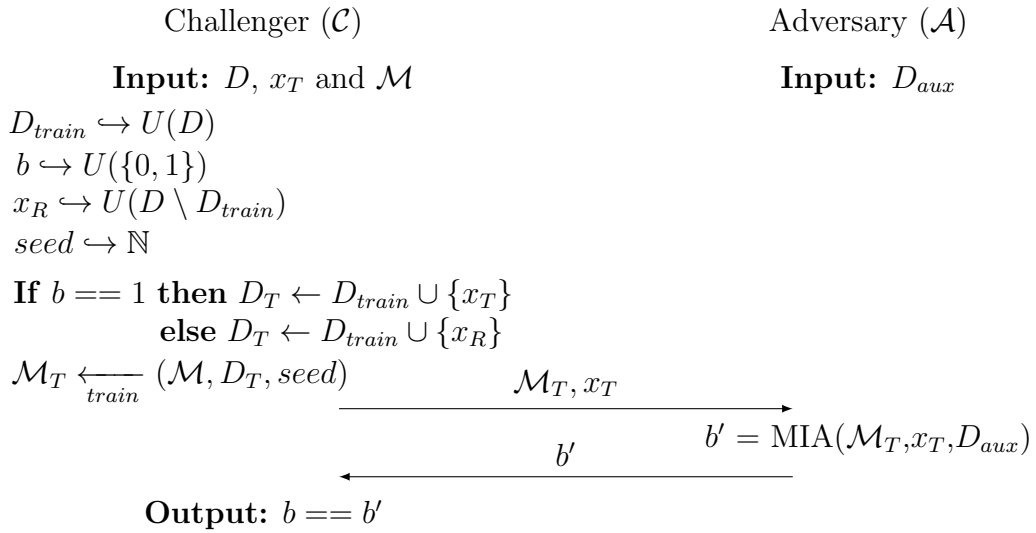
Threat Model. Consider a *target record* x_T and a data release mechanism $\mathcal{M}$ trained on a dataset D_{train} . In the following privacy game, the data owner who releases the data through $\mathcal{M}$, is called the *Challenger*. An adversary trying to attack $\mathcal{M}$ is called *Adversary*. To deploy the attack following the common usage, we assume that the adversary has access to an auxiliary

dataset D_{aux} with the same underlying distribution as D_{train} . The goal of the adversary is to infer whether the target record x_T has been used during the training of the data release mechanism $\mathcal{M}$.

For example, we can think of a hospital trying to release a model trained on patients from the oncology department. In that case, performing a membership inference attack consists of learning if a particular record has been seen during training, disclosing its presence within the oncology department.

Membership Inference Attack Privacy Game. Membership Inference Attacks (MIAs), first introduced by [AMS⁺15] and later formally defined by [SSSS17], are instantiated as the following privacy game.

MIA Privacy Game



If the adversary correctly guesses b , then we call the privacy game a *success*. Evaluation of the MIA using the previous privacy game is done by running it multiple times and reporting the proportion of success.

2.2.3 Attribute inference attack

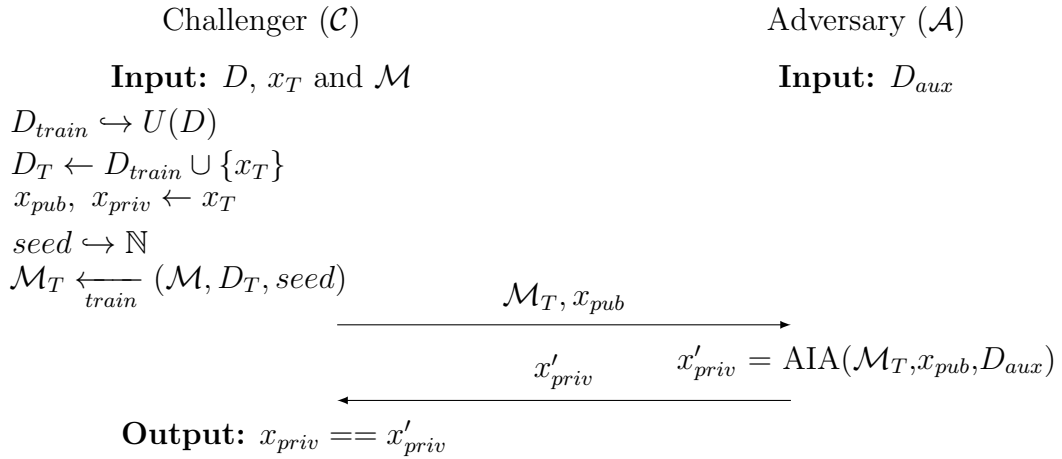
Threat Model. Similarly to 2.2.2, we here consider a *target record* x_T and a data release mechanism $\mathcal{M}$ trained on a dataset D_{train} . The data owner, releasing the data through $\mathcal{M}$, is

called the *Challenger*. The adversary trying to attack $\mathcal{M}$ is called *Adversary*. To deploy the attack, we give access to the adversary to an auxiliary dataset D_{aux} with the same underlying distribution as D_{train} . Unlike 2.2.2, we also give access to the public information of the target x_T to the Adversary, retaining the private ones. Given that public information, the Adversary is trying to infer the private value of the private features from target x_T .

For example, we can think of a hospital trying to release a model to predict the illness of a patient based on his public record. In that case, performing an attribute inference attack consists of trying to infer the illness one patient suffers from based on the released model and the public information of this record. Note that the adversary knows the target record to be in the training dataset of the target model.

Attribute Inference Attack Privacy Game. Attribute Inference Attacks (AIAs), first defined in [FLJ⁺14] are instantiated as the following privacy game.

AIA Privacy Game



If the adversary correctly guesses the private features of the record x_T , then we call the privacy game a *success*. Evaluation of the AIA using the previous privacy game is done by running it multiple times and reporting the proportion of success.

2.3 Differential privacy

2.3.1 Definition

Differential Privacy (DP), first introduced in [Dwo06] is a formal privacy guarantee. This formal guarantee is applicable for various types of data release mechanisms, including machine learning models, synthetic data generators and aggregation mechanisms. It allows the data practitioner to provide theoretical guarantees on the released model by making some modifications to the original mechanism, often by adding a noise addition step during training.

DP relies on the notion of neighboring. Indeed, following the work of [Dwo06] it is possible to define two *neighboring* sets S, S' . Two sets S and S' are two *neighboring* sets if and only if S and S' differ by only one *privacy unit*. In the case where the privacy unit is defined as a “record”, then we can define two *neighboring* datasets D, D' if they differ by only one record.

Formally, the definition of differential privacy, following the definition provided by [Dwo06] is as follows.

Definition 2.6 (ϵ -Differential Privacy). *A data release mechanism $\mathcal{M}$ is said to be ϵ -Differentially Private (ϵ -DP) if for any pair of neighboring datasets S, S' that differ by one privacy unit, and for all subsets $T \in \text{Im}(\mathcal{M})$, the following inequality holds: $\mathbb{P}[\mathcal{M}(S) \in T] \leq e^\epsilon \mathbb{P}[\mathcal{M}(S') \in T]$.*

When we present an ϵ -DP mechanism, ϵ is referred to as the *privacy budget*. The privacy budget controls the level of protection provided by the ϵ -DP mechanism. The smaller ϵ is, the higher the level of protection given by the mechanism is. In practice, there is a tradeoff between the utility of the data release mechanism, and the level of privacy provided by ϵ .

2.3.2 Differentially private generator

PrivBayes is a differentially private version of the BayNet algorithm described above, first proposed in [ZCP⁺17]. They propose to compute their differentially private version of BayNet

as follows. First, they trained a Bayesian Network using an ϵ_1 -DP algorithm, second, they computed the conditional probabilities using a second ϵ_2 -DP technique. By use of the composition theorem ([DKM⁺06]), they end up with an ϵ -DP ($\epsilon = \epsilon_1 + \epsilon_2$) algorithm. Note that when $\epsilon \rightarrow \infty$, this becomes equivalent to the standard BayNet procedure described in 2.1.2.

Chapter 3

Vulnerable record identification in synthetic data publishing

The work presented in this chapter has been published as: Meeus, M., Guépin, F.*, Crețu, A. M., & de Montjoye, Y. A. (2023, September). Achilles' heels: vulnerable record identification in synthetic data publishing. In *European Symposium on Research in Computer Security* (pp. 380-399). Cham: Springer Nature Switzerland. https://link.springer.com/chapter/10.1007/978-3-031-51476-0_19*

Statement of contributions: Matthieu Meeus and I are joint first authors on two papers, presented in Chapter 3 and 5. We collaborated in the design of the methods, experiments and in the writing of the code and the paper for both. We equally contributed to the production of the figures and writing of the paper for both. Matthieu led the writing of the code and experiments for this paper, while I led the writing of the code and experiments for the paper presented in Chapter 5.

3.1 Introduction

There is increased demand from businesses, governments, and researchers to make data widely available to support research and innovation [Fin22], including the development of Artificial Intelligence (AI) models. Data, however, most often relates directly or indirectly to individuals, raising privacy concerns.

Synthetic data is seen as a promising solution to share individual-level data while preserving privacy [BDR19]. Synthetic data is generated by sampling new data values from a statistical model whose parameters are computed from the original, private data. A large range of techniques have been proposed to generate synthetic data [ZCP⁺17, NRD16, XSCIV19a]. Synthetic data, if truly privacy-preserving, can be shared and used freely as it no longer falls under the scope of data protection legislation such as the European Union’s General Data Protection Regulation (EU GDPR) or California’s CCPA. Consequently, synthetic data has garnered significant interest from statistical offices [BSDM19], health care agencies [NHS20] and the financial sector [Fin22].

Ensuring that synthetic data preserves privacy is, however, difficult. On the one hand, while the statistical model aggregates the original data, it is well known that aggregation alone is not sufficient to preserve privacy [DN03, PTDC17]. On the other hand, ensuring that synthetic data generation models achieve formal privacy guarantees is difficult and comes at a cost in utility [AGR23, SOT22].

Membership Inference Attacks (MIA) are a key tool to evaluate the privacy guarantees offered by synthetic data generation models, differentially private or not, in practice [HJC⁺22]. If successful, an MIA is able to infer that a particular record was part of the original data used to train the synthetic data generation model. Increasingly advanced MIAs have been proposed against synthetic data [SOT22, HJC⁺22]. They are run on a per-record basis, instantiating an adversary that aims to distinguish between two worlds: (1) a world in which the synthetic dataset released was generated from a model fitted on data *including* the target record and (2) the alternative world in which the model was fitted on data *excluding* the target record.

MIAs on synthetic data have already successfully disproved the believe that aggregation alone is sufficient to preserve privacy as well as detected issues with differentially private synthetic data generation models [SOT22].

Fully evaluating the privacy risk of a dataset using MIAs is however out-of-reach. Indeed state-of-the-art MIAs rely on shadow models which require training a large number, often in the order of thousands, of synthetic data generators to learn the optimal decision boundary for each record. Running a state-of-the-art MIA on a record takes in our experiments, depending on the generator and the dataset, approximately 1.5 to 6 hours on dedicated computation facilities. Even for the (relatively) small datasets of 1,000 records we consider, fully estimating the privacy risk of a dataset, would take up to 250 days of compute.

Current evaluation of the privacy risk posed by synthetic data, and of their privacy-utility trade-off, is thus currently the result of ever more advanced MIAs evaluated on only a handful of records selected using different methods. Specifically, Stadler et al. [SOT22] evaluate their attack on ten records, half random and half hand-picked outliers selected based on a rare attribute value, while Houssiau et al. [HJC⁺22] evaluate their attack on one record selected based on the lowest log-likelihood.

Such approaches, by not providing a uniform record selection, could strongly underestimate the privacy risks of a synthetic data release, missing highly vulnerable records or wrongly concluding that most records are not at risk.

Contribution. We here propose what is, to the best of our knowledge, the first vulnerable record identification strategy for synthetic data.

In this chapter, we formalize the problem and propose a principled and simple metric to identify vulnerable records in the original, private dataset: the mean distance to its closest neighbors. While simple, we show our method to outperform previous approaches by 7.2 percentage points (p.p.) on average on two datasets and two generators. Both when comparing the performance of ever improving attack methodologies as well as to estimate the potential privacy risk in real world synthetic data publishing, this is a significant difference.

We then extensively evaluate our method. First, we evaluate its applicability across different MIA approaches. We develop a fundamentally different MIA approach and show the risk estimated using our vulnerable record identification method to be consistently and significantly higher compared to prior methods. More specifically, in contrast with Houssiau et al. [HJC⁺22], who train a meta-classifier to infer the membership of a record based on hand-engineered aggregate statistics computed on the synthetic dataset, we develop an attack that trains a meta-classifier directly on the synthetic dataset viewed as *a set of records*. After performing the same extensive empirical evaluation, we find that the performance of our attack increases also significantly when computed on records identified by our method, by 5.2 p.p. on average across datasets and generators. Next, we evaluate the sensitivity of our results on both the number of neighbors included in our metric as well as the distance metric. In both cases, we find that the results do not change significantly and confirm that our chosen metric is justified. Finally, we evaluate our metric on a differentially private synthetic data generator for varying values of the privacy budget ϵ . We confirm that MIAs fail when the privacy budget is decreased to $\epsilon = 1$, and find that our metric consistently identifies more vulnerable records for larger values of ϵ .

With this principled method, we hope to provide a consistent record selection and provide an accurate estimation of the privacy risk in synthetic data publishing. A formal vulnerable record identification method would enable (1) researchers to fairly compare future state-of-the-art attacks and (2) practitioners to, in line with EU legislations, evaluate the privacy protection of synthetic data on the worst-case scenario.

We make our source code allowing to reproduce the results of our experiments available at: <https://github.com/computationalprivacy/MIA-synthetic>.

3.2 Related work

Membership Inference Attacks (MIAs) were first developed by Homer et al. [HSR⁺08] to study whether the contribution of an individual to a Genome-Wide Associated Study can

be inferred from released genomic aggregates. Their attack was based on a statistical test aiming to distinguish between aggregates that include the individual and aggregates computed on individuals randomly drawn from a reference population. Sankararaman et al. [SOJH09] extended the analysis soon after and showed the risk of membership inference to increase with the number of aggregates released m , but decrease with the number of individuals n in the dataset. MIAs have since been widely used to evaluate the privacy risk in aggregate data releases such as location [PTDC17] or survey data [BB20].

MIAs against Machine Learning (ML) models. Shokri et al. [SSSS17] proposed the first MIA against ML models. Their attack relies on a black-box access to the model and the shadow modeling technique. MIAs against ML models have since been extensively studied in subsequent works [YGFJ18, SZH⁺18, TLG⁺19, NSH19, LF20], both from a privacy perspective but also to better understand what a model learns. These works have, for instance, shown the risk to be higher in overfitted models [YGFJ18] and smaller datasets [SSSS17], and to be mitigated by differentially private training, albeit at a cost in utility [JE19, LF20].

Disparate vulnerability of records. Previous work on MIA against ML models has shown that not all records are equally vulnerable to MIAs. The measured risk has e.g. been shown to vary with the label [SSSS17] and to be higher for outlier records [LWB⁺20, CJZ⁺22] and members of subpopulations [CS21]. Feldman proposed a theoretical model demonstrating that models may, in fact, need to memorise rare or atypical examples in order to perform optimally when trained on long-tailed distributions such as those found in modern datasets [Fel20, FZ20]. Carlini et al. [CJZ⁺22] showed this effect to be relative, as removing the most vulnerable records increases the vulnerability of remaining ones. Importantly for this work, Long et al. [LWB⁺20] argued that attacks should be considered a risk even when they only perform well on specific records. Long et al. proposed an approach to select vulnerable records prior to running the attack so as to increase its efficiency and precision. While their work [LWB⁺20] shows that records further away from their neighbors are more vulnerable, it only considers ML classification models.

MIAs against synthetic tabular data. MIAs have been extended to synthetic tabular data,

the focus of our work. From a methodological standpoint, they fall broadly into two classes. The first class of methods directly compares synthetic data records to the original, private records, searching for exact or near-matches [DFRSC15, LWY19, YDD⁺19, GBWT22]. The second class of methods instantiates the shadow modeling technique in the black-box setting, assuming the adversary has knowledge of the algorithm used to generate synthetic data and to an auxiliary dataset. Stadler et al. [SOT22] trained a meta-classifier on aggregate statistics computed on the synthetic shadow datasets, specifically the mean and standard deviation of the attributes, correlation matrices and histograms. Houssiau et al. [HJC⁺22] extended this work using k -way marginal statistics computed over subsets of the attribute values of the targeted record. They also extended the threat model from the standard black-box setting, which assumes knowledge of the generative model used, to the no-box setting that lacks this assumption. In this chapter, we focus exclusively on shadow modeling-based attacks which are the state of the art.

3.3 Identifying vulnerable records

We here propose and validate a simple, yet effective, approach to identify vulnerable records in synthetic data publishing. Our approach is motivated by findings of previous works in the ML literature that records further away from their neighbors present a higher risk compared to randomly selected records. To identify such records, we compute a distance metric $d(x_i, x_j)$ between every pair of records in the dataset $(x_i, x_j) \in D \times D$, with $i \neq j$ ¹. Then, for every record $x_i \in D$, we define its *vulnerability score* $V_k(x_i)$ as the average distance to its closest k neighbors in the dataset.

Definition 3.1 (Vulnerability Score). *Given a dataset D , a record $x_i \in D$, a number of neighbors k and a distance metric d , the vulnerability score of the record is defined as $V_k(x_i) = \frac{1}{k} \sum_{j=1}^k d(x_i, x_{i_j})$, where $x_{i_1}, \dots, x_{i_{|D|-1}}$ is the re-ordering of the other records according to their increasing distance to x_i , i.e., $d(x_i, x_{i_1}) \leq \dots \leq d(x_i, x_{i_{|D|-1}})$, with $i_j \neq i$ for every $j = 1, \dots, |D| - 1$.*

¹Note that since the dataset D can contain repeated records (i.e., two or more individuals sharing the same attributes), the closest record to x_i can be a duplicate $x_j = x_i, j \neq i$ such that the distance between them is zero: $d(x_i, x_j) = 0$.

Finally, we rank the records in D decreasingly by their vulnerability score $V_k(x_{r_1}) \geq \dots \geq V_k(x_{r_{|D|}})$ and return the top- R records as the most vulnerable records. In the unlikely event that the last record(s) included in the top- R and the first one(s) not included have the exact same value for the vulnerability score, then a random subset is chosen.

Our distance metric d carefully distinguishes between categorical and continuous attributes. Recall that the dataset consists of F attributes. When the space of values that can be taken by an attribute is discrete, e.g., the gender of an individual $\mathcal{V}_f = \{\text{female}, \text{male}\}$, we refer to the attribute as *categorical*, while when the space of values is continuous, e.g., the income of an individual, we refer to the attribute as *continuous*. Denote by $\mathcal{F}_{\text{cont}}$ and $\mathcal{F}_{\text{cat}}$ the subsets of continuous and categorical attributes, with $\mathcal{F}_{\text{cont}} \cup \mathcal{F}_{\text{cat}} = \{1, \dots, F\}$ and $\mathcal{F}_{\text{cont}} \cap \mathcal{F}_{\text{cat}} = \emptyset$. We preprocess each record $x_i = (x_{i,1}, \dots, x_{i,F}) \in D$ as follows:

1. We convert every categorical attribute $x_{i,f}$, with $f \in \mathcal{F}_{\text{cat}}$ and $|\mathcal{V}_f|$ possible values $v_1, \dots, v_{|\mathcal{V}_f|}$ to a one-hot encoded vector:

$$h(x_{i,f}) := (\mathbb{1}(x_{i,f} = v_1), \dots, \mathbb{1}(x_{i,f} = v_{|\mathcal{V}_f|})), \quad (3.1)$$

i.e., a binary vector having 1 in the q -th position if and only if the attribute value is equal to v_q . Here, $\mathbb{1}$ denotes the indicator function. We then concatenate the one-hot encoded categorical attributes into a single vector:

$$h(x_i) := (h(x_{i,f}))_{f \in \mathcal{F}_{\text{cat}}} \quad (3.2)$$

2. We normalise every continuous attribute $x_{i,f}$, with $f \in \mathcal{F}_{\text{cont}}$ to range between 0 and 1, using min-max scaling:

$$n(x_{i,f}) := \frac{x_{i,f} - m_f(D)}{M_f(D) - m_f(D)}, \quad (3.3)$$

that is, we scale the values using the minimum $m_f(D) := \min_{j=1, \dots, |D|} (x_{j,f})$ and maximum $M_f(D) := \max_{j=1, \dots, |D|} (x_{j,f})$ values of the attribute estimated from the dataset D . We then

concatenate the normalised continuous attributes into a single vector:

$$c(x_i) := (n(x_{i,f}))_{f \in \mathcal{F}_{\text{cont}}} \quad (3.4)$$

Using this notation, we define our distance metric between two records x_i and x_j as follows (we here recall that $F = |\mathcal{F}_{\text{cat}} \cup \mathcal{F}_{\text{cont}}|$):

$$d(x_i, x_j) := 1 - \frac{|\mathcal{F}_{\text{cat}}|}{F} \underbrace{\frac{h(x_i) \cdot h(x_j)}{\|h(x_i)\|_2 * \|h(x_j)\|_2}}_{\text{cosine similarity between categorical attributes}} - \frac{|\mathcal{F}_{\text{cont}}|}{F} \underbrace{\frac{c(x_i) \cdot c(x_j)}{\|c(x_i)\|_2 * \|c(x_j)\|_2}}_{\text{cosine similarity between continuous attributes}}, \quad (3.5)$$

where $\cdot$ denotes the dot product between two vectors, $*$ denotes scalar multiplication, and $\|a\|_2 := \sqrt{\sum_{l=1}^L a_l^2}$ is the Euclidean norm of a vector $a \in \mathbb{R}^L$.

The resulting distance is a generalisation of the cosine distance across attribute types. It ranges between 0 and 1, with larger values indicating records to be less similar. The distance is equal to 0 if and only if the records are identical.

3.4 Experimental setup

3.4.1 State-of-the-art MIA against synthetic tabular data

Houssiau et al. [HJC⁺22] developed an MIA that instantiates the shadow modeling technique (see Chapter 2.2 for details) by training a random forest classifier on a subset of k -way marginal statistics that select the targeted record. The authors evaluated this approach on only one outlier record and found it to outperform other methods. Throughout our experiments, we found the attack to consistently outperform alternative methods and thus confirm it to be the state-of-the-art attack against synthetic tabular data. We refer to it as the *query-based attack*.

Given a target record x_T and a synthetic dataset D^s sampled from a generator fitted on a dataset D , this attack computes *counting queries* Q^A to determine how many records in the synthetic

dataset match a subset $A \subset \{1, \dots, F\}$ of attribute values of target record x_T . We denote by a_i the i -th attribute for every $x \in D^s$ in the following: $Q^A(x_T) = \text{COUNT WHERE } \bigwedge_{i \in A} (a_i = x_{T,i})$. Counting queries are equal to k -way marginal statistics computed on the attribute values of the target record, multiplied by the dataset size $|D^s|$.

Our intuition behind this attack is twofold. First, when these statistics are preserved between the original and synthetic datasets, the answers are larger by 1, on average, when $x_T \in D$ compared to when $x_T \notin D$, since the queries select the target record. Second, the impact of the target record on the synthetic dataset is likely to be local, i.e., more records matching a subset of its attribute values are likely to be generated when $x_T \in D$, leading to statistically different answers to the same queries depending on the membership of x_T .

Like Houssiau et al. [HJC⁺22], we randomly sample a predetermined number N of attribute subsets A and feed the answers to the associated Q^A , computed on the synthetic shadow dataset, to a random forest meta-classifier. For computational efficiency reasons, we here use a C-based implementation of the query computation [CHCdm22] instead of the authors' Python-based implementation [HJC⁺22].

We further extend the method to also account for continuous columns by considering the less-than-or-equal-to $\leq$ operator on top of the equal operator $=$ that was previously considered exclusively. For each categorical attribute we then use $=$, while using the $\leq$ for each continuous attribute.

3.4.2 A new MIA against synthetic tabular data

MIAs against synthetic data is an active area of research with a competition [All23] now being organised to develop better, more accurate, MIAs. To evaluate the effectiveness of our vulnerable record identification, we developed a new attack which we aimed to be as different as possible to the existing state of the art. We call this new attack *target attention*.

Inspired by developments in natural language processing [VSP⁺17], the target attention model computes record-level embeddings. Through a modified attention mechanism, these embedding

are able to interact with the embedding of the target record, which in turn leads to a dataset-level embedding that is used as input for a binary classifier. The detailed steps are laid out below.

We start by preprocessing the synthetic dataset used as input D^s . First, we compute the one-hot-encoded values for all categorical attributes and apply this consistently to all the synthetic records and the target record. The continuous attributes remain untouched. Second, we compute all the unique records in the dataset and their multiplicity. Third, we compute the distance following equation (3.5) between all unique synthetic records and the target record. Finally, we rank all unique synthetic record by increasing distance to the target record and, optionally, only keep the *top* X closest records, where X is a parameter.

The final pre-processed synthetic dataset $D^{s,p} \in \mathbb{R}^{X \times F'}$ is a matrix containing the top X unique records, their multiplicity and the distance, ranked by the latter, where F' denotes the number of features after one-hot-encoding of the categorical attributes added by 2 to account for the multiplicity and the distance. For consistency, the target record x_t is processed in the same way, with one-hot-encoded categorical attributes, multiplicity 1 and distance 0, resulting in the pre-processed target record $x_{t,p} \in \mathbb{R}^{F'}$.

Figure 3.1 illustrates our target attention model, which we describe in Algorithm 1. The model takes as input the pre-processed synthetic dataset $D^{s,p}$. First, a multilayer perceptron (MLP) is used to compute embeddings of both the pre-processed records as well as the target record. Second, after computing a query vector for the target record and keys and values for the records, an attention score is computed for each record. After applying a *softmax* function across all attention scores, all record values are summed weighted by the respective attention score to eventually lead to the dataset-level embedding which is used as input for a final MLP as binary classifier. The output of the entire network is a single float, which after applying a *sigmoid* function σ results in a probability of membership. This model is then trained with all synthetic shadow datasets using a binary cross-entropy loss function with the known binary value for membership.

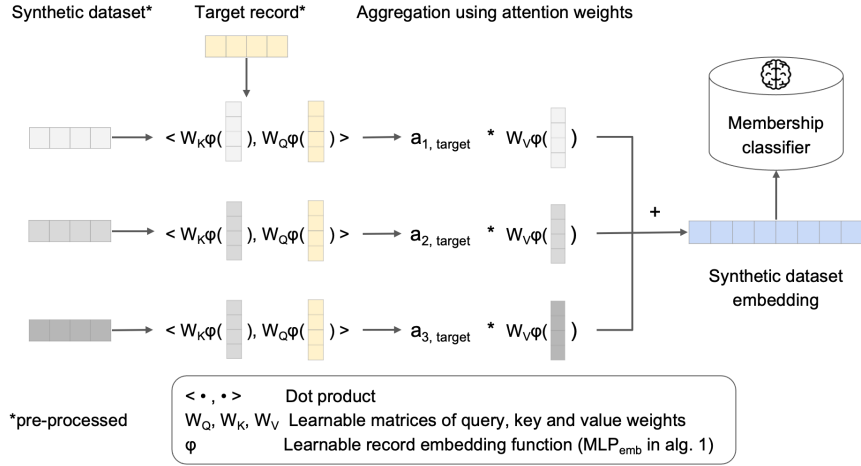


Figure 3.1: Illustration of our target attention architecture.

3.4.3 Baselines for identifying vulnerable records

We compare our approach against three baselines to identify vulnerable records:

Random: randomly sample target records from the entire population.

Rare value (Groundhog) [SOT22] sample a target record that either has a rare value for a categorical attribute or a value for a continuous attribute that is larger than the 95th percentile of the respective attribute.

Log-likelihood (TAPAS) [HJC⁺22]: sample target records having the lowest log-likelihood, assuming attribute values are independently drawn from their respective empirical distributions.

Note that this approach is defined by the authors only for categorical attributes $f \in \mathcal{F}_{\text{cat}}$: for each record x_i , they compute the frequency of the value $x_{i,f}$ in the entire dataset, resulting in $p_{i,f}$. The likelihood of a record is defined by the product of $p_{j,f}$ across all attributes. We here extend this approach to also account for continuous attributes $f \in \mathcal{F}_{\text{cont}}$ by discretising them according to percentiles.

To evaluate the different methods to select vulnerable records, we run the query-based MIA described in Section 3.4.1 on the ten records identified by each of the three methods as well as our own approach.

Throughout the experiments, we used the three generative models described in Chapter 2.1.2 (SynthPop, Baynet and its DP version, PrivBayes) using the implementation available in the `reprosyn` repository [Ala22].

We evaluate our method using two publicly available datasets (Adult and UK Census) described in Appendix B.1.

Algorithm 1 Forward propagation for target-attention

1: Inputs:

1. $D^{s,p} \in \mathbb{R}^{X \times F'}$ the preprocessed synthetic dataset with its unique records, their multiplicity and distance to the target record.
2. The pre-processed target record values $x_{t,p} \in \mathbb{R}^{F'}$.
3. Learnable MLP for the embedding MLP_{emb} with F' input features and F_{emb} output features, weight matrices $\mathbf{W}_q$, $\mathbf{W}_k$, and $\mathbf{W}_v \in \mathbb{R}^{F_{emb} \times F_{att}}$ to compute the target query and record keys and values with attention size F_{att} , and MLP for the prediction MLP_{pred} with F_{att} input features and 1 output feature

2: Output:

$h_{pred} \in \mathbb{R}$, which leads to the predicted probability for binary membership after applying a sigmoid transformation $\sigma(h_{pred})$

3: // Compute the embeddings of all records and the target record

4: $D_{emb} \leftarrow \text{MLP}_{emb}[D^{s,p}]$ with $D_{emb} \in \mathbb{R}^{X \times F_{emb}}$

5: $x_{t,emb} \leftarrow \text{MLP}_{emb}[x_{t,p}]$ with $x_{t,emb} \in \mathbb{R}^{1 \times F_{emb}}$

6: // Compute the target query, and the keys and values for each record

7: $q_t \leftarrow \mathbf{W}_q x_{t,emb}$ with $q_t \in \mathbb{R}^{1 \times F_{att}}$

8: $K \leftarrow \mathbf{W}_k D_{emb}$ with $K \in \mathbb{R}^{X \times F_{att}}$

9: $V \leftarrow \mathbf{W}_v D_{emb}$ with $V \in \mathbb{R}^{X \times F_{att}}$

10: // Compute the target attention scores for each record

11: $a_j \leftarrow \langle q_t, v_j \rangle$ for $j = 1..X$ to get $a \in \mathbb{R}^{1 \times X}$

12: $a_j \leftarrow \exp a_j / (\sum_{j=1}^X \exp a_j)$ for $j = 1..X$

13: $D_{att} \leftarrow aV$ with $D_{att} \in \mathbb{R}^{1 \times F_{att}}$ **// Sum values weighted by target attention**

14: $h_{pred} \leftarrow \sigma(\text{MLP}_{pred}[D_{att}])$ **// Compute the final membership score.**

3.4.4 Parameters of the attack

Table C 1 in Appendix C shows the parameters we use for the attack. We use a given dataset Ω (e.g. Adult) to select our most at-risk records as the target records. We then partition Ω randomly into D_{test} and D_{aux} . D_{aux} is the auxiliary knowledge made available to the adversary, while D_{test} is used to sample n_{test} datasets used to compute the attack performance. n_{shadow} represents the number of shadow datasets used, sampled from the auxiliary knowledge D_{aux} . We considered, in all experiments, the size of the release synthetic dataset (D^s) to be equal to

the size of the private dataset D , $|D| = |D^s| = 1000$. To create the shadow datasets, while ensuring that D_{aux} does not contain the target record, we randomly sample $|D| - 1$ records in D_{aux} . Then we add the target record x_T in half of the datasets, and in the other half, we used another record randomly sampled from D_{aux} .

We run the query-based attack using $N = 100000$ queries randomly sampled from all possible count queries (2^F in total) and a random forest meta-classifier with 100 trees, each with a maximum depth of 10.

We used $F_{emb} = 20$, $F_{att} = 15$ and $|top\ X| = 100$ for the target attention attack. Both MLP_{emb} and MLP_{pred} consist of one hidden layer with 20 and 10 nodes respectively, and use ReLU as activation and a dropout ratio of 0.15. The network is trained for 500 epochs, using the Adamax optimizer [KB14] with a learning rate of $10e - 3$ and batch size of 20. 10% of the training data is used as validation data and the model that achieves the lowest validation loss is selected.

In order to measure attack performances, we compute the Area Under the receiver operating characteristic Curve (AUC) for the binary classification of membership on n_{test} datasets.

Lastly, in line with the auditing scenario, we use $k = 5$ neighbors in the private dataset Ω to select the most vulnerable records using our vulnerability score V_k .

Table 3.1: Mean and standard deviation for AUC for the query-based attack method

Method	UK census		Adult	
	Synthpop	BayNet	Synthpop	BayNet
Random	0.732 ± 0.046	0.535 ± 0.055	0.613 ± 0.084	0.618 ± 0.120
Rare value (Groundhog)	0.802 ± 0.044	0.644 ± 0.086	0.699 ± 0.113	0.704 ± 0.140
Log-likelihood (TAPAS)	0.790 ± 0.041	0.731 ± 0.090	0.742 ± 0.033	0.787 ± 0.057
Distance (ours)	0.879 ± 0.021	0.858 ± 0.040	0.804 ± 0.072	0.810 ± 0.110

3.5 Results

3.5.1 Performance of vulnerable record identification methods

We evaluate the effectiveness of vulnerable record identification methods across the two datasets UK Census and Adult, and two synthetic data generators Synthpop and BayNet. For each dataset, each of the four record identification methods selects 10 target records. We then perform the MIA, as a privacy game, on each of these records and report its AUC.

Figure 3.2 show that our vulnerable record identification method consistently outperforms both the random baseline and the two other techniques previously used in the literature. The median risk of records identified by our method is indeed consistently higher than all other methods, up to 18.6 p.p. higher on the UK Census dataset using BayNet. On average, the AUC of the records we identified are 7.2 p.p. higher than the AUC of the records selected by other methods (Table 3.1). Importantly, our method also consistently manages to identify records that are more or equivalently at-risk compared to other methods.

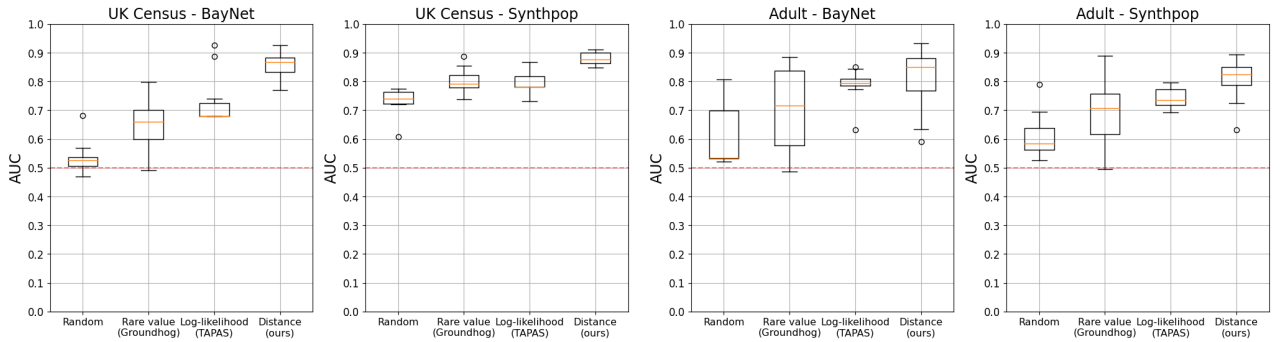


Figure 3.2: **AUC of MIAs across record identification techniques for the query-based attack**, for datasets UK Census and Adult and synthetic data generators BayNet and Synthpop.

3.5.2 Applicability of the method to the target attention attack

We follow the same methodology as above to evaluate the effectiveness of the four vulnerable record identification methods when using the target attention approach.

Figure 3.3 shows that, again, our vulnerable record identification method strongly outperforms both the random baseline and the two other techniques previously used in the literature. The mean risk of records identified by our method is again consistently higher than all other methods, up to 10.89 p.p. higher on UK Census using BayNet. Across datasets and generators, the records we identify are 5.2 p.p. more at risk than the records selected by other methods (Table 3.2).

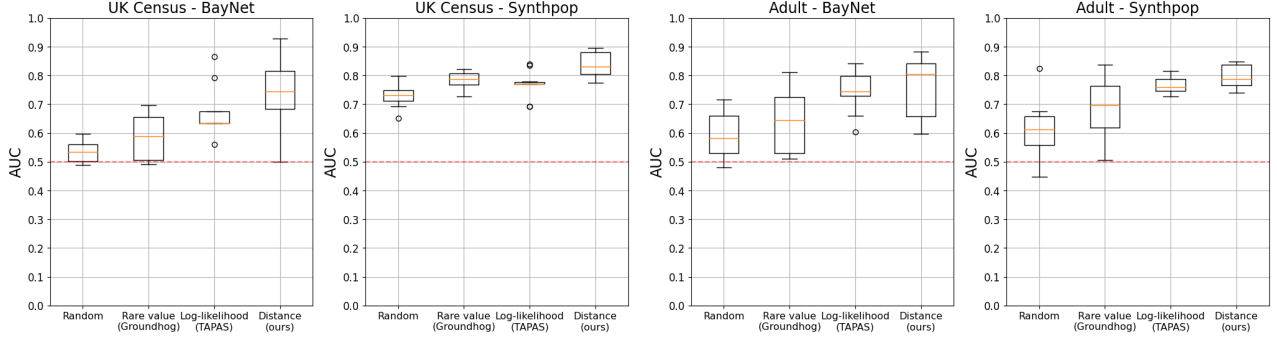


Figure 3.3: **AUC of MIAs across record identification techniques for the target attention attack**, for datasets UK Census and Adult and synthetic data generators BayNet and Synthpop.

Table 3.2: **Mean and standard deviation for AUC for the target attention method**

Method	UK census		Adult	
	Synthpop	BayNet	Synthpop	BayNet
Random	0.727 ± 0.038	0.535 ± 0.035	0.609 ± 0.112	0.594 ± 0.083
Rare value (Groundhog)	0.783 ± 0.030	0.587 ± 0.079	0.691 ± 0.103	0.638 ± 0.104
Log-likelihood (TAPAS)	0.770 ± 0.046	0.674 ± 0.085	0.765 ± 0.027	0.745 ± 0.068
Distance (ours)	0.838 ± 0.041	0.737 ± 0.118	0.797 ± 0.038	0.763 ± 0.103

3.5.3 Robustness analysis for k

Throughout this chapter, we use $k = 5$, meaning that our vulnerability score $V_k(x_i)$ is computed using the 5 closest records to the record of interest. We here evaluate the impact of a specific choice of k for the record identification and associated results. More specifically, we compute the mean distance for each record using values of k ranging from 1 to 50 and report the mean AUC for the ten records selected by the method on the Adult dataset using Synthpop.

Figure 3.4(a) shows our method to be robust to specific choices of k . The mean AUC of the top ten records varies indeed only slightly for different values of k . Looking at the top 10 records

selected for different values of k , we find that only a handful of records are distinct, suggesting our method to not be too sensitive to specific values of k . We had to select a number of neighbors, k , high enough so that it is harder to end up on a dataset consisting only of clusters of records of size k , which would bias the selection of outliers (since every single record is inside a cluster of size k , this would artificially suggest that there are no real outliers, even if the clusters are far apart from one another). However, k also had to be low enough to minimize computational cost, and prevent a reduction in the AUC of the top 10 records. That is why we selected $k = 5$ in this chapter.

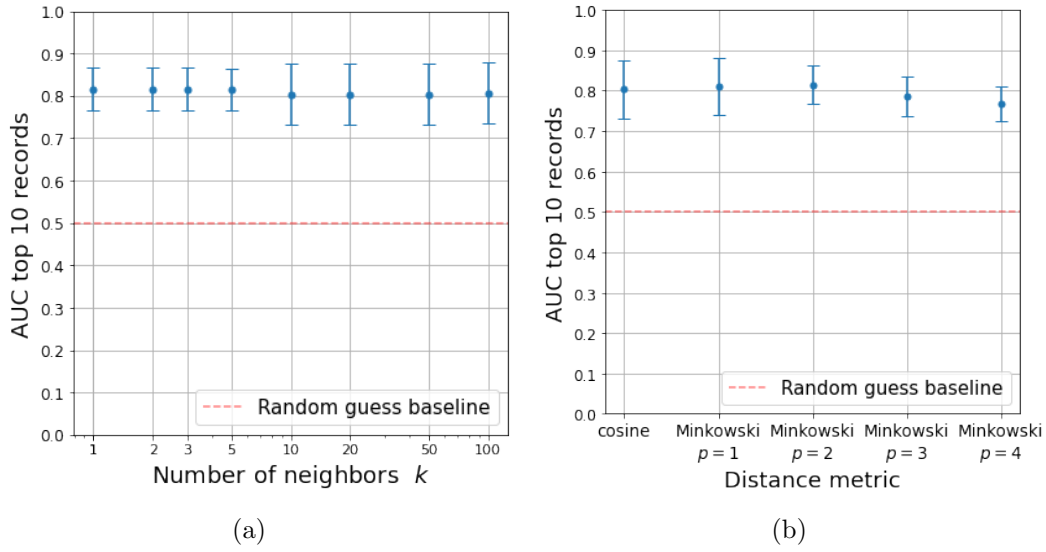


Figure 3.4: **Mean and standard deviation of the AUC of MIAs for 10 records selected by the vulnerability score in various settings.** We vary in (a) the number of neighbors considered and in (b) the distance metrics considered. Results on Adult, Synthpop for the query-based attack.

3.5.4 Robustness analysis for the cosine distance

We here rely on a generalized form of cosine distance across attribute types. We now evaluate the susceptibility of our results to different choices of distance metrics. We do this using several versions of the Minkowski distance between two vectors $a, b \in \mathbb{R}^L$. For a given value $p \in \mathbb{N}^*$, the Minkowski distance is defined as $d_{\text{Minkowski}}^p(a, b) = \sqrt[p]{\sum_{l=1}^L |a_l - b_l|^p}$. Similarly to Equation 3.5,

we generalise this distance metric for two records $x_i, x_j \in D$ to:

$$d(x_i, x_j) := \frac{|\mathcal{F}_{\text{cat}}|}{F} d_{\text{Minkowski}}^p(h(x_i), h(x_j)) + \frac{|\mathcal{F}_{\text{cont}}|}{F} d_{\text{Minkowski}}^p(c(x_i), c(x_j)). \quad (3.6)$$

Notably, the Minkowski distance with $p = 2$ closely relates to the cosine distance. As before, we compute the mean distance for each record using Minkowski distances with p ranging from 1 to 4 and report the mean AUC for the ten records selected by the method on the Adult dataset using Synthpop.

Figure 3.4(b) shows the attack performance to, again, be robust to different choices of distance metric. In particular, mean AUC is not significantly different for the cosine distance and $p \in \{1, 2\}$, while slightly decreasing for higher p values. Cosine being the standard distance metric in the literature, we prefer to use it.

3.5.5 Vulnerable records and differentially private generators

An active line of research aims at developing differentially private (DP) synthetic data generators. As discussed above however, developing models that achieve DP with meaningful theoretical ϵ while maintaining high utility across tasks remains a challenge. MIAs are thus an important tool to understand the privacy protection offered by DP models for various values of ϵ .

We here evaluate the performance of our vulnerable record identification approach when DP generators are used. Specifically, we report the mean and standard deviation AUC for the 10 records identified as being at risk by our approach and by the three baselines for varying values of ϵ using PrivBayes [ZCP⁺17].

Figure 3.5 shows that our approach outperforms both the random baseline and the two other techniques previously used in the literature. As expected though, the AUC drops when ϵ decreases. The attack still performs quite well for $\epsilon = 100$ and manages to identify at-risk records for $\epsilon = 10$. Importantly, the general utility of synthetic data generated by PrivBayes

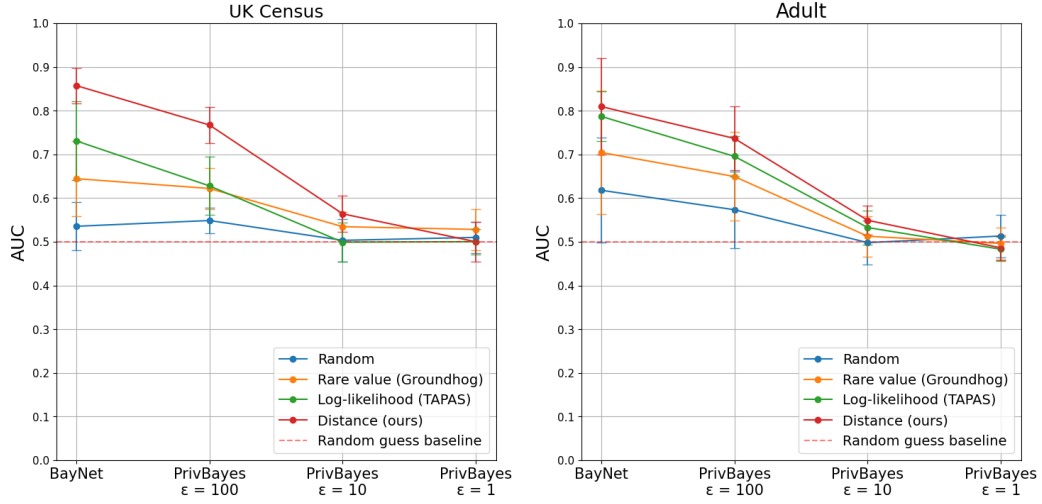


Figure 3.5: Mean and standard deviation of the AUC of the MIAs for varying values of ϵ in DP-based generator PrivBayes. Results on UK Census (left) and Adult (right) for the query-based attack and for the 10 points for each record identification method.

for low value of ϵ is being debated.

3.6 Discussion

Our results show that our simple distance-based method is able to consistently identify vulnerable records, and more effectively so than previously proposed methods. We believe this to be due to limitations of previous work. Indeed *Rare value (Groundhog)* only considers the value of one specific attribute to determine whether a record is at risk while *Log-likelihood (TAPAS)* assumes independence across all attributes to compute a record’s likelihood, an assumption that is typically not met in real-world datasets. In contrast, our distance-based metric considers the neighborhood of the record, therefore encompassing all attributes without making any assumptions on the underlying distribution.

Across datasets and generators, we find that records identified by our method are on average 7.2 p.p. more vulnerable than previous work when using the state-of-the-art MIA. In practice, we believe this to be highly significant. Researchers and practitioners who wish to evaluate new attack methods and the privacy risk of synthetic data publishing are indeed unable to evaluate all records in the dataset, due to the computational cost of shadow modeling (see

3.1). Accurately identifying the records that are most at risk is thus as essential as using the state-of-the-art MIA when evaluating the privacy risk of a synthetic dataset. Indeed, data protection laws such as the EU GDPR require to consider the privacy protection offered to all data subjects when evaluating if a data release is anonymous.

When we use only categorical columns, such as in UK Census, results show that we reduce the standard deviation in the results by a factor of two, across generators. With our method we obtained a standard deviation of 0.021 instead of 0.041 for Synthpop and 0.040 instead of 0.090 for BayNet. This shows that our method selects records with high precision of having high vulnerability.

Lastly, apart from being effective, our principled method is compellingly easy. By being independent of the synthetic data generator and the dataset, it allows for pragmatic and efficient cross-generator and dataset comparison.

Future work. We here distinguished between categorical features and continuous attributes. Future research could explore if considering ordinal as a subset of the categorical features could lead to a better record selection. Similarly, we here focused on the difference in metrics between the cosine similarity and the Minkowski metrics of order $p \in \{1, 2, 3, 4\}$. We leave for future work to see if other distances, such as Minkowski of higher order, the Chebyshev distance or Hamming distance, could lead to a better record selection.

3.7 Conclusion

Due to computational costs (it takes between 1.5 to 6 hours to run an MIA against a single record in our datasets, conducting this approach for every record in the dataset would take more than 250 days of compute), the identification of vulnerable records is as essential as the actual MIA methodology when evaluating the privacy risks of synthetic data, including with respect to data protection regulations such as the EU GDPR. In this chapter, we propose and extensively evaluate a simple yet effective method to select vulnerable records in synthetic data publishing, which we show to strongly outperform previous approaches.

We need to ensure that each vulnerability score is correctly computed (with respect to the distribution of the target dataset), based on the outliers we identified. In the next chapter, we will aim to provide a better evaluation of the vulnerability score.

Chapter 4

A better evaluation of the vulnerability score for machine learning models

The work presented in this chapter is under review, and can be found as an arXiv preprint as: Guépin, F., Krčo, N.*, Meeus, M., & de Montjoye, Y. A. (2024). Lost in the Averages: A New Specific Setup to Evaluate Membership Inference Attacks Against Machine Learning Models. arXiv preprint arXiv:2405.15423.*

Statement of contributions: Nataša Krčo and I are joint first authors on this paper. We collaborated in the design of the methods, experiments, writing of the code and the paper. We equally contributed to the writing of the code, running experiments, production of the figures and writing of the paper.

4.1 Introduction

Machine Learning (ML) models are increasingly trained or fine-tuned by companies, governments, and academic researchers often using data that can be personal. Membership inference attacks (MIAs) have become the standard method to evaluate the privacy risks of aggregate data [HSR⁺08, SOJH09] including when releasing ML models [SSSS17, CCN⁺22]. MIAs quan-

tify the risk of an adversary being able to infer that the target record was a member of the released model’s training set.

MIAs against ML models are typically instantiated as a binary meta-classifier, predicting membership for a given target record [SSSS17, CCN⁺22, SOT22]. The meta-classifier is trained using auxiliary data, and then evaluated on models trained on datasets randomly sampled from a large held-out data pool dedicated to evaluation. The target record is then added to a fraction of the datasets, typically to half of them to create a balanced setup.

Though using dataset sampling as a source of randomness is a natural extension of the ML evaluation pipeline to MIAs, we argue that—in light of recent evidence—this does not properly evaluate the privacy risk posed by an ML model release in practice. Recent research on MIAs against machine learning models has indeed shown how the privacy risk often lies with particular records being memorized [CCN⁺22, MGCdM23, AZT24, SOT22], leading to the development of techniques to identify them [MGCdM23] and new metrics—such as TPR at low FPR—being adopted [CCN⁺22]. Recent evidence [CJZ⁺22, MGCdM23] also suggest that at-risk records tend to be outliers, records that are different from other records in the dataset used to train the model. Using data sampling as a source of randomness to evaluate the privacy risk of releasing is therefore problematic as it evaluates the *average* risk of a target record in a dataset sampled from the underlying distribution instead of the risk posed by the release of the actual model which was trained on a specific dataset.

Contribution. In this chapter, we propose a *specific* evaluation setup for MIAs against ML models which we argue, in light of recent evidence, better captures the privacy risk of releasing an ML model.

First, we formalize the *average* MIA evaluation setup as currently used in the literature with dataset sampling as a source of randomness. We then propose a *specific* evaluation setup which only uses weight initialization as a source of randomness which we argue better captures the risk posed by the release of a given ML model.

Second, we instantiate both evaluation setups using existing state-of-the-art MIAs against

synthetic data generators [HJC⁺22, MGCdM23] and ML models. We show the results from existing MIA attacks to strongly differ when evaluated using the current, average, setup and the more accurate specific setup we propose. For instance, we find that 94% of high-risk records in the Adult dataset would be incorrectly considered as low-risk when using the average setup to evaluate a model trained using synthpop. We validate our finding against a range of datasets and synthetic data models and classifiers.

Third, we derive theoretical results which, combined with empirical evidence, strongly suggest that the risk calculated in the current setup is an average of the risks specific to each dataset sampled for testing. We argue these results validate our use of weight initialization as the (only) source of randomness necessary to evaluate MIAs against ML models.

Finally, we instantiate an MIA attack by a very strong adversary with knowledge of the training dataset D_{target} , a version of the standard adversary in Differential Privacy. We show this attack to lead to drastically and significantly improved membership inference when evaluated in the specific setup (mean AUC going from 0.601 to 0.792) but not in the average setup. These results further validate the need to use a specific setup when evaluating MIAs against machine learning models and emphasize the fact that new stronger attacks, leveraging (some) knowledge of the training dataset, are likely to be developed in the future.

4.2 Related work

Membership Inference Attacks (MIAs). Homer et al. [HSR⁺08] first introduced the concept of MIAs to infer the presence of a trace amount of DNA from released genomic aggregates. Their work was then extended to study how the parameters, e.g. the number of records in the aggregates, impact MIA performance [SOJH09]. MIAs have since been applied to other types of data aggregates for instance location traces [PTDC17] and, most importantly for this work, machine learning (ML) models [SSSS17].

In parallel, new techniques have been proposed to carry out MIAs [LF20, NSH19, YGFJ18, SZH⁺18, CCN⁺22, SOT22]. While the 2008 attack by Homer et al. used statistical hypothesis

techniques, most attacks today rely on the shadow modeling technique introduced by Ateniese et al. [AMS⁺15] and popularized by Shokri et al. [SSSS17]. In particular, LiRA, which combines shadow modelling and statistical testing to determine membership, is the state-of-the-art black-box MIA against ML models [CCN⁺22]. While out of scope for us, attacks that rely only on access to the labels predicted by a model have been developed e.g. Choquette-Choo et al. [CCTCP21] as well as white box attacks e.g. Nasr et al. [NSH19] which uses the gradients and activations of the layers of a deep neural network to predict membership. Finally, black-box attacks leveraging shadow models have been developed against synthetic data generators. These include Stadler et al. [SOT22] and the state-of-the-art query-based attack which we use here by Houssiau et al. [HJC⁺22] (see Chapter 3.4.1 for details).

MIA performance evaluation. To the best of our knowledge, all methods so far evaluate the record-level privacy risk using dataset sampling as a source of randomness. AUC or, alternatively, accuracy were the primary metrics used to measure the success of an MIA against a target record, usually with a balanced test set [SSSS17, CCTCP21, HMDDC19]. As new research emphasizing how certain records might be much more at risk than others [FZ20, MGCdM23, CCN⁺22, SOT22] developed, metrics focusing on the most at-risk records started to be used. This includes metrics focusing on the records identified a posteriori as being particularly at risk e.g. in synthetic datasets [SOT22, MGCdM23] and metrics measuring the risk the most vulnerable records across the entire dataset are exposed to, such as TPR (True Positive Rate) at low FPR (False Positive Rate). Popularized by Carlini et al. [CCN⁺22] for MIAs, TPR at low FPR has become a standard metric to measure the effectiveness of MIAs against the most vulnerable records used to train an ML model. Recent evidence [CJZ⁺22, MGCdM23] also suggests that at-risk records tend to be outliers, records that are different from other records in the dataset used to trained the model.

4.3 MIA evaluation setups

4.3.1 Notation

Evaluation setup. We denote by D_{target} the *target* dataset used to train the target machine learning model $\mathcal{M}_{target}(D_{target})$ and x_{target} the *target* record whose membership the adversary aims to infer. Note that for simplicity we use $\mathcal{M}_{target}$ as a shorthand for $\mathcal{M}_{target}(D_{target})$.

To accommodate—without loss of generality—the average and specific setups, we denote D_{eval} the evaluation data pool available to the model developer to evaluate the privacy risk of releasing a trained model $\mathcal{M}_{target}$, including the dataset used to train $\mathcal{M}_{target}$ (Note that at all time, we have $D_{target} \subset D_{eval}$).

We also use the term *evaluation models* for the models we use to evaluate the effectiveness of an MIA and *evaluation datasets* for the datasets they are trained on.

Finally, an auxiliary dataset D_{aux} is used to train the MIA. We here first consider MIAs trained on D_{aux} , a dataset drawn from the same distribution as D_{eval} (but strictly not overlapping), the most standard assumption in the literature [SSSS17, CCN⁺22]. In Section 4.7 specifically, we also develop an MIA where a very strong adversary, akin to the Differential Privacy adversary, has access to D_{target} minus the membership information of x_{target} as auxiliary dataset.

Privacy risk. We denote by $R_{setup}^{\phi}(x, D)$ the privacy risk for the record x when evaluation datasets are sampled from D and metric ϕ is used to calculate the performance score of the MIA. For metric ϕ , we use both ROC AUC and TPR at low FPR (further simplified as TPR at an FPR of 0.1 except in Section 4.5 where we use 0.3).

4.3.2 Average evaluation setup

The evaluation setup used in the literature relies on the sampling of evaluation datasets as a source of randomness to evaluate the effectiveness of an MIA against an ML model. The privacy

risk R_{avg}^ϕ of a target record x_{target} is evaluated on the trained MIA's ability to distinguish between ML models trained on a dataset containing x_{target} and those not. The source of randomness in this setup is the sampling of evaluation datasets - N different evaluation datasets are sampled from D_{eval} and x_{target} is included in half of them. A separate evaluation model is trained on each evaluation dataset, and the trained MIA is applied to it. The privacy risk for x_{target} is then estimated as the performance of the MIA aggregated across the evaluation models. A diagram of this setup is presented on the left-hand side of Figure 4.1. We write $R_{avg}^\phi(x, D_{eval})$ (simplified as R_{avg}^ϕ) for the risk estimated using the average setup using metric ϕ .

Detailed evaluation steps. We here provide detailed steps for the average evaluation setup.

All details can be found in Algorithm 2.

Algorithm 2 Average MIA evaluation setup

We present the pipeline for the average evaluation setup for membership inference attacks. We denote with $f_{\mathcal{M}}$ the trained meta-classifier (MIA). D_{eval} is the evaluation pool available to the adversary which has not been used to train $f_{\mathcal{M}}$. $D_{target} \subset D_{eval}$ is the target dataset, i.e. the dataset that will be used to train the model that will be released, and $x_{target} \in D_{target}$ is the target record.

- 1: Instantiate prediction set $Y_{pred} = \emptyset$.
 - 2: Instantiate the training seed array l_{seed} , $|l_{seed}| = N$
 - 3: **for** $i = 0 \dots \frac{N}{2}$ **do**
 - 4: Sample $D_{IN} \sim D_{eval} \setminus \{x_{target}\}$, $|D_{IN}| = |D_{target}| - 1$, and add x_{target} to D_{IN}
 - 5: Sample $D_{OUT} \sim D_{eval} \setminus \{x_{target}\}$, $|D_{OUT}| = |D_{target}|$
 - 6: Train evaluation models $\mathcal{M}_{IN}$ with seed $l_{seed}[2 * i]$ and $\mathcal{M}_{OUT}$ with $l_{seed}[2 * i + 1]$.
 - 7: Add label-prediction pairs $(IN, f_{\mathcal{M}}(\mathcal{M}_{IN}))$ and $(OUT, f_{\mathcal{M}}(\mathcal{M}_{OUT}))$ to Y_{pred}
 - 8: **end for**
 - 9: Calculate privacy risk using chosen metric based on prediction set Y_{pred}
-

4.3.3 Specific evaluation setup

In the specific MIA evaluation setup, we remove the randomness coming from the sampling of evaluation datasets, and use the the weight initialization of the evaluation models as the sole source of randomness. More specifically, we train $\frac{N}{2}$ evaluation models on D_{target} , ensuring that each has a different random initialization. For the remaining $\frac{N}{2}$ evaluation models, we

train on D_{target} where x_{target} is swapped out for a reference record x_R , sampled from D_{eval} . We estimate the privacy risk of x_{target} by calculating a performance score over all N evaluation models. Similarly to the previous section, we write here $R_{sp}^\phi(x, D_{eval})$ (simplified as R_{sp}^ϕ) for the risk estimated using the specific setup using metric ϕ . The right-hand side of Figure 4.1 shows a diagram of this setup.

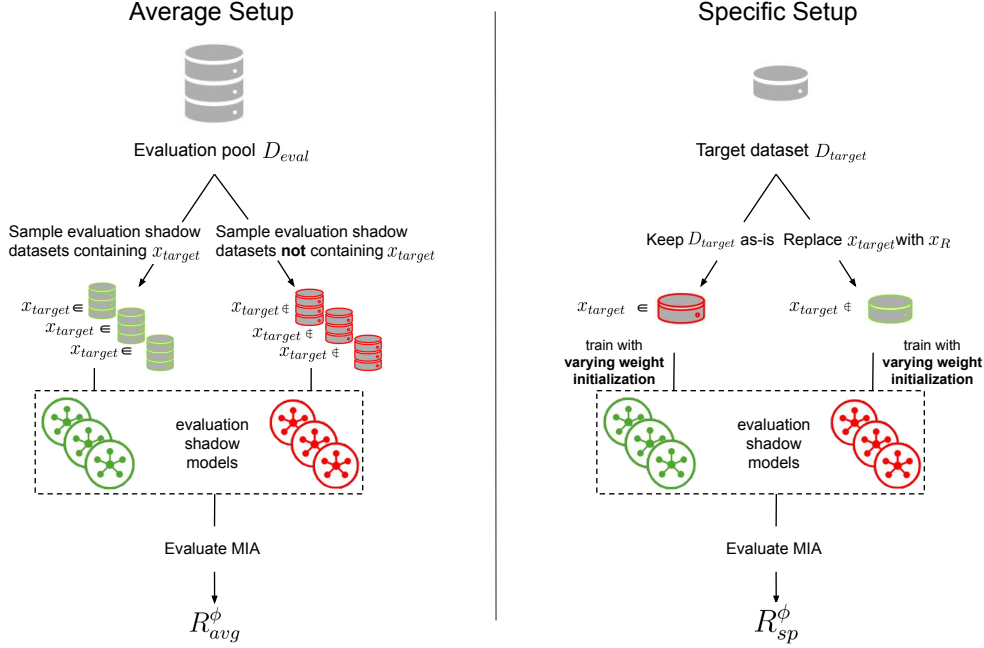


Figure 4.1: **Presentation of the evaluation setups.** On the left: the average evaluation setup, sampling multiple datasets from the evaluation pool D_{eval} and training an evaluation model on each. On the right: the specific evaluation setup: the only evaluation datasets used are the full D_{target} , in which x_{target} is included, and D_{target} in which x_{target} is replaced by a reference record x_R sampled from the evaluation pool D_{eval} .

Details for the specific evaluation setup can be found in Algorithm 3.

4.3.4 Comparing setups

Miss rate. We define the *miss rate* to be the fraction of records classified as high-risk in the specific setup, that are classified as low-risk in the average setup. Given a threshold t , we consider a record x to be high-risk if $R_{setup}^\phi(x, D) > t$, and low-risk otherwise, for $setup \in$

Algorithm 3 Specific MIA evaluation setup

We present the pipeline for the specific evaluation setup for membership inference attacks. We denote with $f_{\mathcal{M}}$ the trained meta-classifier (MIA). D_{test} is the data available to the adversary which has not been used to train $f_{\mathcal{M}}$. $D_{target} \subset D_{eval}$ the target dataset, i.e. the dataset that will be used to train the model that will be released, $x_{target} \in D_{target}$ the target record, and reference record $x_R \in D_{eval} \setminus D_{target}$.

- 1: Instantiate prediction set $Y_{pred} = \emptyset$.
- 2: Instantiate model initialization seed array l_{seed} , $|l_{seed}| = N$
- 3: **for** $i = 0 \dots \frac{N}{2}$ **do**
- 4: Train evaluation model $\mathcal{M}_{IN}$ with initialization seed $l_{seed}[2 * i]$ on $D_{IN} = D_{target}$
- 5: Sample reference record x_R from $D_{eval} \setminus D_{target}$
- 6: Remove x_{target} from D_{IN} and replace it with x_R to construct dataset D_{OUT}
- 7: Train model $\mathcal{M}_{OUT}$ with seed $l_{seed}[2 * i + 1]$
- 8: Add label-prediction pairs $(IN, f_{\mathcal{M}}(\mathcal{M}_{IN}))$ and $(OUT, f_{\mathcal{M}}(\mathcal{M}_{OUT}))$ to Y_{pred}
- 9: **end for**
- 10: Calculate privacy risk using chosen metric based on prediction set Y_{pred}

$\{avg, sp\}$, dataset D , and metric ϕ . Formally, we define the *miss rate* $\forall S \subseteq D_{target}$ as:

$$\mathcal{M}(S, D_{target}, D_{eval}) = \frac{|\{x \in S | R_{avg}^{\phi}(x, D_{eval}) \leq t \wedge R_{sp}^{\phi}(x, D_{target}) > t\}|}{|\{x \in S | R_{sp}^{\phi}(x, D_{target}) > t\}|}$$

RMSE and Pearson correlation. We also use Root Mean Squared Error (*RMSE*) and Pearson correlation to compare the results in the specific and average evaluation setups.

RMSE quantifies the error made when using the average setup instead of specific while the Pearson correlation allows us to compare our empirical results to what we would expect theoretically (see Theorem 4.1). Formally, we compute them $\forall S \subseteq D_{target}$ as:

$$RMSE(S, \phi) = \sqrt{\frac{1}{|S|} \sum_{x \in S} (R_{avg}^{\phi} - R_{sp}^{\phi})^2}$$

$$\rho(S, \phi) = \frac{\sum_{x \in S} (R_{avg}^{\phi} - \bar{R}_{avg}^{\phi}) \cdot (R_{sp}^{\phi} - \bar{R}_{sp}^{\phi})}{\sqrt{\sum_{x \in S} (R_{avg}^{\phi} - \bar{R}_{avg}^{\phi})^2 \cdot \sum_{x \in S} (R_{sp}^{\phi} - \bar{R}_{sp}^{\phi})^2}}$$

Where $\bar{R}_{avg}^{\phi} = \frac{1}{|S|} \sum_{x \in S} R_{avg}^{\phi}$ is the mean average risk and $\bar{R}_{sp}^{\phi} = \frac{1}{|S|} \sum_{x \in S} R_{sp}^{\phi}$ is the mean specific risk.

Note that all the measures used to compare setups are specific to the metric ϕ used to calculate

the performance score of the MIA. In particular, AUC typically ranges from 0.5 to 1 while TPR at low FPR ranges from 0 to 1. The metrics might furthermore be distributed differently. Measures computed using $\phi = \text{AUC}$ can thus not be directly compared to measures computing $\phi = \text{TPR}$. We also use different thresholds for which records are considered high-risk namely $t = 0.8$ for AUC and $t = 0.6$ for TPR.

4.4 MIAs against generative models

With synthetic data becoming a popular privacy-preserving technology in practice [Gar18, Dr 23, Syn24], we first compare both evaluation setups when evaluating SOTA MIAs against synthetic data generators.

Experimental setup. We use the SOTA MIA against synthetic data, the query-based attack, introduced by Houssiau et al. [HJC⁺22] (see Chapter 3.4.1 for more details). We consider two models: synthetic data generators Synthpop [Haz19] and BayNet [ZCP⁺17] (see Chapter 2.1.2 for more details), and two datasets: Adult [RB96] and UK Census [Off11] (see Appendix B.1 for more details). D_{eval} used to evaluate the MIA contains 15,222 records for Adult and 27,193 for UK census. We always consider D_{target} to contain 1000 records. We compute the privacy risk on all records in D_{target} for Adult and, for computational efficiency (per record, each MIA takes about 4 hours), on a random subset of 100 records for UK census. The auxiliary dataset available to train the MIA contains 30,000 records for Adult and 52,390 records for UK census and is strictly not overlapping with D_{eval} .

Discussion on the number of shadow models used. We run an experiment to determine the best number of evaluation models N for the Adult dataset and $\mathcal{M}_{target}$ Synthpop (Figure 4.2). For value N , we perform the full average and specific evaluation pipeline using N evaluation models, and calculate and apply the function $\log$ to the AUC. We repeat this 10 times for the same N , each time sampling new evaluation datasets in the average setup, and training new evaluation models in both setups. For each iteration, we calculate and apply the

function $\log$ to the AUC value. We then calculate the standard deviation of the 10 AUC values for N . We do this process for $N \in \{100, 200, 300, \dots, 2000\}$ for 10 target records. Figure 4.2(a) presents the standard deviations per N , aggregated over the 10 records. We select $N = 1000$, as the standard deviation converges to approximately 0.01 at that point.

We do the same experiment for Adult and $\mathcal{M}_{target}$ a binary classifier in Figure 4.2(b). In this case, we use 5 records and repeat the evaluation pipelines 5 times. We run for less records and times due to the computational cost of this experiment. For $\mathcal{M}_{target}$ a binary classifier, we select $N = 500$, as it provides a sufficiently low standard deviation (0.02), while still being feasible to conduct our main experiments.

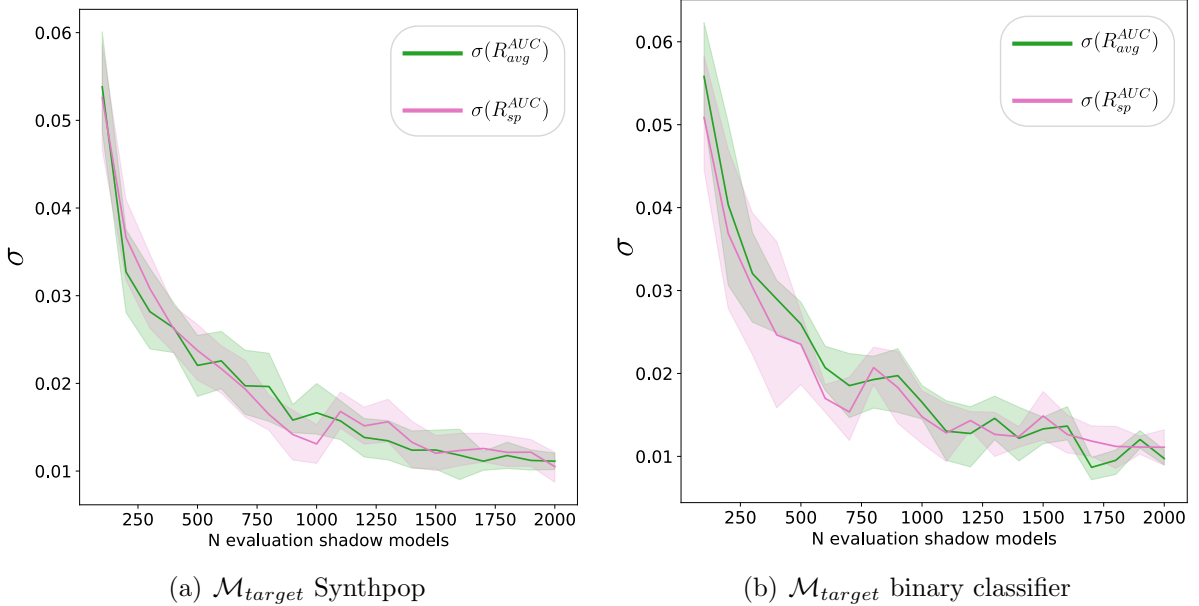


Figure 4.2: **Standard deviation of the AUC as a function of the number of evaluation models used to calculate AUC Adult dataset.**

Implementation details. Table C 2 in Appendix C shows the details for training and evaluating the MIA for each dataset and target model. For target models Synthpop and BayNet (for both datasets), we use the query-based attack introduced by Houssiau et al. [HJC⁺22], and for ResNet and neural network (NN), we use LiRA introduced by Carlini et al. [CCN⁺22]. We train a standard ResNet and a neural network with 1 layer with 100 nodes.

Resources. All of our experiments, except those for CIFAR10, were conducted using only CPUs in series. We provide the following table to specify the time taken to train an MIA and run the specific and average evaluation pipeline on a single target record. Experiments were run using either a machine with 96 CPUs and 500GB of RAM or a machine with 200 CPUs and 910 GB of RAM.

Table 4.1: **CPU time to run each of our experiments in Chapter 4** on a single record

Target model $\mathcal{M}_{target}$	dataset	CPU time (hours)
Synthpop [Haz19]	Adult [RB96]	1.2
	Census [Off11]	2.2
BayNet [ZCP ⁺ 17]	Adult	2.5
	Census	4.2
Binary-classifier	Adult	1.0

To run the CIFAR10 experiments, we use the open-source FFCV [LIE⁺23] library to train each evaluation model in approximately 20 seconds on one NVIDIA A100 GPU. The total time taken to train an MIA and run both the specific and average evaluation pipelines is approximately 5.5 hours per record. We use a machine with two NVIDIA A100 GPUs and 490GB of RAM to run our experiments in parallel for multiple records.

4.4.1 Adult and $\mathcal{M}_{target}$ Synthpop

We first present the results and metrics for one popular model, Synthpop, and dataset Adult.

Figure 4.3(a) shows how the AUC obtained when evaluating the effectiveness of the SOTA MIA in the average setup is, at best, a highly imperfect estimate of the effectiveness of an MIA against the model actually released and the dataset used to train it (the specific setup). As shown by the area shaded in grey ($T=0.8$), 94% of high-risk records would indeed be incorrectly classified as low-risk by an auditor using the average setup.

Figure 4.3(b) further shows that the AUC for a given target record, a value that empirically ranges roughly from 0.5 to 1, would be off by more than 0.1 for 15% of the records in the target dataset and could go up to 0.26. Across all records, using the average setup leads to an error

(RSME) of 0.07.

We obtain similar results using another metric, $\phi = \text{TPR}$, instead of AUC. 95% of high-risk records would be incorrectly classified as low-risk ($T=0.6$) and using the average setup leads to an error (RSME) of 0.11.

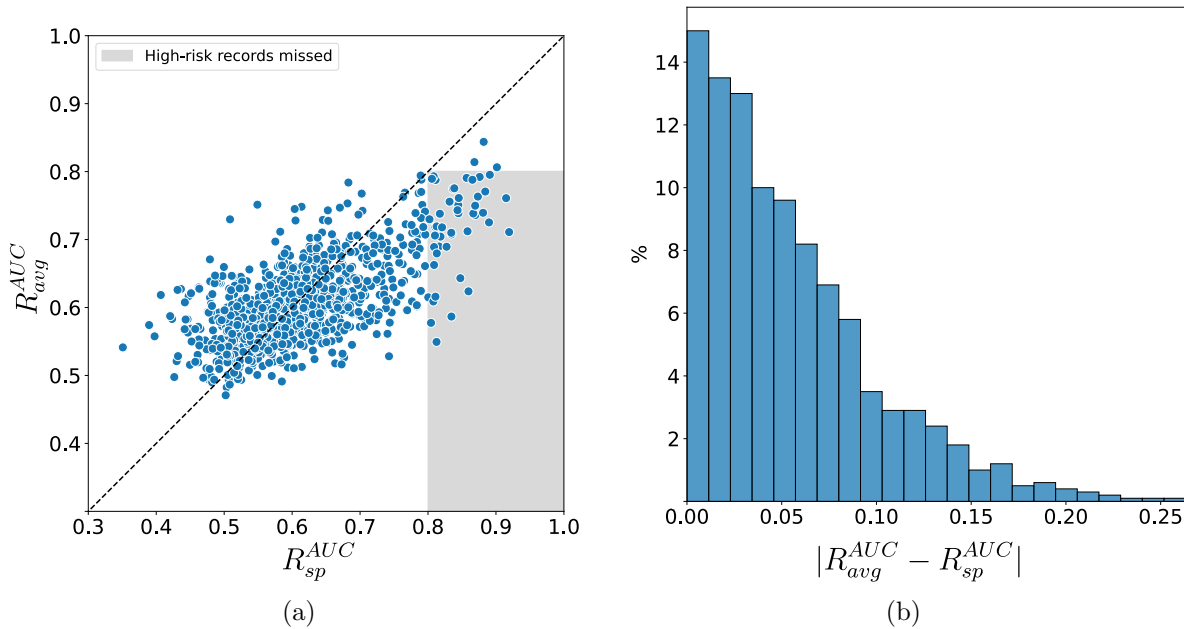


Figure 4.3: **Presentation of the privacy risks per record.** The privacy risk is computed for each record in D_{target} sampled from the Adult dataset, $\mathcal{M}_{target}$ Synthpop, and $\phi = AUC$. (a) per-record specific and average risks. The dashed grey lines mark the high-risk threshold $t = 0.8$. The shaded grey area marks all the high-risk records missed in the average setup. (b) histogram of per-record absolute differences between the specific and average risks.

4.4.2 Other generators and datasets

The results we have presented so far were for one popular model, Synthpop, and one dataset, Adult. We now present the complete results for the two models, Synthpop and BayNet, and two datasets, Adult and UK census.

Table 4.2 shows that we obtain similarly high miss-rate across metrics, models, and datasets. As a reminder, if the average setup was a good approximation of the risk we should observe miss rates close to 0. Instead, we observe large values of miss rates, ranging from 0.67 to 0.94 for $\phi = AUC$ ($T=0.8$) and 0.62 to 0.93 for $\phi = \text{TPR}$ with, as expected, slightly lower values on

Table 4.2: **Miss rate across different datasets and synthetic data generators** for AUC and TPR .

Dataset	$\mathcal{M}_{target}$	Miss rate		$RMSE$	
		AUC	TPR	AUC	TPR
Adult	Synthpop	0.94	0.92	0.07	0.11
	BayNet	0.73	0.63	0.05	0.11
Census	Synthpop	0.94	0.93	0.14	0.24
	BayNet	0.67	0.75	0.04	0.06

average for BayNet which is known to be less prone to memorization [MGCdM23, AGR23]. Table 4.2 then shows that we obtain similarly high values of RMSE across metrics, models, and datasets.

4.5 MIAs against machine learning classifiers

We have so far compared the specific and average evaluation setups when evaluating MIAs against synthetic data generators. We now compare them using SOTA MIAs against ML classifiers.

Experimental setup. We now consider a ResNet image classifier [HZRS16] trained on CIFAR10 [KH⁺09] and a fully connected neural network classifier trained at predicting the binary salary attribute in Adult as the target models $\mathcal{M}_{target}$. We use LiRA proposed by [CCN⁺22] as the SOTA MIA attack.

For the image classifier, we consider a target dataset D_{target} of size $|D_{target}| = 10,000$ and $|D_{eval}| = 30,000$. We evaluate the effectiveness of the MIAs using AUC and TPR at $FPR = 0.1$ on 256 evaluation models (as done by Carlini et al. [CCN⁺22]).

For the neural network, we take D_{target} of size $|D_{target}| = 2000$ and $|D_{eval}| = 15,222$. We evaluate the effectiveness of the MIAs using AUC and TPR at $FPR = 0.3$ on 500 evaluation models. Note that we here use a FPR of 0.3 as this shows to be a particularly difficult task for the MIA, and TPR rates at FPR 0.1 are very low, and therefore not informative.

Results. Table 4.3 shows that, compared to a miss rate close to 0 we would observe if the average setup was a good approximation of the risk, we obtain similarly high miss-rate values across metrics, models, and datasets. Interestingly, the miss rate for the ResNet image classifier trained on CIFAR10, while still high compared to 0, are amongst the lowest we observe across datasets and models. We hypothesize this to be largely due to CIFAR10 being a much larger dataset ($|D_{target}| = 10,000$) than the other datasets considered ($|D_{target}| = 1000$). While a complete analysis is outside the scope of this work, we believe that the “outlier-nature” of a record might be, on average, more likely to be preserved in across datasets (in the average setup) when the dataset is large. However, other factors such as the effective “dimensionality” of a record are also likely to play a role. Finally, Table 4.3 shows that we obtain similarly high values of RMSE across models, and datasets.

Table 4.3: **Miss rate and RMSE across different datasets** for $\mathcal{M}_{target}$ ML classifier for *AUC* and *TPR*.

Dataset	Miss rate		RMSE	
	<i>AUC</i>	<i>TPR</i>	<i>AUC</i>	<i>TPR</i>
CIFAR10	0.40	0.67	0.07	0.12
Adult	1.0	0.58	0.05	0.07

4.6 Theoretical results

We eliminate what we argue is an incorrect source of randomness to evaluate MIAs against ML models: datasets sampling. Another source of randomness was however necessary to evaluate MIAs (as otherwise the result would be a binary value). We thus propose to use weight initialization as the source of randomness, leading to what we call the specific evaluation setup for MIAs against ML models.

Using a theoretical result we prove below and empirical results across models, datasets, and metrics we argue that weight initialization is a appropriate source of randomness and that R_{avg}^ϕ is, in fact, an average of R_{sp}^ϕ .

To do so, we first introduce and then prove the following theorem:

Theorem 4.1. Let $X = [X_1, \dots, X_n]$ where $\forall i \mid X_i$ is a random variable of mean μ_i and standard deviation σ_i , and let $\bar{X} = [\mu_1, \dots, \mu_n]$. Then we have that the expected Pearson correlation between X and $\bar{X}$ is equal to the square root of the ratio of their variances:

$$E[\rho(X, \bar{X})] = \mathcal{V}(\bar{X}, X)$$

$$\text{where } \mathcal{V}(\bar{X}, X) = \sqrt{\frac{V(\bar{X})}{V(X)}}$$

Proof. Let $X = [X_1, \dots, X_n]$, with $\forall i \mid X_i$ being a random variable of mean μ_i and standard deviation σ_i , and let $\bar{X} = [\mu_1, \dots, \mu_n]$.

Then, since $\rho(X, \bar{X}) = \frac{\text{cov}(X, \bar{X})}{\sqrt{V(X) \cdot V(\bar{X})}}$, by performing a member by member evaluation, we have:

$$\rho(X, \bar{X}) = \frac{\sum_i (X_i - \frac{1}{n} \sum_k X_k) \cdot (\mu_i - \frac{1}{n} \sum_k \mu_k)}{\sqrt{V(X) \cdot V(\bar{X})}}$$

which gives us :

$$E[\rho(X, \bar{X})] = \frac{\sum_i E[(X_i - \frac{1}{n} \sum_k X_k)(\mu_i - \frac{1}{n} \sum_k \mu_k)]}{\sqrt{V(X) \cdot V(\bar{X})}}$$

by linearity. Then, it follows

$$\begin{aligned} E[\rho(X, \bar{X})] &= \frac{\sum_i (E[X_i] - \frac{1}{n} \sum_k E[X_k])(\mu_i - \frac{1}{n} \sum_k \mu_k)}{\sqrt{V(X) \cdot V(\bar{X})}} \\ &= \frac{\sum_i (\mu_i - \frac{1}{n} \sum_k \mu_k)^2}{\sqrt{V(X) \cdot V(\bar{X})}} \\ &= \frac{V(\bar{X})}{\sqrt{V(X) \cdot V(\bar{X})}} \\ E[\rho(X, \bar{X})] &= \sqrt{\frac{V(\bar{X})}{V(X)}} \end{aligned}$$

□

With this theorem, we explain the relationship between two arrays X and $\bar{X}$ of random variables with $\bar{X}_i$ being the mean of variable X_i .

Additionally, we note that

$$\bar{X} = E[X] \Rightarrow E[\bar{X}] = E[E[X]] = E[X]$$

i.e. the two arrays would have equal expected values.

To apply this theoretical results to our context, X would be the array of specific risk estimates, while $\bar{X}$ would be the array of average risk estimates. If our hypothesis is correct, i.e. R_{avg}^ϕ is the average of R_{sp}^ϕ , risk estimates for records in D_{target} would have a relationship consistent with Theorem 4.1. Subsequently, this would also empirically show that the randomness in the specific setup associated with the sampling of the reference record only has a minor effect and is a good choice of randomness.

Empirically, this would mean that $\rho(S, \phi)$, the empirical correlation value, and $\mathcal{V}(R_{avg}^\phi, R_{sp}^\phi)$, the square root of variances value should be close to one another. Similarly, this means that $\bar{R}_{avg}^\phi$ and $\bar{R}_{sp}^\phi$ should be close to one another. To evaluate whether these relationships hold we compute $\rho(S, \phi)$ and $\mathcal{V}(R_{avg}^\phi, R_{sp}^\phi)$ in every settings of our experiments.

Table 4.4 shows how $\rho(S, \phi)$ and $\mathcal{V}(R_{avg}^\phi, R_{sp}^\phi)$ are close to each others across datasets and metrics and for both ML models and synthetic data generators. This empirical relationship between R_{avg}^ϕ and R_{sp}^ϕ is consistent with the hypothesis that the average risk is the mean of the specific risk and provides evidence that weight initialization is an appropriate source of randomness.

Table 4.4: **$\rho(R_{avg}^\phi, R_{sp}^\phi)$ and $\mathcal{V}(R_{avg}^\phi, R_{sp}^\phi)$ values.** Values are presented across different datasets, ML models and synthetic data generators for *AUC* and *TPR*.

	Adult						Census				CIFAR10	
	Synthpop		BayNet		ML classifier		Synthpop		BayNet		ML classifier	
	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR	AUC	TPR
ρ	0.65	0.68	0.85	0.76	0.68	0.63	0.80	0.72	0.85	0.78	0.88	0.82
$\mathcal{V}$	0.63	0.60	0.85	0.73	0.75	0.78	0.75	0.68	0.88	0.72	0.81	0.78

4.7 MIA by a strong adversary

We have so far compared evaluation setups using SOTA attacks from the literature. These assume the adversary to have access to an auxiliary dataset drawn from the same distribution but strictly non overlapping with D_{target} .

We now instantiate an MIA attack by a very strong adversary with knowledge of all the evaluation datasets including the training dataset D_{target} , similar to the standard DP adversary. In the average case, this means an adversary with access to the exact evaluation models –thus datasets sampled from D_{eval} – used for MIA evaluation. In the specific case, this means an adversary with access to the full target dataset D_{target} .

We instantiate the stronger attack in our standard setup (Synthpop with the Adult dataset) and run the attack on 500 different target records randomly sampled from D_{target} . We compare the performance of the strong adversary to the adversary with only access to auxiliary data used previously (Section 4.4.1), which we refer to as the *classic* attack.

Figure 4.4 shows the strong adversary to be able to achieve a substantially higher AUC 0.792 ± 0.092 (mean and standard deviation) than the classic adversary with access to an auxiliary dataset (0.601 ± 0.092) in the specific setup. This is, however, not the case in the average setup where the strong adversary only achieves an AUC of 0.636 ± 0.054 compared to the 0.600 ± 0.060 achieved by the classic adversary.

These results further emphasize the need to use a specific setup when evaluating MIAs against machine learning models, especially as new stronger attacks, leveraging knowledge of the training dataset, are likely to be developed in the future. Indeed, while we here focus on a very strong adversary, akin to the standard DP adversary, new attacks could leverage partial knowledge of the target dataset e.g. information leaked by the released synthetic dataset or knowledge about a specific part of the dataset where the target record would lie.

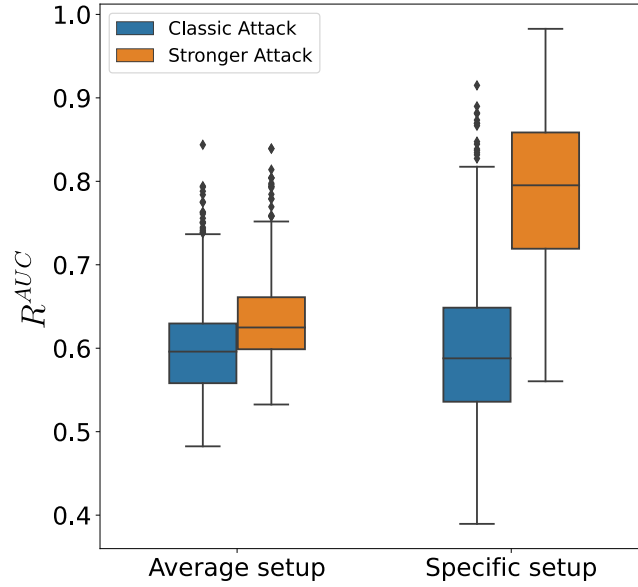


Figure 4.4: AUC results for the classic and stronger attack for 500 points randomly sampled from D_{target} , for both the average and the specific setup.

4.8 Conclusion and limitations

Conclusion. In this chapter, we show that the current source of randomness used to evaluate MIAs against machine learning models is incorrectly averaging the risk across datasets. We instead propose to use a specific setup with weight initialization as the only source of randomness. Using existing state-of-the-art MIAs, we compare the results obtained in the average setup with those obtained in the specific setup and show them to lead to a high number of records to be misclassified as low risk. We show this to be true across more than 10 combinations of datasets, MIA metrics, and ML models including generative and predictive models. We then derive theoretical results which, combined with empirical evidence, strongly suggest that the risk calculated in the current setup is an average of the risks specific to each dataset sampled for testing. We argue these results validate our use of weight initialization as the (only) source of randomness necessary to evaluate MIAs against ML models. We finally instantiate an MIA by a very strong adversary and show the risk posed by this attack to be captured by the specific setup while leading to the same result as the (much weaker) adversary with access to an auxiliary dataset in the average setup.

Taken together, our results strongly emphasize the need to evaluate the effectiveness of MIA

attacks against machine learning model in a specific setup. In particular, we show weight initialisation to be a good source of randomness. In the next chapters, we will now aim to remove the auxiliary data assumption from the shadow modeling technique in different scenarios.

Broader impact. Our work done in this chapter adds to the existing literature on privacy risk evaluation [AZT24, SOT22, CJZ⁺22] and, in particular, enables more accurate record-level risk estimation when releasing machine learning models. We hope this work to help entities dealing with highly sensitive data, such as those in healthcare [LTS⁺20] or the financial sector [Syn24], to better understand the potential data leakage when releasing machine learning models and ensure a high standard of privacy. We believe this work to be particularly impactful for statistical “outliers”, i.e. individuals whose data are significantly different from others, and to help ensure that their privacy is preserved to an equal standard as that of people whose data is closer to the average. Finally, we note that our results might lead to the development of stronger attacks which could be used by malicious adversaries. However, we believe that the knowledge our work provided to data controllers and for further research outweighs the potential risk.

Limitations. The first limitation of our work is the fact that we perform and evaluate two attacks: the query-based attack [HJC⁺22] for synthetic data generators, and LiRA [CCN⁺22] for ML models. Though these two attacks are the current state-of-the-art for their respective models, there might exist attacks that are less sensitive to outliers and therefore to the evaluation setup. Second, we used weight initialization as our only source of randomness for the synthetic data generators. However, for training machine learning models other sources of randomness can be considered, such as batch ordering. Following the approach developed in [CJdMT23], we can explore the impact of these other sources of randomness. Third, we provide intuition about the properties of datasets where using the average instead of the specific evaluation setup might lead to incorrect conclusions. More specifically, we hypothesise that small datasets are particularly vulnerable, but due to the computational cost of training and evaluating MIAs, we were unable to conduct a full study on this topic (per record, each MIA takes several hours to

be conducted, this had to be multiplied by the number of different size considered). Finally, we show how the effectiveness of attacks relying on stronger assumptions, namely some knowledge of the target dataset, might be currently under-evaluated. The attack we instantiate however relies on a much stronger adversary which, while considered by Differential Privacy, might not be realistic in general. It is left for future work to evaluate whether and when stronger but generally realistic attacks might exist.

Chapter 5

Removing the auxiliary assumption for MIAs against synthetic generators

The work presented in this chapter has been published as: Guépin, F., Meeus, M.*, Crețu, A. M., & de Montjoye, Y. A. (2023, September). Synthetic is all you need: removing the auxiliary data assumption for membership inference attacks against synthetic data. In *European Symposium on Research in Computer Security* (pp. 182-198). Cham: Springer Nature Switzerland. https://link.springer.com/chapter/10.1007/978-3-031-54204-6_10*

Statement of contributions: Matthieu Meeus and I are joint first authors on two joint papers, presented in Chapter 3 and 5. We collaborated in the design of the methods, experiments and in the writing of the code and the paper for both. We equally contributed to the production of the figures and writing of the paper for both. Matthieu led the writing of the code and experiments for the paper presented in Chapter 3, while I led the writing of the code and experiments for this paper.

5.1 Introduction

Data is crucial in statistical modeling, machine learning systems, and decision-making processes, driving research and innovation. However, data often pertains directly or indirectly to individuals and may contain sensitive information, such as medical records and financial transactions, raising privacy concerns.

Synthetic tabular data is a promising solution to share data while limiting the risk of re-identification [BDR19]. A synthetic data generator is a statistical model trained on the original, private dataset and used to generate synthetic records. The generated synthetic records would then not be linkable to any specific individual while retaining most of the statistical utility of the original dataset. Extensive research has been dedicated to exploring a wide range of techniques for generating synthetic data [ZCP⁺17, Haz19, XSCIV19b, JYVDS19, Ala22]. Since, if truly anonymous, synthetic data would fall outside the scope of data protection legislation such as the European Union’s General Data Protection Regulation (EU GDPR) [Eur16] or California Consumer Privacy Act [BUK19], various sectors including finance [Fin22], health-care [TWRM20], and research [EYL⁺20] have expressed significant interest in its adoption in practice.

However, synthetic data alone does not necessarily preserve privacy. First, it is long known that aggregation alone does not effectively safeguard privacy [DN03, PTDC17]. Second, achieving formal privacy guarantees for synthetic data generation models poses implementation challenges and currently comes at a cost in utility [SOT22, AGR23].

Membership inference attacks (MIAs) have thus been used to evaluate the privacy preservation capabilities of synthetic data in practice. An MIA aims to infer if a specific target record is part of the generative model’s training set. Recent work has shown that synthetic data is vulnerable to MIAs, with state-of-the-art attacks relying on the shadow modeling approach [SSSS17, SOT22, HJC⁺22]. This approach involves training a membership classifier to distinguish between synthetic datasets generated from so-called shadow datasets with or without a particular target record. Importantly, these attacks require the adversary to have access to an

auxiliary dataset that follows the same distribution as the original, private dataset, from which the adversary will sample their shadow datasets.

We here argue that this is often a strong assumption in practice [SZH⁺18]. While general datasets of images are widely available, medical datasets or datasets of financial transactions – some of the main use cases for synthetic tabular data – are not only not widely available but also very specific e.g. to certain geographies, type of diseases, etc. The practical feasibility of an attack is also an important criterion from a legal perspective when assessing what constitutes anonymous data. Recital 26 of the EU GDPR [Eur16] indeed states that “account should be taken of *all the means reasonably likely to be used*, such as singling out, either by the controller or by another person to identify the natural person directly or indirectly.”

Contribution. In the work presented in this chapter, we show how synthetic data can effectively replace the auxiliary dataset when running MIAs, removing the strong assumption made by attacks so far and making our attack –in our opinion– more reasonably likely from a legal perspective.

First, we consider an adversary with black-box access to the synthetic data generator, which is used to generate shadow datasets for running the MIA. We evaluate the shadow modeling attacks of Houssiau et al. [HJC⁺22] and Meeus et al. [MGCdM23] on two real-world datasets, two synthetic data generators and across ten target records identified by the vulnerable record identification method of Meeus et al. [MGCdM23]. Our results show that MIAs based on synthetic data alone leak the membership of their most vulnerable records 65.5% of the time on average across datasets and generators. This is 15.5 percentage points (p.p.) better than the random guess baseline. We then compare the MIA performance to the traditional setting that assumes access to an auxiliary dataset from the same distribution. We find that our adversary only loses 11.6 p.p. when compared to this much stronger assumption.

Second, we consider an even weaker adversary that exclusively uses the released synthetic data to perform shadow modeling-based MIAs. This adversary obtains an average accuracy of 62.8%. This result is especially meaningful as having access to the released synthetic dataset is an assumption almost always met in practice. Even here, we show the attack to still work

12.8 p.p. better than the random guess baseline.

Third, we identify a potential *double counting* issue which might lower the accuracy of an attack when using synthetic data to replace the auxiliary dataset. We formalize the problem and propose an empirical setup, where we artificially solve the double counting issue, to compute an upper-bound on the accuracy of an attack using only synthetic data. We show the upper-bound to reach 85.8%, 8.7 p.p. higher than the auxiliary data scenario, emphasizing how synthetic only attacks might in the future outperform what is today considered the risk posed by a strong adversary.

MIAs are the main tool to evaluate the privacy-preserving capabilities of synthetic data. However, the strong auxiliary data assumption they currently rely on might lead some to question the practical risk posed by these attacks [SZH⁺18, DCM⁺22] and whether they are ‘reasonably likely’. We here show how this assumption can be relaxed, as adversaries having solely access to the synthetic data generator (For example, in federated learning, the attacker could provide part of the training set to train the resulting model, ultimately ending up with a fully trained black-box generator) or even released synthetic data are still able to develop meaningful attacks. We finally find that future attacks based on synthetic data might outperform traditional attacks if the double counting issue can be resolved.

We make our source code allowing to reproduce the results of our experiments available at: <https://github.com/computationalprivacy/MIA-synthetic>.

5.2 Attack scenarios

We exclusively consider state-of-the-art MIAs, which are based on the shadow modeling technique, we refer to Chapter 2.2 for further details. We assume that the adversary has access to the synthetic dataset $D^s \sim \phi_T(\mathcal{G}(D))$ (as presented in Chapter 2.1.1), where ϕ_T is referred to as the target generator. We will refer to the size of the synthetic dataset as $n_{\text{synthetic}}$. The adversary aims to infer whether a particular record, referred to as the target record x_T , was part of the original dataset, i.e., whether $x_T \in D$ or $x_T \notin D$. In line with the literature, we

consider the standard setting under which the adversary knows the fitting procedure $\mathcal{G}$ used to train the statistical model on the original data.

To model the uncertainty of the adversary about the dataset, we consider four attack scenarios. First, **(S0) Auxiliary** is the traditional setup where the adversary has access to an auxiliary dataset sampled from the same distribution as the private dataset. We then propose two new scenarios assuming a weaker adversary: **(S1) Black-box**, where the adversary has access to the target generator ϕ_T and can query the generator an arbitrary number of times to sample synthetic records and **(S2) Published**, where the adversary has only access to a released synthetic dataset D^s of the same size as the private dataset. Lastly, as a fourth scenario, we construct an artificial setup **(S3) Upper bound** to evaluate the upper bound of MIAs against synthetic data while only using synthetic records.

5.2.1 (S0) Auxiliary

As a baseline, we consider the traditional attack scenario [HJC⁺22, SOT22] where the adversary has access to an auxiliary dataset D_{aux} sampled from the same distribution $\mathcal{D}$ as the private dataset D , i.e. $D_{aux} \sim \mathcal{D}$. D_{aux} is then used to construct the n_{shadow} shadow datasets by uniformly sampling records from D_{aux} without replacement. The meta-classifier is then trained to predict membership with as input features extracted from the synthetic shadow datasets.

Next, the meta-classifier is evaluated on n_{test} synthetic datasets, synthesized from test data that is disjoint from the data used for training. The binary membership prediction is then aggregated across all n_{test} synthetic datasets to a final accuracy used as the MIA performance metric.

5.2.2 (S1) Black box

Next, we remove the auxiliary dataset assumption. We consider an adversary who is able to query the target generator Φ_T for synthetic records, i.e. has black-box access to the generator.

This scenario could, for instance, arise when the end user of the synthetic data would require access to more synthetic records than there were present in the original dataset, e.g. to train ML models. The adversary will here use the black box access to generate m synthetic records to directly construct the shadow datasets.

Note that, unlike the baseline setting (S0) Auxiliary, the shadow datasets and (consequently) the meta-classifier are now **specific** to the target generator on which it is evaluated. In other words, this setup requires the adversary to train $n_{shadow} \times n_{test}$ generators and n_{test} meta-classifiers while in the standard setting (S0), an adversary needs to train $n_{shadow} + n_{test}$ generators and only one meta-classifier.

Again, an adversary will query the trained meta-classifier for one binary prediction for membership per test dataset, which we aggregate to a final accuracy across all n_{test} generators.

For computational reasons (to allow us to not retrain each target generator in the adversary pipeline), we sample $m > |D|$ synthetic records for every target generator Φ_T , which we use to sample the shadow datasets in our experiments.

5.2.3 (S2) Published

In this scenario, we further remove the access to the target generator Φ_T assumption. The only knowledge about the original data available to the adversary is the released synthetic dataset D^s . We here assume that the size of the released synthetic dataset is the same as the original, private set, formally $n_{synthetic} = |D^s| = |D|$.

In this scenario, the adversary trains another generator Φ_S , using the synthetic dataset as training points, i.e., $\Phi_S = \mathcal{G}(D^s)$. With this new generator Φ_S , the adversary generates m new synthetic records to be used to construct the shadow datasets.

We evaluate the MIA performance for this scenario in the same way as in scenario (S1) Black box above.

5.2.4 (S3) Upper bound

When an MIA against synthetic data for a particular target record is successful, the meta-classifier is able to distinguish whether the target record was part of the original dataset or not. In other words, the meta-classifier is able to detect the effect of the presence of the target record in the original dataset on the generated synthetic data. As shown by Meeus et al. [MGCdM23], this effect is more significant for records more distant to their closest neighbours.

In scenarios (S1) and (S2), the adversary uses this synthetic data to construct the shadow datasets. When the target record x_T was part of the target generator’s training dataset, we hypothesize that using these synthetic records to construct the shadow datasets could deteriorate the performance of the meta-classifier in two ways. First, as we use synthetic data that is likely impacted by the presence of x_T already to construct the shadow datasets, the two “worlds” (presence or absence of x_T) in the shadow datasets are likely to be less distinguishable overall by the meta-classifier. Second, this could create a discrepancy in the training (on the shadow datasets) and inference task (on the target generator) of the meta-classifier. We call both effects the *double counting issue* and hypothesize that this could impact the attack performance.

We formalize this issue by first defining the concept of adjoining synthetic datasets to then define the *trace* of x_T .

Definition 5.1. Let $D = (x_1, \dots, x_n)$ be a dataset, then an **adjoining dataset** with respect to x_T will be such that $\exists k \mid D^T = (x_1, \dots, x_k, \mathbf{x}_T, x_{k+2}, \dots, x_n)$ and $x_{k+1} \neq x_T$. We call **adjoining synthetic datasets** the resulting synthetic datasets generated by the same generator model $\mathcal{G}$ trained on the respective datasets. Namely, $D^{s,T} \sim \Phi = \mathcal{G}(D^T)$ and $D^s \sim \Phi = \mathcal{G}(D)$ are called two adjoining synthetic datasets.

Definition 5.2. Let $\mathcal{D}^s$ and $\mathcal{D}^{s,T}$ be two adjoining synthetic datasets with respect to x_T . Then, the **trace** of x_T is defined as the impact of excluding (respectively including) the target record in the training dataset D ($D \cup \{x_T\} = D^T$) of a synthetic data generator $\Phi = \mathcal{G}(D)$ ($\Phi = \mathcal{G}(D^T)$) on the generated synthetic data $D^s \sim \Phi$ ($D^{s,T} \sim \Phi$), written $|\cdot|_\Phi$. Formally, $\text{trace}(x_T) = |\mathcal{D}^s - \mathcal{D}^{s,T}|_\Phi$.

At inference time, the meta-classifier is expected to recognize the trace of x_T i.e. $|\mathcal{D}^s - \mathcal{D}^{s,T}|_\Phi$. When synthetic data is used to construct the shadow datasets and the target record has not been part of the training data for the target generator, the meta-classifier has been trained to recognize this same trace and hence, the adversary does not encounter the double counting issue.

However, when the target generator has seen the target record during training, the adversary uses the synthetic dataset $\mathcal{D}^{s,T}$ to construct shadow datasets, each of which will contain x_T with 50% probability as well. In other words, the synthetic shadow datasets will now be either $\mathcal{D}_2^s \sim \Phi = \mathcal{G}(\mathcal{D}^{s,T} \cup \{x_{random}\})$ or $\mathcal{D}_2^{s,T} \sim \Phi = \mathcal{G}(\mathcal{D}^{s,T} \cup \{x_T\})$ with 50% probability. The meta-classifier is hence trained to recognize the trace of trace of x_T i.e. $|\mathcal{D}_2^s - \mathcal{D}_2^{s,T}|_\Phi$, while at inference time it is still expected to recognize the trace of x_T , i.e. $|\mathcal{D}^s - \mathcal{D}^{s,T}|_\Phi$.

To avoid the double counting issue, we here design a hypothetical attack as a slight modification from scenario (S1). We now artificially ensure that the target x_T is never seen during the training of the generator, to then use the same setup as in (S1). Specifically, when the target is not seen during training, nothing changes, and the adversary has black box access to Φ . In contrast, for a target generator that has seen the target record during training (the target generator will generate $\mathcal{D}^{s,T}$ with D^T as training dataset), we ensure the adversary to have access to an adjoining synthetic dataset $\mathcal{D}^s$, by training the same generator Φ on an adjoining dataset D of D^T .

This scenario serves as an **upper bound** for an MIA with access only to synthetic data, since now we artificially avoid the double counting issue. We further evaluate this scenario in the same way as in scenario (S1).

5.3 Experimental setup

In this section, we describe the experimental setup for the attacks: the synthetic data generation models, datasets, the meta-classifier methods used and the exact attack parameters.

We here used two generators in our experiments, Synthpop and BayNet. Generator details can be found in Chapter 2.1.2.

We performed our experiment over two different datasets, Adult and UK Census. A precise description of those datasets can be found in Appendix B.1.

5.3.1 Meta-classifier methods

We consider two previously proposed methods to extract features from the synthetic shadow datasets and to train the meta-classifier.

Query based. Introduced by Houssiau et al. [HJC⁺22], all details are provided in Chapter 3.4.1. We here used 100,000 randomly sampled count queries of the 2^F possibilities and used a random forest classifier with 100 trees and maximum depth of 10 to predict membership.

Target Attention. Introduced in Chapter 3.4.2, this method takes as input (part of) the synthetic dataset and is the first trainable feature extractor for MIAs against synthetic data. We use the exact implementation and parameters as laid out in chapter 3.4.2.

5.3.2 Parameters of the attack

Table C 3 of Appendix C shows the parameters used throughout our experiments. Here, D_{aux} represents the auxiliary dataset and D_{test} the dataset that is used to sample the test datasets. Both are random and disjoint subsets of the entire dataset. Further, m represents the number of synthetic records queried from the trained generator, n_{shadow} the number of shadow datasets used for training the meta-classifier, and finally n_{test} the number of datasets used for testing.

Throughout our experiments, the size of the released synthetic dataset is equal to the size of the private dataset D , i.e., $n_{synthetic} = |D| = 1000$, and similarly for the shadow datasets, i.e. $|D_{shadow}| = |D_{shadow}^s| = 1000$. In scenarios (S1) and (S3) where $m > n_{synthetic}$, we train the meta-classifier using shadow datasets randomly sampled from the m synthetic records. At

inference time, we use a random subset of $n_{synthetic} = 1000$ synthetic records to query the trained meta-classifier.

When constructing both the n_{shadow} shadow datasets for training and n_{test} datasets for testing, we ensure that the target record x_T is present with 50% probability. This ensures that the evaluation of the attack on the n_{test} datasets is balanced, with a random guess baseline of 50% accuracy for binary prediction of membership.

Lastly, for each dataset, we run the attack on 10 target records selected using the vulnerable record identification method proposed by Meeus et al. [MGCdM23]. For each record in the original dataset, the method computes its vulnerability score as the mean cosine distance, generalized across attribute types, to its five closest neighbours. The records that are the most distant from their closest neighbours, i.e. have the largest mean distance, are selected as vulnerable records.

5.4 Results

We here evaluate how the performance of the MIA varies across our attack scenarios over two synthetic data generators and two datasets.

5.4.1 Query based attack

We first use the state-of-the-art, query based attack method as introduced by Houssiau et al. [HJC⁺22] to compare the MIA performance across different scenarios.

We start by evaluating our weak adversaries (S1) Black box and (S2) Published, where the adversary has only access to the target generator Φ_T or the released synthetic dataset respectively.

Figure 5.1 and Table 5.1 show that, across datasets and generators, an adversary in the (S1) Black box scenario achieves an average accuracy of 65.5%. This is 15.5% better than the

random guess baseline of 50%. This shows that the traditional, strong assumption of having access to an auxiliary dataset can be removed while still successfully inferring membership.

Next, we aim to make the attack as realistic as possible. To achieve that goal, we weaken the assumptions for the adversary to only have access to the published synthetic dataset ((S2) Published). Remarkably, we find that the MIA performance remains fairly constant when compared to the (S1) Black box scenario. Figure 5.1 shows that across datasets and generators, we achieve an average accuracy of 62.8%, which is only 2.7 p.p. lower than the (S1) Black box scenario. These results show that MIAs against synthetic data can still be successful, i.e. 12.8 p.p. better than the random baseline, when the released dataset is the only information available to the adversary. Given that releasing synthetic data instead of the original dataset is often the ultimate goal of generating synthetic data, we argue that scenario (S2) Published represents a minimal assumption that is almost always met in practice. Our results show that even in this realistic case, records detected by the vulnerable record identification method of Chapter 3.4.2 are at risk.

Table 5.1: **MIA accuracy results (mean and standard deviation for 10 target records).** This results are presented across datasets and generators, for the query based attack.

Scenario	UK census		Adult		Average
	Synthpop	BayNet	Synthpop	BayNet	
S0: Auxiliary	$78.6 \pm 3.5 \%$	$78.4 \pm 3.4 \%$	$74.3 \pm 6.2 \%$	$77.0 \pm 8.6 \%$	$77.1 \pm 5.4 \%$
S1: Black-Box	$66.3 \pm 3.0 \%$	$64.6 \pm 5.3 \%$	$64.1 \pm 6.6 \%$	$67.2 \pm 4.4 \%$	$65.6 \pm 4.8 \%$
S2: Published	$61.9 \pm 3.3 \%$	$61.8 \pm 3.3 \%$	$63.1 \pm 4.9 \%$	$64.4 \pm 5.1 \%$	$62.8 \pm 4.2 \%$
S3: Upper Bound	$91.1 \pm 4.0 \%$	$89.3 \pm 5.0 \%$	$80.3 \pm 5.1 \%$	$82.5 \pm 1.1 \%$	$85.8 \pm 3.8 \%$

We then compare the performance of our weak adversary (S1) Black-box to the traditional strong adversary (S0), where we assume access to an auxiliary dataset D_{aux} of real records from the same distribution as the target dataset D . Figure 5.1 shows that our (S1) adversary achieves an accuracy 11.6 p.p. lower compared to the baseline scenario (S0), on average across datasets and generators. This is expected for two possible reasons. First, the synthetic data might not be perfectly representative of the original distribution $\mathcal{D}$. Thus, the training distribution of the meta-classifier from scenario (S1), consisting of features extracted from shadow generators trained on not perfectly representative data, might be quite different from the one on which it is evaluated, leading to worse performance. Scenario (S0) does not suffer from this issue, since

the meta-classifier is trained on features extracted from shadow generators trained on subsets of D_{aux} , which was itself sampled from the underlying distribution $\mathcal{D}$. Second, there is the potential double counting issue, which we investigate next.

Figure 5.1 shows that (S3) Upper Bound achieves an MIA performance of 20.3 p.p. more than (S1) Black Box. These results suggest that the double issue might be significantly affecting the performance of the weaker adversaries and that fixing this issue could, in the future, bridge the gap between our weak adversaries and the (S3) Upper bound scenario.

Lastly, we find that on average, as reported in Table 5.1 across datasets and generators, this adversary achieves an accuracy of 85.8%, which is 8.7 p.p. higher than in the (S0) Auxiliary scenario. This suggests that synthetic data is representative enough to construct shadow datasets for a successful MIA, and potentially more representative than an auxiliary dataset allowing to outperform scenario (S0).

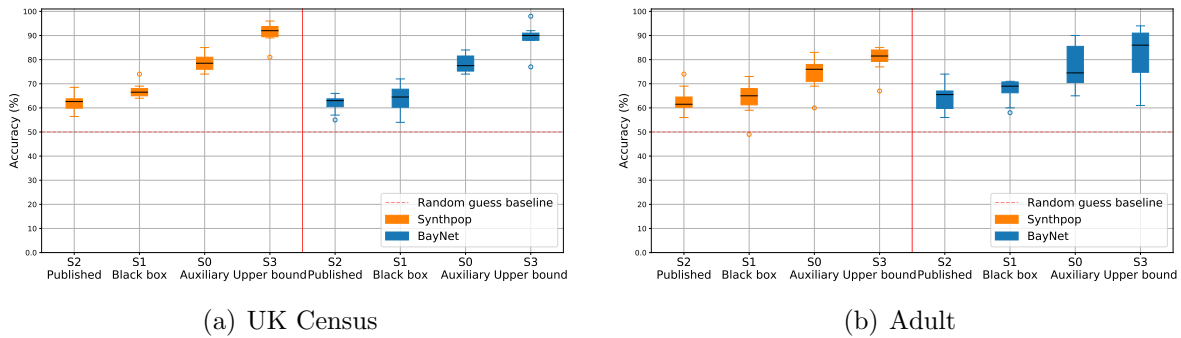


Figure 5.1: **Comparison of MIA accuracy for the query based attack method across 4 different scenarios.** It present results accross S0, S1, S2 and S3, for both generators Synthpop and BayNet. Figure (a) shows results for UK Census, while figure (b) displays results for Adult.

5.4.2 Target attention attack

In this section, we evaluate if our results and conclusion are consistent across attack methods. To do this, we run the target attention attack method as proposed in Chapter 3.4.2 using the same generator and datasets, with the same attack scenarios (S0-3). Figure 5.2 and Table 5.2 summarize the results.

First, we find that the adversary from scenario (S1) is still successful using the target attention

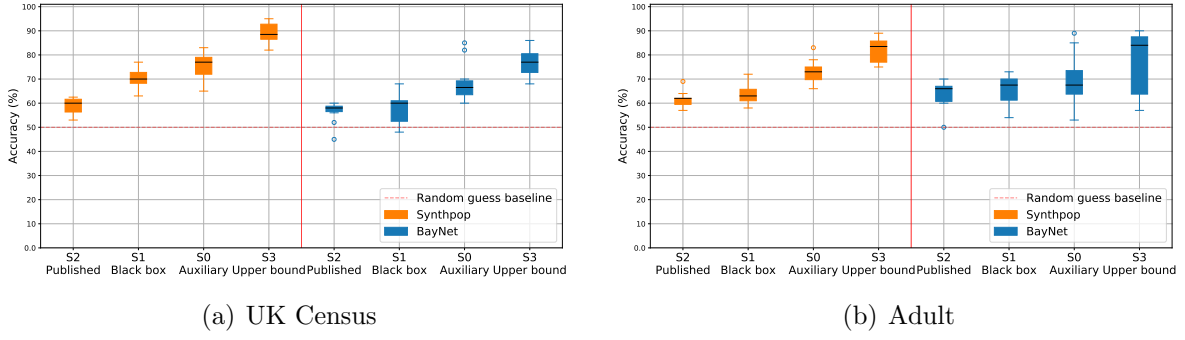


Figure 5.2: **Comparison of MIA accuracy for the target attention attack method across 4 different scenarios.** We consider all scenarios S0, S1, S2 and S3, for both generators Synthpop and BayNet. Figure (a) shows results for UK Census, while figure (b) displays results for Adult.

attack. Across datasets and generators, the average accuracy of the MIA lies at 63.3 %, which is 13.3 p.p. better than the random baseline. This confirms that after removing access to the auxiliary dataset, records remain vulnerable against MIAs, even when using a distinct attack method.

Second, in scenario (S2), the MIA using the target attention method achieves 60.2%, a drop of 3.1 p.p. compared to scenario (S1). These results show that the most realistic scenario, even across attack methods, can be considered as a realistic threat with a performance significantly above the random guess baseline.

Next, we find that the difference between the baseline scenario (S0) Auxiliary and scenario (S1) is on par with the results for the query-based attack. Across datasets and generators, the average accuracy drops by 8.5 p.p. while still achieving an average score of 63.3%.

Finally, in scenario (S3), we confirm our findings that the double counting issue is the main reason affecting the performance of the weaker adversaries, also when using the target attention method. The MIAs achieve an average of 81.2% accuracy, which is 9.4 p.p. higher than (S0) and 17.9 p.p. higher than (S1).

The fact that our findings are consistent across two very distinct attack methods suggests that even when new attack methods are developed, MIAs against synthetic data using only synthetic data will be successful.

Table 5.2: **MIA accuracy results (mean and standard deviation for 10 target records).** This results are presented across datasets and generators, for the target attention attack.

Scenario	UK census		Adult		Average
	Synthpop	BayNet	Synthpop	BayNet	
S0: Auxiliary	75.4 ± 5.4 %	68.7 ± 7.9 %	73.2 ± 4.7 %	69.7 ± 10.3 %	71.8 ± 7.1 %
S1: Black-Box	61.5 ± 3.3 %	62.1 ± 6.3 %	64.1 ± 4.3 %	65.5 ± 6.2 %	63.3 ± 5.0 %
S2: Published	58.9 ± 3.0 %	56.4 ± 4.3 %	61.5 ± 3.3 %	63.8 ± 5.6 %	60.2 ± 4.1 %
S3: Upper Bound	88.9 ± 4.4 %	76.8 ± 5.2 %	82.0 ± 4.9 %	77.2 ± 13.0 %	81.2 ± 6.9 %

5.4.3 Robustness analysis for number of synthetic records m

In scenario (S1) Black Box, we assume the adversary to have black box access to the target generator, i.e. the adversary can query the generator for synthetic records an arbitrary number of times. In our experiments we set the number of synthetic records m to 20,000.

We now evaluate the effect of the value of m on the attack performance. Across the two datasets, for the BayNet generator, Figure 5.3 shows how the MIA performance for scenario (S1) varies for increasing m .

Across datasets, the MIA accuracy remains fairly robust for varying number of synthetic records made available to the adversary. For m varying from 5,000 to 100,000, the mean MIA accuracy does not change significantly. First, this shows that $m = 20,000$, as used in our experimental setup, is a good approximation for black box access to the generator. Further, along with the MIA results for scenario (S2) Published, this confirms that releasing a number of synthetic records m larger or equal to the number of original records, allows the adversary to build meaningful MIAs.

5.5 Future work

5.5.1 Impact of releasing less synthetic records

Intuitively, for a training dataset of fixed size, the more synthetic records we generate, the more information the synthetic dataset might start leaking.

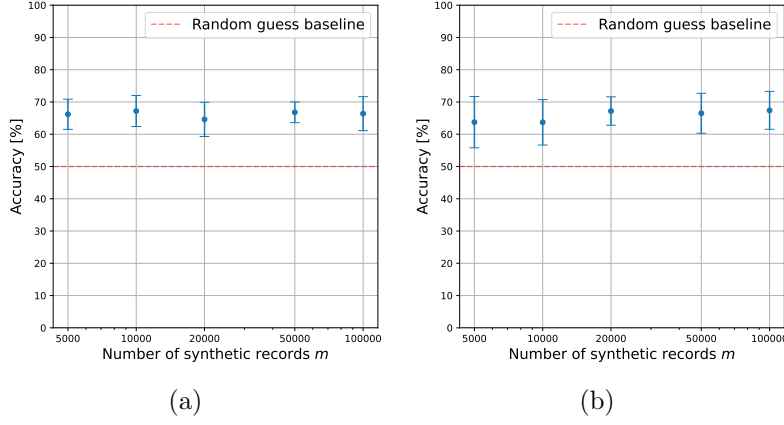


Figure 5.3: **Mean and standard deviation of MIA accuracy in scenarios S1.** This scenario give a black-box access for varying number m , the number of synthetic records available to the adversary. Results for BayNet and the query-based attack using (a) UK Census (b) Adult.

In scenarios (S0) and (S2), the adversary only has access to a limited number of synthetic records $m = |\mathcal{D}|$. As synthetic data is often used to replace the original dataset, we argue that it is reasonable in practice to generate the same amount of synthetic records as the number of training records.

However, we hypothesize that releasing fewer synthetic records for a fixed size of the training dataset, namely $m < |\mathcal{D}|$, could reduce the accuracy of our attack. Releasing less synthetic records typically comes at a cost in utility. We leave the evaluation of this potential trade-off between m and the accuracy of our attack on the released synthetic data for future work.

5.5.2 Differentially private synthetic generation methods

As main contribution in this work, we show that it is possible to attack a synthetic data generator based only on the generated synthetic data.

We leave for future work to confirm whether these effects translate to synthetic data generators with formal privacy guarantees, such as differentially private generators proposed in the literature [ZCP⁺17, JYVDS19]. Previous work has shown that Differential Privacy (DP) comes at a cost in utility [SOT22, AGR23] and that the MIA accuracy drops for decreasing value of the privacy budget ϵ [MGCdM23]. We expect that, while exhibiting similar trends, our findings would translate to DP generators.

5.5.3 Bridging the gap with the upper bound

Our results show that scenario (S3) achieves a significant MIA accuracy, namely higher than scenarios (S1) and (S2), and even higher than scenario (S0). We leave for future work to address the double counting issue we identified in practice, to bridge the gap between scenarios (S1, S2) and the upper bound scenario (S3).

Potentially, an adversary could remove the synthetic records close to the target record, prior to using the synthetic data to construct the shadow models. This could reduce the impact of the double counting issue, but might also introduce bias into the shadow datasets. Additionally, note that in scenario (S1) we currently train the meta classifier using shadow datasets randomly sampled from $m = 20000$ synthetic records, to then infer a prediction on a random subset of $n_{\text{synthetic}} = 1000$ synthetic records. An adversary could for instance infer the prediction on multiple subsets of the m synthetic records to potentially make a more optimal, ensemble prediction.

5.6 Conclusion

Sharing data plays a pivotal role in research and innovation. Increasingly, synthetic data has been proposed to share privacy-preserving tabular data, by synthesizing records that are not directly linkable to real records, while retaining data utility.

Membership Inference Attacks (MIAs) are the standard to audit the privacy preservation of synthetic data, and recent work has shown that these attacks can successfully infer the membership of certain records in the private dataset. State-of-the-art MIAs rely on shadow modeling, which traditionally assumes an adversary to have access to an auxiliary dataset.

First, this auxiliary data assumption is hard to meet in practice. Second, GDPR Recital 26 [Eur16] states that, to legally meet anonymization standards, all means reasonably likely for an adversary to possess should be considered.

We here proposed a more realistic attack by removing the auxiliary data assumption. Across two real world datasets and two synthetic data generators, we find that MIAs are still successful when only using synthetic data.

Specifically, we find that on average, an adversary with black box access to the generator achieves 65.5% accuracy, while an adversary with only access to the released synthetic dataset attains an accuracy of 62.8%. The latter result is particularly significant as it demonstrates that an adversary can extract sensitive information from released synthetic data without any additional information.

Moreover, we identify a double counting issue and establish an upper bound for MIA accuracy against synthetic data when only synthetic data is available. Using current state-of-the-art attacks, this upper bound stands at 85.8%, which is, remarkably, higher than traditional attacks using auxiliary data. This finding highlights the potential for future researchers to bridge the existing gap of MIA performance between realistic scenarios and the upper bound.

Our results provide compelling evidence that MIAs against synthetic data pose a realistic threat in practice. We hope this helps researchers and practitioners to better evaluate risks associated with releasing synthetic data, while encouraging the development of methods to address these concerns.

In the next chapter, we will show that, while removing some information from the data released (the release of individual records, even if only synthetic, but keeping the global statistics available), it is still possible to remove the auxiliary data assumption from the shadow modeling technique.

Chapter 6

A Zero Auxiliary Knowledge Membership Inference Attack on Aggregate Location Data.

*The work presented in this chapter has been published as: Guan, V. *, Guépin, F. *, Cretu, A. M., & de Montjoye, Y. A. (2024). A Zero Auxiliary Knowledge Membership Inference Attack on Aggregate Location Data. Proceedings on Privacy Enhancing Technologies. <https://doi.org/10.56553/popets-2024-0108>*

Statement of contributions: Vince and I are joint first author on this paper. We collaborated in the design of the methods, experiments, writing of the code and the paper. We equally contributed to the writing of the code, running experiments on the CDR dataset, and production of the figures. I led experiments on the Milano dataset while Vince led writing the paper.

6.1 Introduction

Studies of movement and density patterns are at the core of many important domains, such as epidemiology [HY21, GMC⁺20], humanitarian response [YJR⁺22], and finance [HBR13]. The

collection of human mobility and location data to that end raises question to people, as they see their personal mobility information being released [VZ16, Zho17, Hop21], in particular since this data carry information regarding a person’s social, economic, and political life [GdBK19].

These concerns are grounded on the high vulnerability of individual-level location data to re-identification attacks, due to their sparsity, unicity and temporal regularity [ZB11, dMHVB13, TdM22]. Trying to prevent those attacks, data practitioners commonly release location data by using an aggregation technique beforehand [ABC⁺20, PBBL11, XSZ⁺15]. For example, the Public Health Agency of Canada studied citizens’ movement during the COVID-19 pandemic, using aggregate location data from millions of mobile devices, provided by TELUS [Off23, Oli21]. British researchers conducted similar COVID-19 mobility analysis [JWA⁺20, TC23], using aggregate location data obtained from O2 and Facebook. Once aggregated, the level of individual protection is deemed sufficient [Off23], and it is then sold to interested parties [Sav21, BK23]. As an example, one of those parties was the U.S. government, which used commercial aggregate location data for law enforcement purposes and military intelligence [Sav21]. But that’s only one sector, aggregate location data can also be used in urban design, to optimize public transit networks [ML21, Kak22, O219], or in finance to understand consumer behavior [Saf24, Pre24].

Motivation. As outlined in the E.U. Article 29 Working Party’s guidance on anonymization techniques, aggregation reduces the risk of re-identification but does not eliminate all privacy risks [Com14]. In particular, aggregates may still be vulnerable to membership inference attacks (MIAs), whose goal is to infer if an individual’s data was included in the data release, e.g. aggregate data. MIAs have become the de facto standard in privacy auditing due to their practical threat model and theoretical properties. From a practical perspective, a successful MIA is a direct privacy violation whenever participation in the data release is sensitive [LQS⁺13]. Furthermore, MIAs can be used as building blocks for other attacks, by first inferring a user’s participation and then inferring their sensitive attributes. From a theoretical perspective, the success rate of an MIA is upper bounded following the application of differential privacy (DP) [DMNS06, YGFJ18, HOT⁺23]. Hence, MIAs can be used as an auditing tool for DP implementations [JUO20, NST⁺21]. Today, MIAs are widely used to assess the privacy

risk of a broad range of data releases, including aggregate genetic data [HSR⁺08, SOJH09], aggregate survey data [BB20], aggregate location data [PTDC17, OJGdM19], machine learning models [SSSS17, JE19, NST⁺21] and synthetic data releases [SOT22, HJC⁺22, MGCdM23, GMCdM23].

MIAs pose an especially strong privacy threat on aggregate location data, since location data is often processed alongside sensitive attributes, such as socioeconomic status [TC23] and vaccination status [Hop21]. In a notable example, a high-ranking priest resigned after being outed as homosexual by a radical group that matched his smartphone data with location data from Grindr, a popular dating app among the LGBTQ+ community [BK23]. It is therefore important to understand the practical risk that MIAs pose on aggregate location data, particularly by a realistic adversary, who only possesses information about their target.

The first and most prominent MIA on aggregate location data was proposed by Pyrgelis et al. [PTDC17]. Their “Knock-Knock” (KK) MIA works by training a binary classifier on a set of aggregates, wherein the adversary includes the target trace half of the time, and labels the aggregates accordingly. However, in addition to knowing the target trace, KK MIA requires the adversary to have access to a large auxiliary dataset of **individual-level traces** over the **same locations** and from a **similar population** as in the aggregate release. This is, when it comes to location data, a very strong assumption. This reliance on a strong adversary has led companies and practitioners to dismiss the risk posed by MIAs on location data. To the best of our knowledge, all previous works studying MIAs on aggregate location data require a similar auxiliary dataset [PTDC20, ZZZ20, OJGdM19].

Contributions. To assess the realistic privacy risk of releasing aggregate location data, we introduce the Zero Auxiliary Knowledge (ZK) MIA. ZK MIA is the first MIA on aggregate location data that does not require the adversary to have access to an auxiliary dataset. To remove this strong assumption, we develop a novel synthetic data-based approach, in which the adversary generates a reference dataset of synthetic traces, using only statistical parameters estimated from the aggregate. Training aggregates are then created using the synthetic reference (see Figure 6.1 for details). To account for privacy mechanisms applied to the release, we develop

techniques to correct the parameter estimation for bias and noise, which enables ZK MIA to effectively attack privacy-aware aggregates as well. We also demonstrate that a paired sampling technique further improves MIA performance by isolating the contribution of the target trace within the high-dimensional aggregate. In the setting of ϵ -DP aggregate location data, we show that paired sampling enables MIAs to approach the worst-case ϵ -DP bound, offering a significant increase in performance to previous implementations.

We evaluate our Zero Auxiliary Knowledge (ZK) MIA against the state-of-the-art Knock-Knock (KK) MIA from Pyrgelis et al. [PTDC17] using two location datasets: i) a large-scale call detail record (CDR) dataset, and ii) the Milan Twitter dataset [SdM15] from the Telecom Italia Big Data Challenge [BDNL⁺15]. We apply the MIAs on raw and privacy-aware aggregates computed over 1000 users. Our results show that our ZK MIA closely matches the performance of KK MIA, without depending on extensive prior knowledge. On raw aggregates, both MIAs achieve 0.99 AUC on both datasets, suggesting that aggregation in itself is an ineffective safeguard. Both MIAs also surpass 0.9 AUC on both datasets under common privacy settings, including $\epsilon = 1$ event-level DP noise addition.

We further relax assumptions and show that the adversary does not need the full target trace for ZK MIA to succeed. Indeed, ZK MIA still achieved 0.84 AUC on the CDR dataset, with $\epsilon = 1$ event-level DP in place, when the adversary only knew a random 10% of the target trace.

After extensive evaluations across different privacy mechanisms, namely the suppression of small counts [CKL⁺09] and ϵ -DP noise addition [DMNS06], we argue that the commonly used ϵ -DP implementations on aggregate location data [Des21] do not protect against realistic privacy threats, such as our ZK MIA. We conclude that the only effective mitigation is the application of strong user level DP or user-day level DP guarantees, which is not yet a common practice [Tel24, MDMZ⁺23, O219, Saf24, Pre24].

We make our source code allowing to reproduce the results of our experiments available at: https://github.com/computationalprivacy/ZK_MIA.

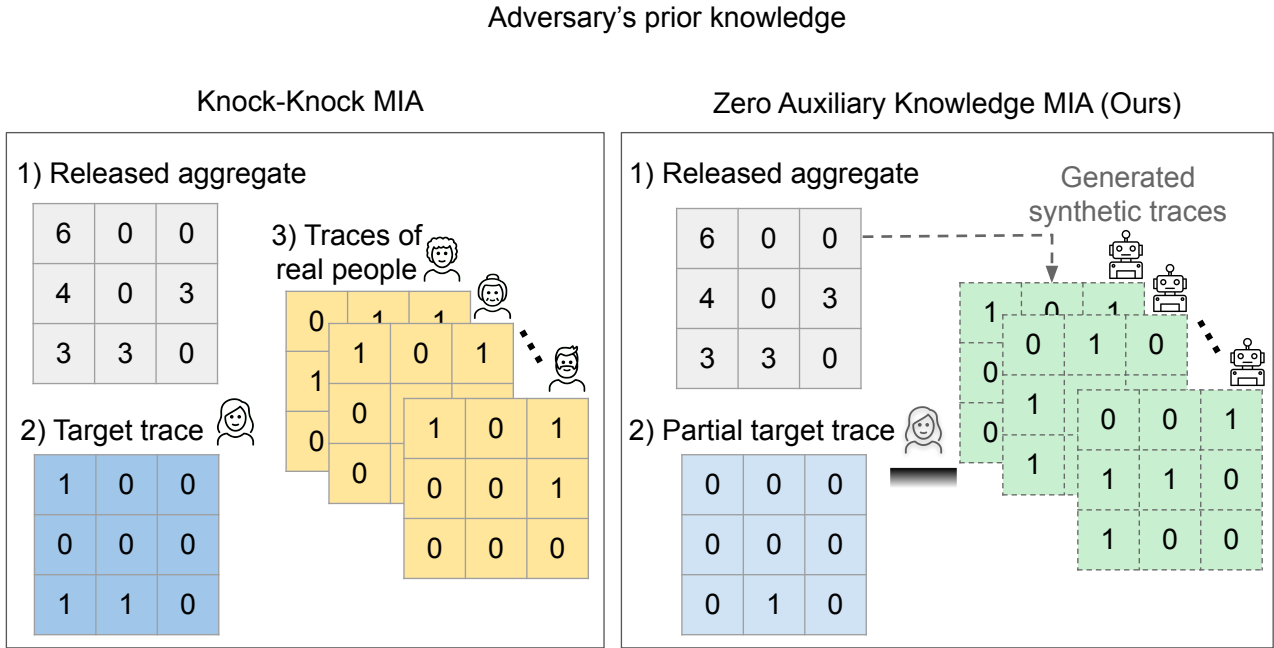


Figure 6.1: **Adversary's prior knowledge presented for both scenario of Chapter 6.** This figure present adversary's prior knowledge in the previous work, Knock-Knock MIA [PTDC17] (left), and our work, Zero Auxiliary Knowledge MIA (right). The ZK adversary does not require knowledge of location traces of real people to run the MIA.

6.2 Definitions and threat model

We formally define location traces and aggregates in Section 6.2.1 and overview aggregate-level privacy measures in Section 6.2.2. In Section 6.2.3 and 6.2.4, we outline the membership inference problem on aggregate location data and introduce the concept of a membership classifier. We present the threat model for our Zero Auxiliary Knowledge MIA and compare it against previous threat models for MIAs on location aggregates in Section 6.2.5. Table C 4 in Appendix C contains a glossary of common terms used throughout the chapter.

6.2.1 Location Traces and Aggregates

Let $\mathcal{S} = \{s_1, \dots, s_{|\mathcal{S}|}\}$ represent the set of all regions of interest (ROIs) where location data is collected. Similarly, $\mathcal{T} = \{t_1, \dots, t_{|\mathcal{T}|}\}$ denotes the set of time intervals, also known as epochs, during which data collection occurs. In this chapter, we assume that the geographic positions

(i.e. approximate longitude and latitude) of the ROIs are known. For example, $\mathcal{S}$ may represent a set of square regions that partition a city into a grid, and $\mathcal{T}$ may represent contiguous hours over one month.

We focus on the scenario where location data of a set of users Ω is collected over the ROIs $\mathcal{S}$ and the epochs $\mathcal{T}$. We define the *location trace* L^u of a user $u \in \Omega$ as the set of geo-tagged and time-stamped visits (s, t) that u made within $\mathcal{S}$ during $\mathcal{T}$. We formally represent a user's location trace as the binary matrix

$$L_{s,t}^u = \begin{cases} 1 & \text{if user } u \text{ visited ROI } s \text{ during epoch } t \\ 0 & \text{otherwise.} \end{cases} \quad (6.1)$$

Let $\mathcal{U} \subset \Omega$ be a group of m users whose location data is aggregated. We define an *aggregate* $A^{\mathcal{U}}$ to be the aggregate count statistics for $\mathcal{U}$ over $\mathcal{S} \times \mathcal{T}$. Formally, this is defined by the sum

$$A^{\mathcal{U}} = \sum_{u \in \mathcal{U}} L^u. \quad (6.2)$$

The entry $A_{s,t}^{\mathcal{U}}$ therefore corresponds to the number of users in $\mathcal{U}$ who visited ROI s during epoch t .

6.2.2 Privacy measures on location aggregates

The data collector may be wary of the privacy risks of releasing the raw aggregate $A^{\mathcal{U}}$, and therefore apply privacy measures before releasing it. Details on Differential Privacy (DP) [DMNS06], considered as the gold standard for releasing information while protecting the privacy of individuals with formal guarantees, can be found in Chapter 2.3.

Differential Privacy applied to location aggregates

When we try to apply the DP definition to location dataset, we need to be careful when we define the privacy unit required to define the notion of neighbouring datasets. The most common definitions, in increasing level of privacy protection, are event-level, user-day level, and user-level DP (see Desfontaines [Des21] for an overview). The privacy unit for event-level DP is an individual data entry by any given user. For aggregate location data, this would be a single visit by a user to a ROI s during an epoch t . The privacy unit for user-day level would be all visits registered by any given user over a day. Finally, for user-level, the unit would be all visits in any given user’s trace.

Randomised ε -DP mechanisms can be designed by adding noise sampled from the Laplace distribution [DMNS06]. $A_{DP}^{\mathcal{U}}(\varepsilon) = A^{\mathcal{U}} + \text{Lap}(\frac{\Delta}{\varepsilon})$ would satisfy Equation 2.6 and be an ε -DP aggregation mechanism, where Δ is the global sensitivity, determined by the privacy unit.

In this Chapter, we assume the common practice of post-processing to ensure legitimate aggregate counts [GF16, PTDC17, PTDC20, ZFVH22]. Negative counts are set to 0, counts exceeding the group size m are set to m , and counts are rounded down to the nearest integer. These transformations will preserve ε -DP due to the post-processing theorem [DKM19]. We note that the adversary can always apply these transformations themselves if the data collector does not do so already.

Suppression of Small Counts (SSC)

SSC is a privacy mechanism that aims to protect user privacy by hiding rare values. It has been frequently used across different types of datasets [CHCdM22, GHR⁺19, PTDC20], including mobility datasets [KAB23, ABC⁺20]. Instead of releasing the raw aggregate $A^{\mathcal{U}}$, the data collector may choose a threshold $k \in \mathbb{N} \cup \{0\}$, and release the suppressed aggregate

$$A_{SSC}^{\mathcal{U}}(k)_{s,t} = \begin{cases} A_{s,t}^{\mathcal{U}} & \text{if } A_{s,t}^{\mathcal{U}} > k \\ 0 & \text{if } A_{s,t}^{\mathcal{U}} \leq k. \end{cases} \quad (6.3)$$

$A_{SSC}^{\mathcal{U}}(k)$ therefore contains the true count of users who visited a ROI s during epoch t as long as the count exceeds k . Lesser visited pairs (s, t) that record k or less visits are reported as 0 instead. Suppression can also be applied following ε -DP noise addition, such that (6.3) is applied on a noisy aggregate $A_{DP}^{\mathcal{U}}(\varepsilon)$. This produces $A_{DP,SSC}^{\mathcal{U}}(\varepsilon, k)$, an ε -DP aggregate whose final counts have been suppressed with threshold k . This transformation would preserve ε -DP due to post-processing [DKM19], and may add a layer of complexity that mitigates attacks in practice.

Figure 6.2 provides an example illustrating the application of the various privacy measures discussed here.

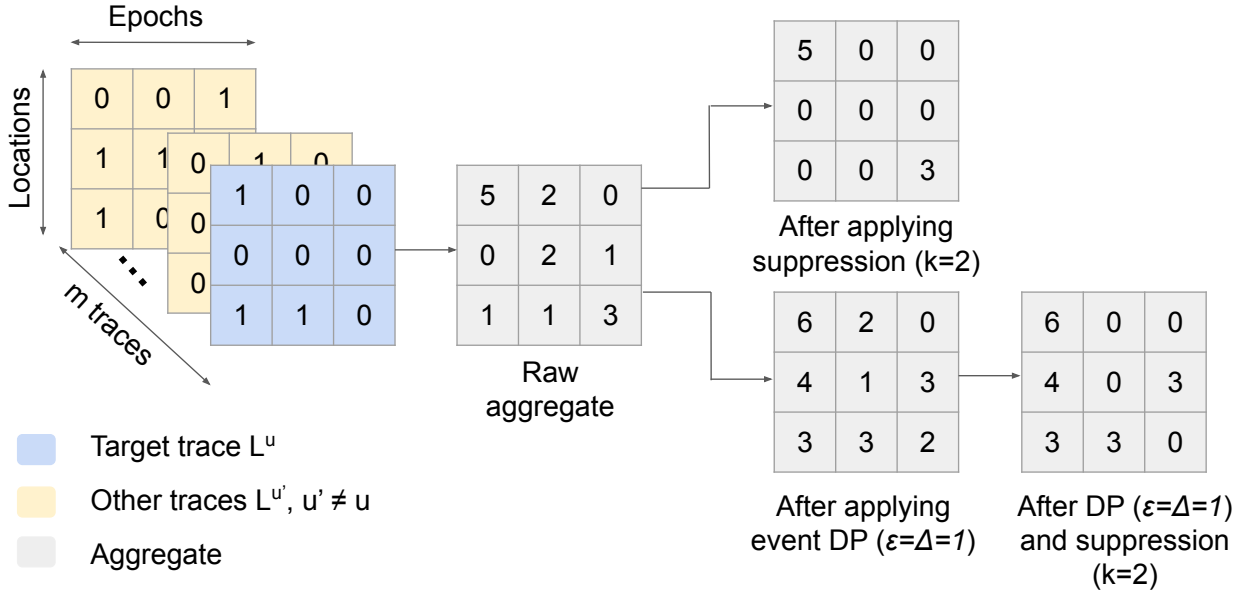


Figure 6.2: **Application of aggregation for location data with various protections.** This is an example of how suppression of small counts and differential privacy may be applied to an aggregate with 3 ROIs (rows) and 3 epochs (columns).

6.2.3 Problem formulation

To complete the definition of a MIA given in Chapter 2.2.2, we here present a definition in the context of aggregate count statistics. We assume that the data collector releases aggregate count statistics $\bar{A}^{\mathcal{U}}$ over the ROIs $\mathcal{S}$ and the epochs $\mathcal{T}$, for the m users in the group $\mathcal{U}$. There are various cases depending on the privacy measures applied prior to release:

$$\bar{A}^{\mathcal{U}} = \begin{cases} A^{\mathcal{U}}, & \text{if the raw aggregate counts are released,} \\ A_{DP}^{\mathcal{U}}(\varepsilon), & \text{if only } \varepsilon\text{-DP is applied,} \\ A_{SSC}^{\mathcal{U}}(k), & \text{if only threshold } k \text{ SSC is applied,} \\ A_{DP,SSC}^{\mathcal{U}}(\varepsilon, k), & \text{if threshold } k \text{ SSC is applied after } \varepsilon\text{-DP.} \end{cases}$$

The goal of an adversary Adv performing an MIA on $\bar{A}^{\mathcal{U}}$ is to determine whether their target u^* contributed to $\bar{A}^{\mathcal{U}}$, inferring IN for $u^* \in \mathcal{U}$ and OUT for $u^* \notin \mathcal{U}$.

6.2.4 Membership classifier

Given an aggregate release $\bar{A}^{\mathcal{U}}$ over m users, an adversary infers membership of the target u^* within the aggregation group $\mathcal{U}$ by using a binary membership classifier. Classifiers are commonly instantiated as machine learning models [PTDC17, PTDC20, ZZZ20], but statistical models, like the log-likelihood function, have been applied as well [HSR⁺08, BB20]. To train the classifier, Adv typically creates a balanced set of labeled size m training aggregates [PTDC17, ZZZ20, PTDC20, OJGdM19]. Half of the aggregates include the target trace L^{u^*} and are labeled IN , and the other half are labeled OUT . Training the classifier will create a decision boundary in the underlying space of aggregate releases [BN06]. In the case of aggregate location data over ROIs $\mathcal{S}$ and epochs $\mathcal{T}$, the decision boundary is characterized by a hypersurface that partitions the matrix space $\mathbb{R}^{|\mathcal{S}| \times |\mathcal{T}|}$ into two sets.

6.2.5 Threat model

In this section, we present our Zero Auxiliary Knowledge (ZK) MIA threat model. It is commonly assumed in MIAs across various domains that the adversary has access to an auxiliary dataset and complete knowledge of the target record [PTDC17, NSH19, SZH⁺18, TLG⁺19, YGFJ18, SSSS17]. Our ZK threat model relaxes both assumptions by eliminating the need for an auxiliary dataset and allowing for only partial knowledge of the target trace.

For context, we also describe threat models of previous MIAs on aggregate location data. All threat models consider an adversary Adv , whose goal is to determine whether a specific target user u^* is included in the released aggregate $\bar{A}^{\mathcal{U}}$. The aggregate is computed across m users over ROIs $\mathcal{S}$ and epochs $\mathcal{T}$. We assume that the locations of the ROIs $\mathcal{S}$ are known.

Knock-Knock [PTDC17]: The adversary has an auxiliary dataset $Ref = \{L(u_1), \dots, L(u_{|Ref|})\}$ of user traces, over the same locations and a similar population as the released aggregate. Ref has at least m traces, including the full target trace L^{u^*} .

LocMIA [ZZZ20]: The adversary knows u^* 's social network and has an auxiliary dataset $Ref = \{L(u_1), \dots, L(u_{|Ref|})\}$ of user traces, over the same locations and a similar population as the released aggregate $\bar{A}^{\mathcal{U}}$. Ref has at least m traces, including the traces of u^* 's friends, but not L^{u^*} .

Zero Auxiliary Knowledge (ours): The adversary knows a subset of the target u^* 's visits. Equivalently, the adversary knows a partial target trace $\tilde{L}^{u^*}$, such that $supp(\tilde{L}^{u^*}) \subset supp(L^{u^*})$.

KK MIA and LocMIA are reliant on the adversary's access to an extensive auxiliary dataset Ref . In particular, Adv samples individual traces from Ref to create training aggregates. These traces are also assumed to range over the same locations, and belong to a similar population as the traces aggregated in the release $\bar{A}^{\mathcal{U}}$, in order to properly train the membership classifier (Section 6.2.4). However, individual traces are known to be sensitive [dMHVB13], and are unlikely to be made available, particularly when the data is aggregated as a privacy measure. Furthermore, because Ref must contain at least m traces, this assumption is impractical for even moderately sized aggregates. Although LocMIA removes prior knowledge about the

target trace L^{u^*} , it must assume knowledge of u^* 's friends' traces to create a suitable proxy. More importantly, LocMIA still requires a large auxiliary dataset of individual traces.

In contrast, our Zero Auxiliary Knowledge adversary only requires that the adversary has knowledge about the target's location history. We emphasize that the adversary does not need to know the full trace L^{u^*} . Our threat model encompasses the case where only a few of the target's visits are known to the adversary. For example, the adversary may infer some of u^* 's visits from social media activity or direct observation. Figure 6.3 provides more details on the architecture of our attack.

6.3 Related work

MIAs on aggregate location data. MIAs on aggregate location data have been shown to be successful on multiple location datasets [PTDC17, PTDC20, ZZZ20], using a binary classifier to perform the inference task. The performance of the MIAs on small aggregates (< 500) is especially well-studied, as the influence of the target is easier to distinguish [PTDC17]. For example, the KK MIA by Pyrgelis et al. [PTDC17], achieved $AUC > 0.83$ when attacking size 100 aggregates across two different mobility datasets. LocMIA [ZZZ20] is another MIA on aggregate location data, which removes prior knowledge about the target trace. Instead, LocMIA assumes access to social network information, and the traces of the target's friends, in order to construct a proxy for the target's real trace. However, both KK MIA and LocMIA crucially require the adversary to have access to a large auxiliary dataset to train the binary classifier. In contrast, our ZK MIA does not require any auxiliary dataset, and only requires partial information about the target trace (e.g. A random 10% proportion of their visits). ZK MIA therefore addresses the research gap of the MIA risk posed by a less knowledgeable adversary. The distinctions in prior knowledge are discussed in depth in Section 6.2.5. Our ZK MIA also features a novel approach, being the first MIA on aggregate location data to use synthetic trace generation.

Generation of synthetic location traces. There are many techniques for generating

synthetic location traces that capture realistic human mobility patterns [KG17, KTVG18, OSRY18, KLBV07, JZGR16, LHK⁺09]. However, since our ZK MIA requires generating suitable synthetic traces without using additional information, this heavily limits the scope of applicable techniques. RNNs, GANs, and copulas have been used to generate synthetic traces that collectively approximate a real mobility dataset [KG17, KTVG18, OSRY18]. However, these techniques require real traces to train the model, which the ZK adversary does not have. Many of the state-of-the-art mobility models are also unsuitable because they simulate small-scale continuous trajectories (e.g. walks on campus) [KLBV07, LHK⁺09]. In contrast, location aggregates typically comprise discrete traces over a metropolitan region. We therefore identified a probabilistic unicity model by Farzanehfar et al. [FHdM21], which requires only four statistical parameters to guide the synthetic generation. We demonstrate that we can non-trivially adapt this model for the ZK MIA. In particular, we develop methods to precisely estimate these parameters from the aggregates in order to produce realistic synthetic location aggregates.

MIAs with reduced auxiliary data. Previous attempts have been made [SSSS17, SZH⁺18, TLG⁺19, YGFJ18, CGdM24, GMCdM23] to relax the standard assumption of an adversary’s access to an auxiliary dataset that has high statistical similarity with the attacked dataset, e.g. sampled from the same distribution [SSSS17, NSH19]. In the setting of machine learning (ML) models, where an MIA infers whether a record was a part of the ML model’s training set, Shokri et al. [SSSS17] proposed an MIA without auxiliary data. Instead, they use synthetic data, which they generate using the ML model’s confidence scores. Similarly, Salem et al. [SZH⁺18] trained an MIA using unrelated data, e.g., training on text data to attack an image model. These approaches require access to the ML model, and they train on features that are specific to ML models, such as the top- k confidence scores, which may be shared across ML models pertaining to different types of data. In contrast, the features used to train an MIA on aggregate location data explicitly depend on the specific regions, times, and population over which the aggregates are computed. In the context of synthetic generators, Guépin et al. [GMCdM23] performed an MIA against synthetic data using the released synthetic dataset as the auxiliary dataset. However, in the case of aggregate location data, the release cannot be directly used as a reference dataset, since it does not contain individual records.

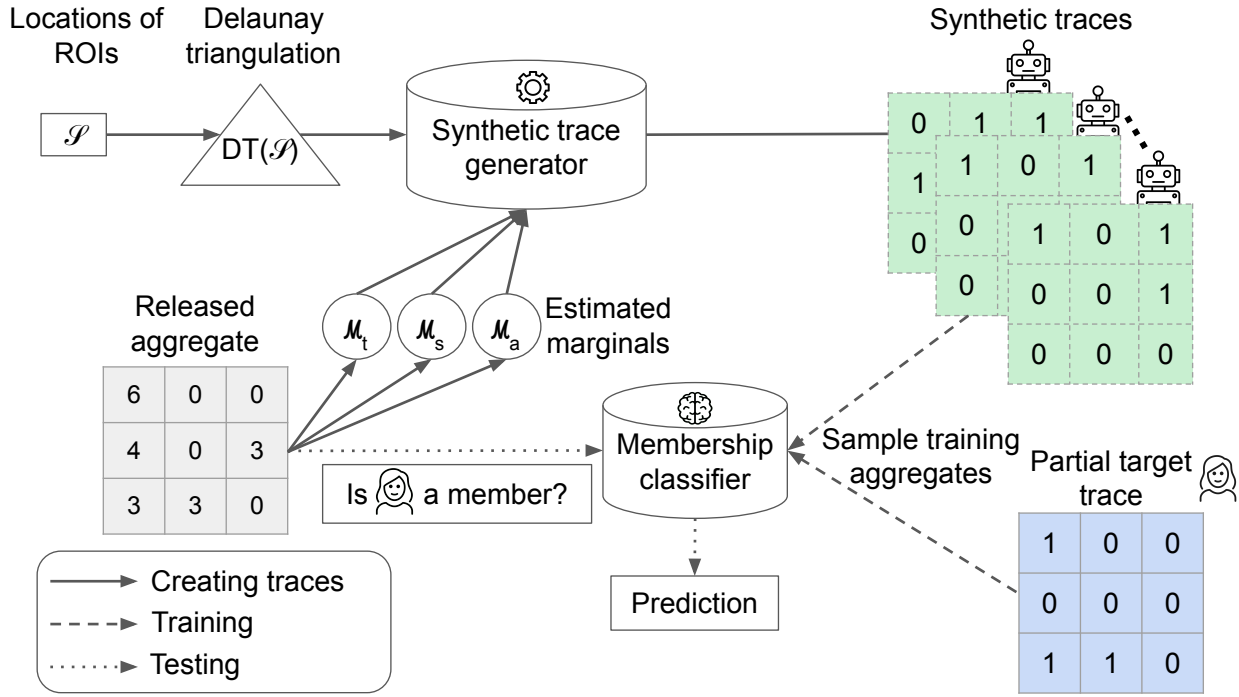


Figure 6.3: **Architecture of our Zero Auxiliary Knowledge MIA attack.** *Adv* first creates synthetic traces, then uses them with the partial target trace to train the membership classifier, before predicting membership.

6.4 Methodology

We dedicate Sections 6.4.1-6.4.3 to explaining the synthetic-based methodology of our Zero Auxiliary Knowledge MIA. Section 6.4.4 explains the paired sampling mechanism on training aggregates, which boosts the performance of ZK MIA and KK MIA, as shown in Section 6.6.4.

6.4.1 Zero auxiliary knowledge MIA framework

We implement the Zero Auxiliary Knowledge MIA as a binary classifier. However, whereas *Adv* uses their auxiliary dataset as a reference for creating training aggregates for KK MIA and LocMIA, *Adv* instead uses the reference of synthetic traces that they created from the released aggregate. Furthermore, if the full target trace L^{u^*} is not known, *Adv* may instead use a partial trace when creating the *IN* training aggregates. Figure 6.3 illustrates ZK MIA's overall attack architecture.

6.4.2 Generating synthetic traces from aggregate location data

In order to generate synthetic traces for our Zero Auxiliary Knowledge MIA, we adapt a probabilistic mobility model [FHdM21]. Farzanehfar et al.[FHdM21] developed this model to reproduce unicity patterns in large populations. The model requires four statistical parameters, described below and illustrated in Figure 6.4. Recall that for a discrete random variable X taking values in $x_1, \dots, x_N$, its probability mass function (p.m.f.) $\mathcal{P} : \{x_1, \dots, x_N\} \rightarrow [0, 1]$ maps each possible value to its corresponding probability.

1. The marginal space distribution $\mathcal{P}_S : \mathcal{S} \rightarrow [0, 1]$ is a p.m.f. such that $\mathcal{P}_S(s)$ is proportional to the number of visits to ROI s by users in Ω across all epochs in $\mathcal{T}$.
2. The marginal time distribution $\mathcal{P}_T : \mathcal{T} \rightarrow [0, 1]$ is a p.m.f. such that $\mathcal{P}_T(t)$ is proportional to the number of visits during epoch t by users in Ω across all ROIs in $\mathcal{S}$.
3. The marginal activity distribution $\mathcal{P}_A$ models the total number of visits recorded within $\mathcal{S}$ during $\mathcal{T}$ by a user drawn from Ω .
4. The Delaunay triangulation, denoted $DT(\mathcal{S})$, is a triangulation with vertices corresponding to the set of positions (longitude and latitude) of ROIs in $\mathcal{S}$. $DT(\mathcal{S})$ has the property that no vertex lies inside the circumcircle of any triangle in $DT(\mathcal{S})$ [D⁺34].

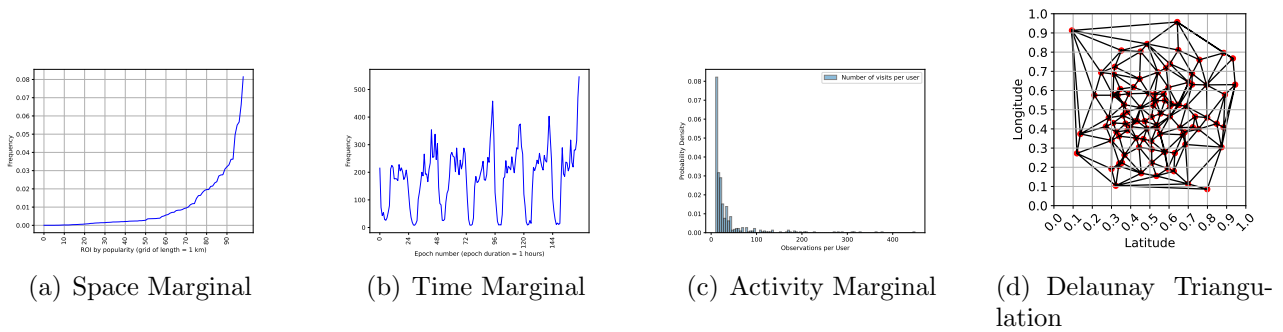


Figure 6.4: **Statistical parameters needed in entry to generate synthetic location traces.** The four statistical parameters for the unicity model by Farzanehfar et al. [FHdM21] include marginal distributions in Figures 6.4(a)-6.4(c) (the dataset's true marginal distributions are shown in red) and the Delaunay triangulation of ROIs in Figure 6.4(d).

We note that the Delaunay triangulation $DT(\mathcal{S})$ is determined by the locations of the ROIs. Since the locations of the ROIs are assumed to be known (Section 6.2.5), $DT(\mathcal{S})$ can be immediately obtained from the release. We explain how the other statistical inputs, the three marginal distributions, can be approximated from the released aggregate in Section 6.4.3.

We now describe our procedure, adapted from Farzanehfar et al. [FHdM21], for generating synthetic traces using the four inputs. For each synthetic trace L^{syn} , we first sample the number of visits n_{visits} from the activity marginal $\mathcal{P}_A$. This determines the number of nonzero entries in the $\mathcal{S} \times \mathcal{T}$ matrix $L_{s,t}^{syn}$. Second, we sample an origin ROI s_0 from the space marginal $\mathcal{P}_S$, and a connected sub-graph $C(s_0)$ from the Delaunay triangulation $DT(\mathcal{S})$, such that $s_0 \in C(s_0)$. $C(s_0)$ will correspond to the set of ROIs that may be visited in L^{syn} . This is done to emulate the natural tendency to move to and from the same proximate locations (e.g. home and work). Finally, we sample n_{visits} spatiotemporal visits (s, t) for which we set $L_{s,t}^{syn} = 1$. For each visit, s is sampled from the space marginal $\mathcal{P}_S$ restricted to $C(s_0)$, and t is sampled from the time marginal $\mathcal{P}_T$. All the sampling steps are independent.

We make two modifications of the original algorithm [FHdM21]. First, to avoid over-saturating unpopular regions, we sample the origin ROI s_0 according to the space marginal $\mathcal{P}_S$ rather than uniformly. Second, to allow users to visit multiple ROIs within the same epoch, we sample the epochs with replacement rather than without replacement.

Our procedure is summarized in Algorithm 4. We note that the procedure for generating a synthetic trace can also be interpreted as running a Markov chain $\{X_i : i = 1, \dots, n_{visits}\}$ over the state space of spatiotemporal pairs $(s, t) \in C(s_0) \times \mathcal{T}$ with transition probabilities to $(s', t') \in C(s_0) \times \mathcal{T}$ proportional to the product of the pmfs $\mathcal{P}_S(s')\mathcal{P}_T(t')$.

6.4.3 Obtaining accurate marginals

In order to generate suitable synthetic traces for ZK MIA, *Adv* must estimate the marginal distributions $\mathcal{P}_S, \mathcal{P}_T, \mathcal{P}_A$, computed over the full population Ω , using the aggregate release $\bar{A}^{\mathcal{U}}$. In this section, we motivate and justify the techniques that we developed to obtain strong

Algorithm 4 GENERATESYNTHETICTRACE

```

1: Inputs:
    $\mathcal{P}_S$ : Approximated space marginal over ROIs
    $\mathcal{P}_T$ : Approximated time marginal over epochs
    $\mathcal{P}_A$ : Approximated activity marginal over trace sizes
    $DT(\mathcal{S})$ : Delaunay triangulation of ROIs
2: Output:
    $L^s$ : A synthetic trace
   // We sample an origin ROI.
3:  $s_0 \leftarrow \text{sample\_from\_distribution}(\mathcal{P}_S, 1)$ 
4: // Use  $DT(\mathcal{S})$  to create a connected subgraph of ROIs including the origin ROI
5:  $C(s_0) \leftarrow \text{generate\_connected\_subgraph}(s_0, DT(\mathcal{S}), \text{n\_rois\_subgraph} = 10)$  (default value from [FHdM21])
6: // Normalize  $\mathcal{P}_s$  restricted to  $C(s_0)$ .
7:  $\mathcal{P}_{C(s_0)} \leftarrow \text{normalize}(\text{restrict}(\mathcal{P}_s, C(s_0)))$ 
8: // Sample the trace size (# visits).
9:  $\text{n\_visits} \leftarrow \text{round}(\text{sample\_from\_distribution}(\mathcal{P}_A, 1))$ 
10: // Randomly sample n_visits ROIs and epochs with replacement.
11:  $\text{ROIs} \leftarrow \text{sample\_from\_distribution}(\mathcal{P}_{C(s_0)}, \text{n\_visits})$ 
12:  $\text{epochs} \leftarrow \text{sample\_from\_distribution}(\mathcal{P}_T, \text{n\_visits})$ 
13: return  $L^s \leftarrow [(\text{ROIs}[i], \text{epochs}[i]) \text{ for } i = 1 \dots \text{n\_visits}]$ 

```

estimates $\hat{\mathcal{P}}_S, \hat{\mathcal{P}}_T, \hat{\mathcal{P}}_A$. This task is especially challenging when privacy measures distort the aggregate data. We develop separate techniques to correct for bias in the case of SSC, and to correct for noise in the case of DP. The effects are shown in Figures 6.5 and 6.7 respectively. Algorithm 5 present a summary of the following steps.

Estimating space and time marginals

Suppose that the data collector releases the aggregate $\bar{A}^u$, which may or may not have privacy measures. Adv can directly compute the empirical space and time marginals, which we denote $\hat{\mathcal{P}}_S^0$ and $\hat{\mathcal{P}}_T^0$, from the released aggregate matrix $\bar{A}^u$.

$$\hat{\mathcal{P}}_S^0(s) = \frac{1}{\|\bar{A}^u\|_1} \sum_{t=1}^{|\mathcal{T}|} \bar{A}_{s,t}^u \quad (6.4)$$

$$\hat{\mathcal{P}}_T^0(t) = \frac{1}{\|\bar{A}^u\|_1} \sum_{s=1}^{|\mathcal{S}|} \bar{A}_{s,t}^u \quad (6.5)$$

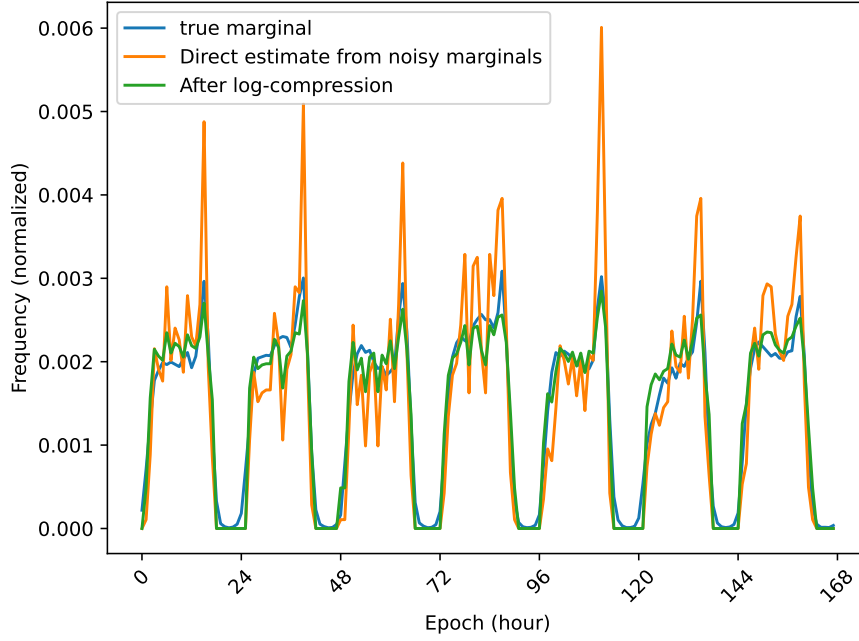


Figure 6.5: **Log compression for SSC aggregates.** SSC biases the estimate obtained from the aggregate by creating more extreme values. The true time marginal from the CDR dataset (plotted for the first week) is better approximated after the empirical estimate from the aggregate ($m = 1000$, $k = 1$) undergoes log compression $\log(1 + \gamma x)$ with γ chosen as in (8).

Raw aggregate: In the case where the released aggregate provide the raw counts, i.e. $\overline{A}^{\mathcal{U}} = A^{\mathcal{U}}$, the empirical marginals tend to be highly accurate. An example is shown in Figure 6.6. The accuracy of these estimates is intuitive because we expect the mobility patterns of an aggregation group $\mathcal{U}$ to resemble those of the population Ω . Thus, we set $\hat{\mathcal{P}}_- = \hat{\mathcal{P}}_-^0$ if the released aggregate is unmodified. We use the subscript $-$ to indicate generality for both the space S and time T marginals.

Suppressed aggregate: However, if the data collector applies SSC with threshold k , i.e. $\overline{A}^{\mathcal{U}} = A_{SSC}^{\mathcal{U}}(k)$, then this will systematically bias the empirical marginal $\hat{\mathcal{P}}_-^0$, because popular ROIs and epochs are more likely to evade suppression. It is therefore easy to see that suppression will reduce the observed probabilities of less popular entries and boost the probabilities of more popular entries.

To correct the bias, we flatten the empirical estimate $\hat{\mathcal{P}}_-^0$ by boosting low frequency counts and reducing high frequency counts. Upon the insight that $\hat{\mathcal{P}}_-^0$ can be likened to an audio signal,

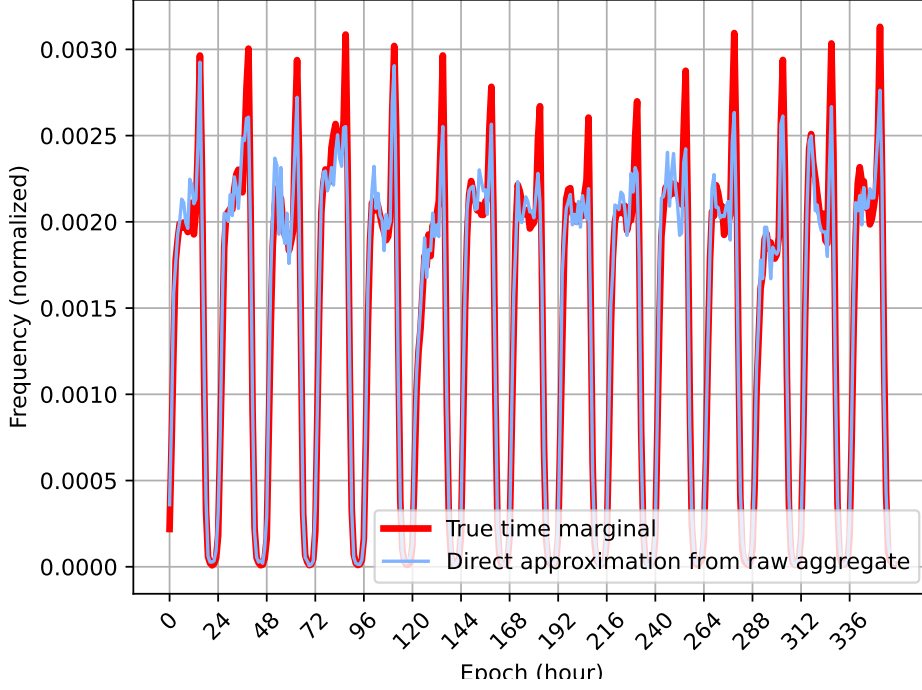


Figure 6.6: **Time marginal approximation.** Time marginal approximation from a raw aggregate over $m = 1000$ users from the CDR dataset.

we adapt the logarithmic compression technique used to reduce dynamic range [M15, MAR04]

$$x \rightarrow \log(1 + \gamma x), \quad x \geq 0, \quad (6.6)$$

where the scaling factor $\gamma \geq 0$ regulates the compression level [M15]. In music signal processing, $x \geq 0$ corresponds to the intensity of a given frequency. In our case, $x \geq 0$ corresponds to probabilities within the empirical marginal $\hat{\mathcal{P}}_-^0$. We choose

$$\gamma(\hat{\mathcal{P}}_-^0) = \max_{x \in \hat{\mathcal{P}}_-^0 : x > 0} \frac{1}{x} \quad (6.7)$$

to automatically parameterize γ based on the smallest observed non-zero probability. We therefore estimate $\hat{\mathcal{P}}_- = \log(1 + \gamma(\mathcal{P}_-^0))$, where we omit the normalization constant.

We do not argue that our choice of method and parameter γ is optimal. However, the debiasing substantially improves the estimate, as shown in Figure 6.5, and it is done without additional information.

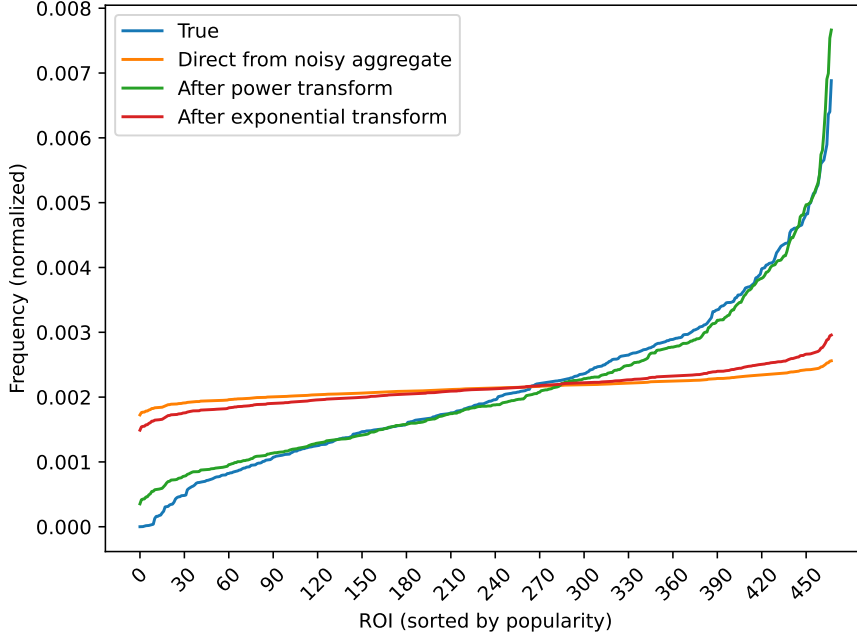


Figure 6.7: **Power transformation for DP aggregates.** DP noise compresses the estimate obtained from the aggregate. The true space marginal from the CDR dataset (organized by popularity) is better approximated after the empirical estimate from the aggregate ($m = 1000$, $\frac{\Delta}{\varepsilon} = 1$) undergoes power transformation x^p with p selected according to Algorithm 7.

DP aggregate. If ε -DP noise is added to each entry in the aggregate release, i.e. $\bar{A}_{s,t}^u = A_{DP}^u(\varepsilon)_{s,t} = A_{s,t}^u + \text{Lap}(\frac{\Delta}{\varepsilon})$, then the noise will overpower the signal in the computation of the empirical marginals (Equation 6.5). This follows from the fact that location aggregates are high-dimensional sparse matrices [PTDC20]. Therefore, conversely to the SSC case, the probabilities within the empirical marginals are compressed, since each probability is characterized mostly by thousands of independent noise samples. This effect is visualized for several different noise scales in Figure 6.8(a). It is also important to notice that, under strong sparsity assumptions, $\hat{\mathcal{P}}_{\mathcal{S}}^0$ converges to the discrete uniform distribution on $\mathcal{S}$ as the number of epochs in the observation period $|\mathcal{T}| \rightarrow \infty$, as proved in Theorem 6.3.

To correct the low variance of the observed probabilities, we propose the power transformation x^p with $p > 1$, followed by renormalization. It is easy to see that this will increase the variance since the probabilities are in $[0, 1]$. Automatically calibrating the power $p > 1$ is a delicate matter. To do so, we start with $p = 1$ and augment p gradually until the transformed distribution achieves the target variance σ^2 . Without any prior knowledge, we consider the case where each

probability is randomly drawn. Equivalently, each probability x_i is sampled from $Unif(0, 1)$, and then renormalized so that the total probability is 1. Let p_i denote the probabilities after normalization, and $\bar{p}$ be the mean of the normalized probabilities. For the space marginal, the variance is

$$\sigma^2 = \frac{\sum_{i=1}^{|S|} (p_i - \bar{p})^2}{|S|}. \quad (6.8)$$

For the variance from the time marginal, we replace $|S|$ with $|T|$ in the above equation. The algorithm for selecting p is given in Algorithm 7 of the Appendix. Figure 6.7 shows that our automatically parameterized power transformation significantly improves the estimate. In contrast, the exponential transformation $(e^x - 1)/\gamma$, which is the inverse of the log compression function $\log(1 + \gamma x)$, fails to denoise because the inverse is inapplicable after considering normalization.

Convergence of empirical marginals

Definition 6.1. (*Oracle average count*) Given a raw aggregate $A^{\mathcal{U}}$, we define the oracle average count function $\mu_S : \mathcal{S} \rightarrow \mathbb{R}^+$ over the set of time intervals $\mathcal{T}$ as

$$\mu_S(s) = \lim_{|\mathcal{T}| \rightarrow \infty} \frac{\sum_{t=1}^{|\mathcal{T}|} A_{s,t}^{\mathcal{U}}}{|\mathcal{T}|}. \quad (6.9)$$

Letting $|\mathcal{T}| \rightarrow \infty$ corresponds to extending the observation period indefinitely. Thus, $\mu_S(s)$ represents the expected number of users $\mathcal{U}$ who visit ROI s at a randomly selected epoch, given infinite location data over the ROIs $\mathcal{S}$.

Definition 6.2. (*Strong sparsity*) We say that $A^{\mathcal{U}}$ is strongly sparse if

$$\mu_S(s) = 0 \quad \forall s \in \mathcal{S} \quad (6.10)$$

Equivalently, $\sum_{t=1}^{|\mathcal{T}|} A_{s,t}^{\mathcal{U}} \in o(|\mathcal{T}|) \quad \forall s \in \mathcal{S}$. This is a strong assumption, as it implies that the visitation rate to each ROI decreases at a sublinear rate.

Lemma 6.1. *Given a fixed geographic region in which location data is collected,*

$$\lim_{|\mathcal{S}| \rightarrow \infty} \frac{\sum_{s=1}^{|\mathcal{S}|} A_{s,t}^{\mathcal{U}}}{|\mathcal{S}|} = 0 \quad (6.11)$$

Proof. $\sum_{s=1}^{|\mathcal{S}|} A_{s,t}^{\mathcal{U}}$ corresponds to the number of users who registered a visit during epoch t . Letting $|\mathcal{S}| \rightarrow \infty$ corresponds to increasingly creating finer regional partitions within the fixed geographic region. $\sum_{s=1}^{|\mathcal{S}|} A_{s,t}^{\mathcal{U}}$ is invariant to increasing $|\mathcal{S}| \rightarrow \infty$, since the same users are observed over the same time. It follows that $\lim_{|\mathcal{S}| \rightarrow \infty} \frac{\sum_{s=1}^{|\mathcal{S}|} A_{s,t}^{\mathcal{U}}}{|\mathcal{S}|} = 0$. $\square$

Lemma 6.2. *Suppose that $Y \sim L \vee 0$, with $L \sim \text{Lap}(b)$. Then, Y has mean*

$$\mathbb{E}[Y] = \frac{b}{2}$$

Proof. Let $L \sim \text{Lap}(b)$. Then, its probability density function (pdf) f_L is given by

$$f_L(x) = \frac{1}{2b} e^{\frac{|x|}{b}} \text{ for } x \in \mathbb{R}$$

which is symmetric about $x = 0$. Hence, $P(X \leq 0) = \frac{1}{2}$. It follows that $Y = X \vee 0$ has the pdf f_Y

$$f_Y(x) = \begin{cases} 0, & \text{for } x < 0 \\ \frac{\delta(x)}{2}, & \text{for } x = 0 \\ \frac{1}{2b} e^{-\frac{x}{b}}, & \text{for } x > 0 \end{cases}$$

where $\delta(x)$ is the Dirac delta function representing the accumulated probability mass at zero.

We then evaluate

$$\begin{aligned}
\mathbb{E}[Y] &= \frac{1}{2b} \int_0^\infty x e^{-\frac{x}{b}} dx \\
&= \frac{1}{2b} \left(-b x e^{-\frac{x}{b}} \Big|_0^\infty + b \int_0^\infty e^{-\frac{x}{b}} dx \right) \\
&= \frac{1}{2b} \left(-b^2 e^{-\frac{x}{b}} \Big|_0^\infty \right) \\
&= \frac{1}{2b} (b^2) = \frac{b}{2}
\end{aligned}$$

□

Theorem 6.3. (Convergence of empirical marginals to uniform distribution under ε -DP) Let $\Delta > 0$ be the global sensitivity and suppose that ε -DP is applied on an aggregate release $\bar{A}^\mathcal{U} = A_{DP}^\mathcal{U}(\varepsilon)$ with post-processed non-negative counts. If the original raw counts $A^\mathcal{U}$ are strongly sparse, then the empirical space and time marginals, $\mathcal{P}_S^0$ and $\mathcal{P}_T^0$, each converge to discrete uniform distributions:

- $\hat{\mathcal{P}}_S^0 \rightarrow \text{Unif}(\mathcal{S})$ in distribution as $|\mathcal{T}| \rightarrow \infty$
- $\hat{\mathcal{P}}_T^0 \rightarrow \text{Unif}(\mathcal{T})$ in distribution as $|\mathcal{S}| \rightarrow \infty$

Proof. We first consider $\hat{\mathcal{P}}_S^0$. It suffices to show that as $\epsilon \rightarrow 0$, $\hat{\mathcal{P}}_S^0(s_0) \xrightarrow{\text{a.s.}} \frac{1}{|\mathcal{S}|}$ for each $s_0 \in \mathcal{S}$.

Let $\epsilon > 0$ and let $b = \frac{\Delta}{\epsilon}$. Recall that ε -DP with post-processed non-negative counts is obtained by $\bar{A}_{s,t}^\mathcal{U} = (A_{s,t}^\mathcal{U} + L_{s,t}(b)) \vee 0$, where $A_{s,t}^\mathcal{U}$ is the true number of visits by users in $\mathcal{U}$ to (s, t) and $\{L_{s,t} \sim \text{Lap}(b) : s \in \mathcal{S}, t \in \mathcal{T}\}$ are i.i.d Laplacian noise samples (Section 3.5.5). By definition,

$$\begin{aligned}
\hat{\mathcal{P}}_S^0(s_0) &= \frac{\sum_{t=1}^{|\mathcal{T}|} \bar{A}_{s_0,t}^\mathcal{U}}{\sum_{s=1}^{|\mathcal{S}|} \sum_{t=1}^{|\mathcal{T}|} \bar{A}_{s,t}^\mathcal{U}} \\
&= \frac{\sum_{t=1}^{|\mathcal{T}|} ((A_{s_0,t}^\mathcal{U} + L_{s_0,t}(b)) \vee 0)}{\sum_{s=1}^{|\mathcal{S}|} \sum_{t=1}^{|\mathcal{T}|} ((A_{s,t}^\mathcal{U} + L_{s,t}(b)) \vee 0)} \\
&= \frac{\sum_{t=1}^{|\mathcal{T}|} ((A_{s_0,t}^\mathcal{U} + L_{s_0,t}) \vee 0)}{\sum_{s=1}^{|\mathcal{S}|} \sum_{t=1}^{|\mathcal{T}|} ((A_{s,t}^\mathcal{U} + L_{s,t}) \vee 0)} \\
&= \frac{\sum_{t=1}^{|\mathcal{T}|} (A_{s_0,t}^\mathcal{U} + L_{s_0,t})}{\sum_{s=1}^{|\mathcal{S}|} \sum_{t=1}^{|\mathcal{T}|} (A_{s,t}^\mathcal{U} + L_{s,t})}.
\end{aligned}$$

We now express $(A_{s,t}^{\mathcal{U}} + L_{s,t}) \vee 0 = L_{s,t} \vee 0 + X_{s,t}$, for some $X_{s,t}$, in order to apply Lemma 6.2 later. Since $A_{s,t}^{\mathcal{U}} \geq 0$, there are three cases:

$$X_{s,t} = \begin{cases} A_{s,t}^{\mathcal{U}}, & \text{if } L_{s,t} \geq 0 \\ A_{s,t}^{\mathcal{U}} + L_{s,t}, & \text{if } L_{s,t} < 0 \text{ and } (A_{s,t}^{\mathcal{U}} + L_{s,t}) \vee 0 > 0 \\ 0, & \text{if } L_{s,t} < 0 \text{ and } (A_{s,t}^{\mathcal{U}} + L_{s,t}) \vee 0 = 0 \end{cases}$$

We therefore have

$$\hat{\mathcal{P}}_S^0(s) = \frac{\frac{\sum_{t=1}^{|\mathcal{T}|} X_{s_0,t}}{|\mathcal{T}|} + \frac{\sum_{t=1}^{|\mathcal{T}|} L_{s_0,t} \vee 0}{|\mathcal{T}|}}{\sum_{s=1}^{|\mathcal{S}|} \left(\frac{\sum_{t=1}^{|\mathcal{T}|} X_{s,t}}{|\mathcal{T}|} + \frac{\sum_{t=1}^{|\mathcal{T}|} L_{s,t} \vee 0}{|\mathcal{T}|} \right)}. \quad (6.12)$$

By sparsity, for each $s \in \mathcal{S}$

$$\frac{\sum_{t=1}^{|\mathcal{T}|} A_{s,t}^{\mathcal{U}}}{|\mathcal{T}|} \rightarrow 0 \text{ as } |\mathcal{T}| \rightarrow \infty \quad (6.13)$$

Also, by the Strong Law of Large Numbers, since $\{L_{s,t}\}$ are i.i.d., and $\mathbb{E}[Lap(b)] = 0$, we have

$$\frac{\sum_{t=1}^{|\mathcal{T}|} L_{s,t}}{|\mathcal{T}|} \xrightarrow{\text{a.s.}} 0 \text{ as } |\mathcal{T}| \rightarrow \infty$$

By linearity,

$$\frac{\sum_{t=1}^{|\mathcal{T}|} A_{s,t}^{\mathcal{U}} + L_{s,t}}{|\mathcal{T}|} \xrightarrow{\text{a.s.}} 0 \text{ as } |\mathcal{T}| \rightarrow \infty$$

Hence, in all three possible cases,

$$\frac{\sum_{t=1}^{|\mathcal{T}|} X_{s,t}}{|\mathcal{T}|} \xrightarrow{\text{a.s.}} 0 \text{ as } |\mathcal{T}| \rightarrow \infty$$

This allows us to simplify

$$\hat{\mathcal{P}}_S^0(s) \xrightarrow{\text{a.s.}} \frac{\frac{\sum_{t=1}^{|\mathcal{T}|} L_{s_0,t} \vee 0}{|\mathcal{T}|}}{\sum_{s=1}^{|\mathcal{S}|} \frac{\sum_{t=1}^{|\mathcal{T}|} L_{s,t} \vee 0}{|\mathcal{T}|}}.$$

Since for all s, t , $L_{s,t} \vee 0 \sim \text{Lap}(b) \vee 0$, Lemma 6.2 implies $\mathbb{E}[L_{s,t} \vee 0] = \frac{b}{2}$. Hence, by the Strong Law of Large Numbers,

$$\frac{\sum_{t=1}^{|\mathcal{T}|} L_{s,t} \vee 0}{|\mathcal{T}|} \xrightarrow{\text{a.s.}} \frac{b}{2} \text{ as } |\mathcal{T}| \rightarrow \infty$$

Finally, for any set of ROIs $\mathcal{S}$, and any $s \in \mathcal{S}$,

$$\widehat{\mathcal{P}}_S^0(s) \xrightarrow{\text{a.s.}} \frac{\frac{b}{2}}{\sum_{s=1}^{|\mathcal{S}|} \frac{b}{2}} = \frac{b}{|\mathcal{S}|b} = \frac{1}{|\mathcal{S}|},$$

A symmetric argument proves $\widehat{\mathcal{P}}_T^0 \rightarrow \text{Unif}(\mathcal{T})$ in distribution as $|\mathcal{S}| \rightarrow \infty$, using Lemma 6.1 instead of strong sparsity. $\square$

Remark. We note that strong sparsity is assumed in Eq. (14) to prove that $\frac{\sum_{t=1}^{|\mathcal{T}|} X_{s,t}}{|\mathcal{T}|} \xrightarrow{\text{a.s.}} 0$ as $|\mathcal{T}| \rightarrow \infty$. Although we expect the oracle average count $\mu_S(s)$ to be very small for most $s \in \mathcal{S}$, due to the sparsity of aggregate location data, it is unlikely to observe $\mu_S = 0$ for real data. Substituting $\mu_S(s)$ in place of 0 in Eq. (14) will not yield the uniform probability $\widehat{\mathcal{P}}_S^0(s) \xrightarrow{\text{a.s.}} \frac{1}{|\mathcal{S}|}$, but it will be a close approximation, provided that $\frac{\Delta}{\epsilon} \gg \mu_S$ and that the number of epochs is large.

In practice, fixed dimensions for S and T will prevent the empirical marginals from completely converging to the uniform distribution. This is demonstrated for different noise scales on the Milan dataset (which has $|\mathcal{S}| = 100$ and $|\mathcal{T}| = 168$) in Figure 6.8.

Estimating the activity marginal

The released aggregate $\overline{A}^{\mathcal{U}}$ does not leak granular information about the activity marginal $\mathcal{P}_A$. However, Adv may obtain the empirical mean number of visits per user according to the released aggregate,

$$\widehat{\mu}_0 = \frac{1}{m} \sum_{s,t \in \mathcal{S} \times \mathcal{T}} \overline{A}_{s,t}^{\mathcal{U}}. \quad (6.14)$$

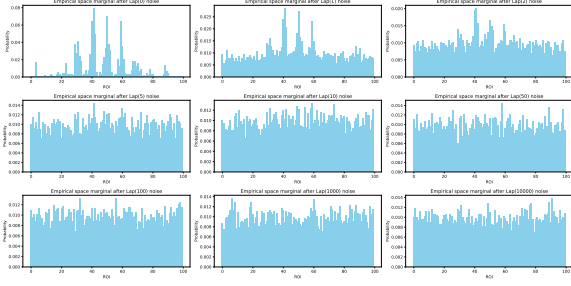
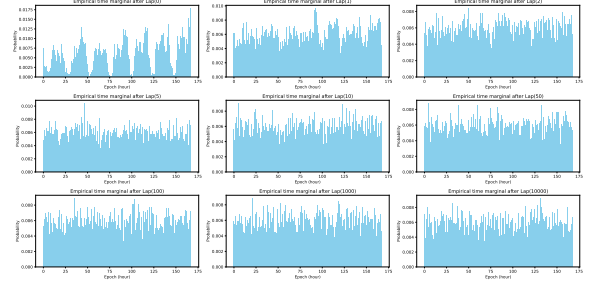
(a) Milan space marginals estimated from ε -DP aggregates(b) Milan time marginals estimated from ε -DP aggregates

Figure 6.8: **Representation of the space and time marginals obtained from ε -DP aggregates over $m = 1000$ users from the Milan dataset.** They are plotted for different noise scales $\frac{\Delta}{\varepsilon}$. Interestingly, the distribution does not converge to a uniform distribution as the noise scale increases, due to the increasing variance of $Lap(\frac{\Delta}{\varepsilon})$

If the aggregate is raw, then we expect $\hat{\mu}_0$ to be a strong estimate due to well-known regularity results about population-wide mobility activity [SBC⁺13, SMS⁺08, FHdM21]. In this case, *Adv* sets $\hat{\mu} = \hat{\mu}_0$.

However, if either SSC or ε -DP is applied, then the estimate $\hat{\mu}_0$ would fail. Algorithm 6 in the Appendix describes how *Adv* can obtain a better estimate $\hat{\mu}$. Given an aggregate release $\bar{A}^u$ of size m , *Adv* can use $\hat{\mathcal{P}}_S$ and $\hat{\mathcal{P}}_T$ to iteratively improve their estimate, starting with μ_0 . Given guess $\hat{\mu}_n$, *Adv* creates a synthetic aggregate by generating m synthetic traces, parameterized by $\hat{\mathcal{P}}_S, \hat{\mathcal{P}}_T$ and $\hat{\mathcal{P}}_A \sim \hat{\mu}_n$. *Adv* may then apply the same privacy measures that were applied on $\bar{A}^u$. $\hat{\mu}_{n+1}$ is obtained by increasing or decreasing $\hat{\mu}_n$ relative to the difference in counts with $\bar{A}^u$.

Once $\hat{\mu}$ is obtained, *Adv* can simply pick $\hat{\mathcal{P}}_A \sim \hat{\mu}$, such that each synthetic trace has $\hat{\mu}$ visits. However, it is well known that human mobility activity follows a heavy-tailed distribution, i.e., a heavier tail than the exponential distribution. The best approximations are lognormal, beta, or power-law distributions [FHdM21, SBC⁺13, SMS⁺08].

It would be reasonable for *Adv* to use a heavy-tailed distribution with mean $\hat{\mu}$, but these distributions require a second parameter, e.g. skewness, to determine the distribution shape. *Adv* can use well-known parameters from other cities' datasets to complete the estimate [SBC⁺13, SMS⁺08], but to ensure that *Adv* does not use additional knowledge, we assume that they use

the sub-optimal estimate $\hat{\mathcal{P}}_A \sim \text{Exp}(\hat{\mu})$, as shown in Figure 6.9.

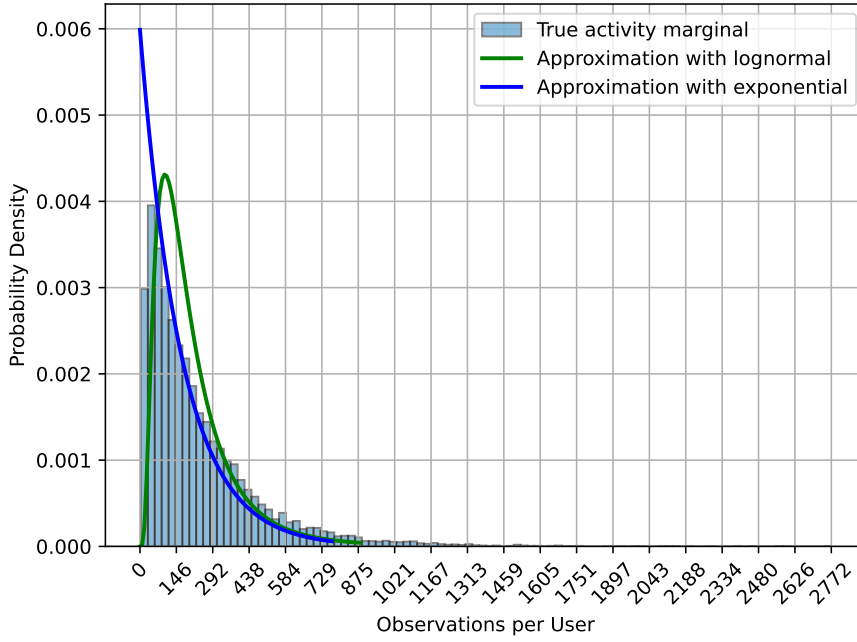


Figure 6.9: **Exponential estimate for activity marginal.** Once the mean number of visits is approximated, the activity marginal from the CDR dataset is best approximated by a lognormal distribution with an optimal skew parameter. However, it can also be approximated without additional parameters by an exponential distribution.

Algorithms

We here present the main algorithms required to generate synthetic traces from the released aggregate for our ZK MIA, using the three marginal probability distribution, as described in the previous paragraphs.

The following algorithm (Algorithm 5) estimates the three marginal probability distributions required to run Algorithm 4. We estimate the marginals via our denoising and debiasing techniques (from Section 6.4.3), depending on the application of privacy measures on $\overline{A}^U$.

The following algorithms describes our procedure for achieving an estimate $\hat{\mu}$ for the mean number of visits per user in the dataset given a privacy-aware aggregate release. Recall that $\mathcal{P}_A$ is set to $\text{Exp}(\hat{\mu})$. Algorithm 7 describes our procedure for computing which degree p will work best in the power transformation, to correct the empirical marginal $\hat{\mathcal{P}}_-^0$ obtained directly from a ϵ -DP aggregate release.

Algorithm 5 APPROXIMATE MARGINALS FROM AGGREGATE

```

1: Inputs:
    $\bar{A}^{\mathcal{U}}$  : Released aggregate
    $m$ : Aggregate group size
    $p$ : Specified probability distribution family
2: Output:
    $\mathcal{P}_S$ : Approximated space marginal over ROIs
    $\mathcal{P}_T$ : Approximated time marginal over epochs
    $\mathcal{P}_A$ : Approximated activity marginal over trace sizes
   // Compute direct estimates.
3:  $\mathcal{P}_S, \mathcal{P}_T \leftarrow \text{compute\_empirical\_marginals}(\bar{A}^{\mathcal{U}})$ 
4:  $\mu_{visits}^0 \leftarrow \text{sum\_entries}(\bar{A}^{\mathcal{U}})/m$ 
5: if  $\bar{A}^{\mathcal{U}} = A^{\mathcal{U}}$  then
6:   // Return direct estimates if no privacy.
7:    $\mathcal{P}_A \leftarrow \text{fit\_dist}(p, \mu_{visits}^0)$ 
8:   return  $\mathcal{P}_S, \mathcal{P}_T, \mathcal{P}_A$ 
9: end if
10: if  $\bar{A}^{\mathcal{U}} = A_{SSC}^{\mathcal{U}}(k)$  and  $k > 0$  then
11:   // Apply log compression if SSC.
12:    $\mathcal{P}_S, \mathcal{P}_T \leftarrow \text{log\_compression}(\mathcal{P}_S, \mathcal{P}_T)$ 
13: end if
14: if  $\bar{A}^{\mathcal{U}} = A_{DP}^{\mathcal{U}}(\varepsilon)$  or  $\bar{A}^{\mathcal{U}} = A_{DP,SSC}^{\mathcal{U}}(\varepsilon, k)$  then
15:   // Apply power transformation if DP.
16:    $\mathcal{P}_S, \mathcal{P}_T \leftarrow \text{power\_transform}(\mathcal{P}_S, \mathcal{P}_T)$ 
17: end if
18: // Apply Algorithm 6
19:  $\mu_{visits} \leftarrow \text{estimate\_mean}(\mu_{visits}^0, A, m, \mathcal{P}_S, \mathcal{P}_T, k, \varepsilon)$ 
20:  $\mathcal{P}_A \leftarrow \text{fit\_dist}(p, \mu_{visits})$ 
21: return  $\mathcal{P}_S, \mathcal{P}_T, \mathcal{P}_A$ 

```

6.4.4 Paired sampling for training

When MIAs target high-dimensional aggregate data, such as location data, the membership classifier must handle noise arising from thousands of entries, which are unrelated to the target record. For example, many of the *IN* training aggregates may coincidentally have high counts in entries that are absent in the target record. This would skew the decision boundary of the membership classifier, which may lead to false positives when testing. We may similarly obtain false negatives due to spurious patterns within the *OUT* training aggregates. These challenges are compounded by the implementation of privacy measures. For example, ε -DP would add noise to each entry. Given the dimensionality and nature of the aggregate data, spurious patterns will likely skew the decision boundary, even when hundreds or thousands of

Algorithm 6 ESTIMATEMEAN

```

1: Inputs:
    $\mu_0$ : Initial guess for mean visits
    $A$ : Released aggregate
    $DT(\mathcal{S})$ : Delaunay triangulation of ROIs
    $\mathcal{P}_s$ : Estimated space marginal
    $\mathcal{P}_t$ : Estimated time marginal
    $k$ : Suppression threshold
    $\epsilon, \Delta$ : DP parameters
    $m$ : Aggregate group size
2: Additional parameters
3: tol: Tolerance for stopping
4: max_iter: Max iterations
5: Output:
    $\mu$ : Approximated mean visits per user
6:  $\mu \leftarrow \mu_0$ 
7: for  $i = 1$  to max_iter do
8:    $A_1 \leftarrow \text{initialize\_matrix}()$ 
9:   // Create a synthetic aggregate of size  $m$ .
10:  for  $j = 1$  to  $m$  do
11:    // Generate synthetic trace via with  $\mu$  visits.
12:     $A_1 \leftarrow A_1 + \text{generate\_synthetic\_trace}(\mathcal{P}_s, \mathcal{P}_t, \mu, DT(\mathcal{S}))$ 
13:  end for
14:   $A_1 \leftarrow \text{apply\_privacy\_measures}(A_1, k, \epsilon, \Delta)$ 
15:  // Increase or decrease the estimate  $\mu$  accordingly
16:   $\mu \leftarrow \mu_0 + (\text{sum}(A) - \text{sum}(A_1)) / m$ 
17:  if  $|\mu - \mu_0| < \text{tol}$  then return  $\mu$ 
18:  end if
19: end for
20: return  $\mu$ 

```

Algorithm 7 PSELECTION

```

1: Inputs:
    $\sigma_0$ : Reference variance
    $\mathcal{P}$ : Space or time marginal to be modified
    $\epsilon_{tol}$ : Error tolerance
2: Output:
    $p$ : Degree for transformation  $x^p$  that sets the variance of  $\mathcal{P}$  to approximately match  $\sigma_0$ 
3:  $\sigma \leftarrow \text{compute\_variance}(\mathcal{P})$  // We compute the variance from the original
   marginal value.
4:  $p \leftarrow 1$ 
5: while  $|\sigma_0 - \sigma| > \epsilon_{tol}$  do
6:    $\sigma \leftarrow \text{compute\_variance}(\text{pow}(\mathcal{P}, p))$ 
7:   // Increment the power  $p$  until estimate is in range.
8:    $p \leftarrow p + 0.01$ 
9: end while
10: return  $\sigma$ 

```

training aggregates are sampled. Given a fixed number of training samples, we demonstrate that the way in which the training set is sampled strongly influences the performance of the MIA. In particular, the sampling technique can guide the convergence of the decision boundary in order to prevent misclassification due to noise.

To the best of our knowledge, all previous MIAs on aggregate location data sampled their training set via independent random sampling [PTDC17, OJGdM19, ZZZ20, PTDC20]. Training aggregates are created by independently sampling groups of m users from the population Ω and labeling them according to the target u^* 's presence.

On the one hand, independent sampling discourages overfitting to the training data by exposing the classifier to a wide variation of samples. On the other hand, independent sampling does nothing to prevent spurious patterns from distorting the decision boundary.

We propose a paired sampling technique to guide the convergence of the decision boundary. The idea is to use sampling to help the classifier identify the differential impact of the target record at the aggregate level. Paired sampling independently samples groups of $m - 1$ users from $\Omega \setminus \{u^*\}$. Then, an *IN* sample is created by adding u^* as the group's last member, and an *OUT* sample is created by adding another randomly selected user. The training set is therefore characterized by a set of *IN/OUT* pairs, which differ in exactly one record (the target's). If noise is added to aggregates prior to release, then *Adv* must inject the same noise sample ε to each paired sample, $A^{u^{IN}}$ and $A^{u^{OUT}}$, to ensure that the target's differential impact is preserved between each *IN/OUT* pair.

Paired sampling therefore actively encourages the membership decision boundary to be formed based on relevant criteria related to the target. It also discourages spurious decision boundaries because of the high degree of similarity between *IN/OUT* pairs. An extreme value in an aggregate entry from an *IN* sample will be matched with a similar value from its paired *OUT* sample, with high probability. However, we note that using paired sampling effectively halves the training variation compared to independent sampling. Our experiments in Section 6.6.4 demonstrate that paired sampling outperforms independent sampling across all tested settings of ε -DP noise addition. Hence, guiding the decision boundary towards relevant membership

criteria often takes precedence over maximizing training variation. For completeness, we note that while we developed, studied, and named paired sampling independently, we later found that a similar idea was used in Bauer and Bindschaedler [BB20] but had not been compared to independent sampling nor used elsewhere in the literature so far to the best of our knowledge.

6.5 Experimental setup

To evaluate the efficacy of our ZK MIA, we compare it with the state-of-the-art Knock Knock (KK) MIA [PTDC17, PTDC20] using aggregated location data from two different datasets. Those datasets, CDR and Milan, are described further in Appendix B.2.

6.5.1 MIA implementation

We perform a fair comparison between KK MIA and ZK MIA by training the binary membership classifier using the same parameters and architecture. This also helps us isolate the effect of removing auxiliary data on performance. We use a Logistic Regression binary classifier with default hyperparameters and $\mathbb{L}_1$ regularization, implemented using the Python package *sklearn* [PVG⁺11]. The number of training groups $n_{train} = 400$ matches previous implementations [PTDC17, PTDC20], and the groups are selected using paired sampling, unless specified otherwise. We additionally fine tune the decision boundary using $n_{val} = 100$ balanced independently sampled validation groups. We flatten the aggregates and feed them directly into the classifier as a vector, without any processing, such as PCA or feature extraction. Finally, as done in Pyrgelis et al. [PTDC17], *Adv* applies the same privacy measures to training and validation aggregates, if the released aggregate is privacy-aware.

Knock-Knock. To implement the Knock-Knock MIA, we provide *Adv* with a reference group *Ref* of 5000 real user traces (including the target trace L^{u^*}) when attacking the larger CDR dataset. We set $|Ref| = 2500$ for the Milan dataset. We note that this significantly surpasses previous reference sizes ($|Ref| = 1100$) implemented by Pyrgelis et al. [PTDC17].

Zero Auxiliary Knowledge. Our Zero Auxiliary Knowledge MIA is structurally identical to KK (PS), but ZK has a synthetic reference Ref , rather than a set of real traces. This reference is created according to the methodology detailed in Section 6.4. Setting $|Ref| = 5000$ allows for a direct comparison of the functionality of synthetic traces to real ones for the purpose of the MIA. However, we remark that capping the number of synthetic traces at 5000 is an artificial restriction, since Adv may generate arbitrarily more synthetic traces and achieve better performance, as shown in Figure 6.10. By default, we assume full access to the target trace L^{u*} . This assumption is relaxed for Experiment 6.6.3.

Figure 6.10 illustrates how increasing the number of synthetic traces available to the adversary improves the MIA’s performance up to marginal returns.

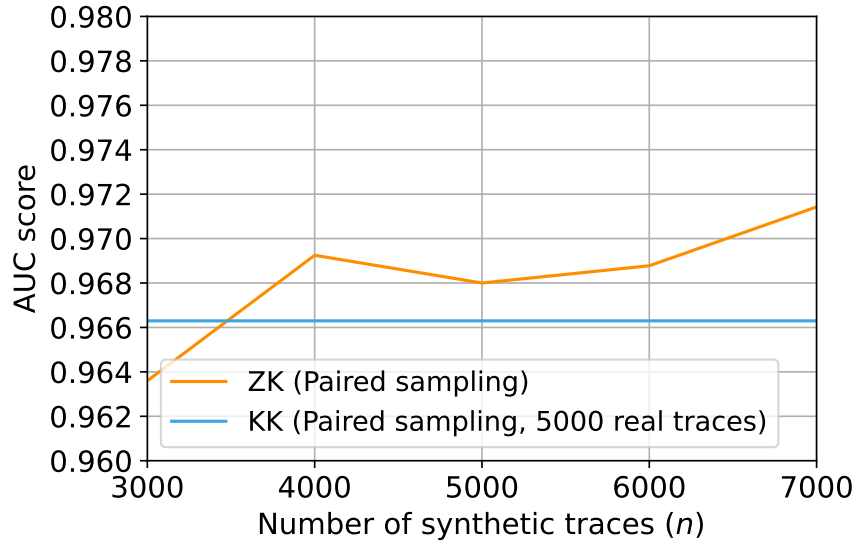


Figure 6.10: **AUC of the Zero Auxiliary Knowledge MIA for different number of synthetic traces.** This results are generated in the setting $m = 1000$, $k = 1$, and $\varepsilon = 1$ -DP at the event level.

6.5.2 Evaluation

Default experimental parameters. We randomly select $n_{targets} = 50$ targets from the dataset for evaluation. These targets are re-used for each experiment. Furthermore, in each experiment, $n_{test} = 100$ *independently sampled* balanced test aggregates are created for each target, and shared across both MIAs to ensure that the test sets are identical. As done

in [PTDC17], the test aggregate user groups are sampled from a disjoint set of users to the Knock-Knock adversarial reference Ref , plus the target u^* . This corresponds to roughly $11,000 - 5000 = 6000$ user traces for CDR and $5000 - 2500 = 2500$ user traces for Milan.

We perform all experiments on size $m = 1000$ aggregates, which matches the largest aggregate size tested in Pyrgelis et al. [PTDC17] and exceeds the largest aggregate size tested in Zhang et al. [ZZZ20] (800). We do not vary m since the relationship between aggregate group size and MIA effectiveness has already been documented extensively [PTDC17, PTDC20, ZZZ20]. We perform the Knock-Knock and Zero Auxiliary Knowledge MIAs in this setting under different privacy measures. We also perform experiments such that Adv only knows a fraction p_{u^*} of the target trace L^{u^*} , but by default, we assume that they know the full trace, i.e. $p_{u^*} = 1$.

Evaluation metrics. In the past, MIAs on aggregate location data have been primarily evaluated using the area under the ROC curve (AUC score) as a metric [PTDC17, ZZZ20]. For this reason, and its suitability for assessing the strength of a binary classifier, we use the mean AUC over all targets as our primary metric. However, we also include the mean attack accuracy over all targets as a secondary metric, we list those scores in additional tables for reference.

6.6 Experimental results

6.6.1 Suppression of Small Counts

We first compare the performances of KK and ZK on aggregates whose counts have been suppressed according to threshold k . We apply SSC with thresholds $k \in \{0, 1, 2, 3, 4, 5\}$ on test aggregates of size $m = 1000$. We note that the case $k = 0$ corresponds to releasing a raw aggregate. We remark that there is a trivial rule that sufficiently determines non-membership in this special case.

Rule ($k = 0$): If u^* visits ROI s during epoch t and no users in the aggregation group $\mathcal{U}$ visit

(s, t) , then u^* cannot be in $\mathcal{U}$, i.e.,

$$\exists s \in \mathcal{S}, t \in \mathcal{T} : A_{s,t}^{\mathcal{U}} = 0 \wedge L_{s,t}^{u^*} = 1 \implies u^* \notin \mathcal{U}$$

We therefore incorporate this rule when $k = 0$, such that both MIAs first check if the released aggregate elicits the contradiction. If so, we immediately predict *OUT*. Otherwise, we train, validate, and test the classifier as usual. The rule is invalid for $k > 0$, since it would predict *OUT* whenever u^* has a visit to a suppressed entry.

Results. Figure 6.11 shows that membership inference is a trivial task when applied to raw aggregates ($k = 0$). Table 6.1 shows the results for our secondary metric, the accuracy.

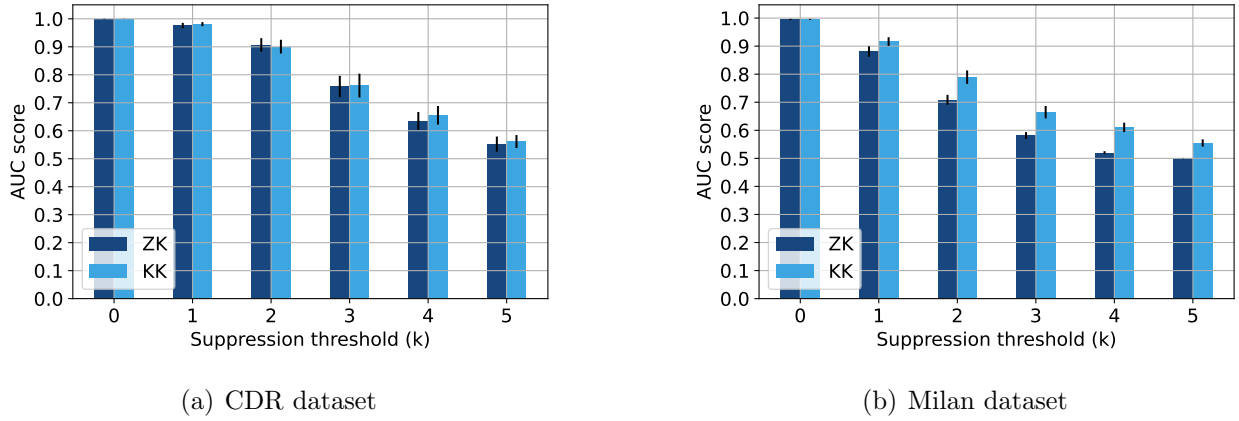


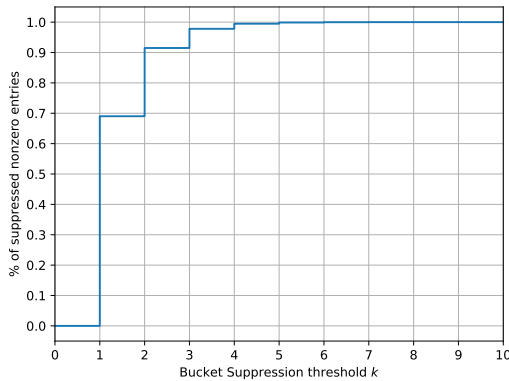
Figure 6.11: **Results comparison between ZK and KK while varying suppression thresholds.** Mean AUC scores with standard error for ZK and KK on size 1000 aggregates with various suppression thresholds

Table 6.1: **Mean accuracy scores with standard error for KK and ZK over various suppression thresholds.** Results are computed on size 1000 aggregates from the CDR and Milan datasets across various suppression thresholds k .

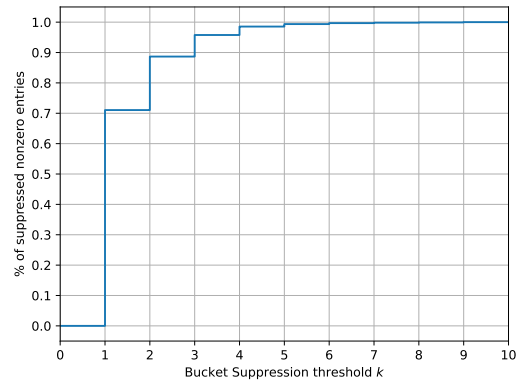
k	CDR dataset		Milan dataset	
	KK	ZK	KK	ZK
0	0.980 ± 0.012	0.991 ± 0.008	0.990 ± 0.005	0.990 ± 0.003
1	0.907 ± 0.028	0.879 ± 0.025	0.767 ± 0.018	0.700 ± 0.017
2	0.807 ± 0.031	0.827 ± 0.026	0.683 ± 0.019	0.631 ± 0.013
3	0.685 ± 0.032	0.687 ± 0.031	0.600 ± 0.016	0.550 ± 0.009
4	0.597 ± 0.024	0.603 ± 0.027	0.566 ± 0.011	0.512 ± 0.003
5	0.543 ± 0.018	0.528 ± 0.019	0.536 ± 0.010	0.500 ± 0.000

Both ZK and KK achieve near perfect AUC (≥ 0.99) (respectively near perfect accuracy) on both datasets. This implies that aggregation is not an effective privacy mechanism in itself to protect high-dimensional location data from MIAs by weak or strong adversaries.

The results for $k > 0$ reveal two general patterns. First, ZK compares closely with KK across different levels of SSC. On the CDR dataset, ZK’s AUC stays within 0.02 of KK for each k . We observe slightly worse results on the Milan dataset, but ZK still stays within 0.9 AUC of KK for each k . Second, there is a monotonic decrease in performance when the threshold k is increased. For $k = 5$, the AUC is always less than 0.55. This follows from the fact that suppression reduces the amount of available information, and 99% of all nonzero entries are suppressed by $k = 5$, as shown in Figure 6.12. Therefore, although SSC eventually mitigates the MIAs, it may come at the cost of destroying virtually all utility. Those observation still holds in the case of our secondary metric, the accuracy.



(a) CDR dataset



(b) Milan dataset

Figure 6.12: **The percentage of nonzero entries that are suppressed in a size $m = 1000$ aggregate.** We plot this percentage after undergoing SSC with threshold k .

Both MIAs perform worse on the Milan dataset compared to the CDR dataset. This is expected because we observe 6 times less data per target in the Milan dataset, as shown in Figure B 1 of the Appendix B.2. Indeed, people generally tweet less often than they text and call. ZK MIA is more affected by dataset sparsity, given its dependence on marginal distribution estimates, which become less reliable in sparser datasets.

6.6.2 ϵ -DP noise addition

Informed by practical applications of DP [Des21], we consider event level and user-day level to be the privacy units, and we vary the privacy budget $\epsilon \in \{0.1, 0.5, 1.0, 5.0, 10.0\}$ for each unit.

Event level DP. An event is equivalent to a visit by a user to $(s, t) \in \mathcal{S} \times \mathcal{T}$. To offer privacy protection over an event, we set the global sensitivity $\Delta = 1$. ϵ -DP is then ensured by adding $Lap(\frac{1}{\epsilon})$ noise to each count in the aggregate matrix.

User-day level DP. In order to protect each user's daily contributions without adding excessive noise, it is common to restrict user contributions prior to aggregation to achieve a smaller global sensitivity Δ [ABC⁺20, HD21]. We analysed daily activity distributions, and had them preprocessed such that a user may only contribute up to $\Delta = 20$ visits in any given day for CDR, and $\Delta = 10$ visits in any given day for Milan. ϵ -DP at the user-day level is then ensured by adding $Lap(\frac{\Delta}{\epsilon})$ noise to each count in the aggregate matrix.

Results. Figure 6.13 shows that ZK MIA matches KK MIA across all tested DP settings (we additionally provide Tables 6.2 and 6.3 for results using our secondary metric, the accuracy). Indeed, ZK maintained a mean AUC within 0.06 of KK (PS) across each of the 10 privacy settings for both datasets. KK and ZK notably succeeded for many of the tested privacy budgets ϵ , particularly in the event level setting. Indeed, we observed $AUC \geq 0.9$ for both MIAs whenever the noise scale $\frac{\Delta}{\epsilon} \leq 2$ for the CDR dataset, and $\frac{\Delta}{\epsilon} \leq 1$ for the Milan dataset. These settings are in line with many real-life applications [KAB23, Des21]. Conversely, user-day level DP with privacy budget $\epsilon \leq 1.0$ effectively reduced both MIAs to an AUC below 0.55. We discuss the significance of these results with respect to practical mitigations in Section 6.7. We see our observations to hold in the case of our secondary metric, the accuracy. Table 6.2 present results for event level DP, while Table 6.3 present results for user-day level DP.

Table 6.2: **Mean accuracy scores with standard error for KK and ZK for event level DP.** Results are computed on size 1000 aggregates from the CDR and Milan datasets across various privacy budgets ε for event level DP.

ε	CDR dataset		Milan dataset	
	KK	ZK	KK	ZK
0.1	0.588 ± 0.019	0.555 ± 0.014	0.539 ± 0.006	0.549 ± 0.007
0.5	0.848 ± 0.028	0.791 ± 0.019	0.744 ± 0.010	0.634 ± 0.007
1.0	0.920 ± 0.026	0.907 ± 0.018	0.850 ± 0.016	0.594 ± 0.008
5.0	0.906 ± 0.035	0.934 ± 0.019	0.881 ± 0.022	0.660 ± 0.018
10.0	0.923 ± 0.029	0.934 ± 0.018	0.920 ± 0.021	0.671 ± 0.019

Table 6.3: **Mean accuracy scores with standard error for KK and ZK for user-day level DP.** Results are computed on size 1000 aggregates from the CDR and Milan datasets across various privacy budgets ε for user-day level DP.

ε	CDR dataset		Milan dataset	
	KK	ZK	KK	ZK
0.1	0.502 ± 0.014	0.502 ± 0.010	0.497 ± 0.006	0.496 ± 0.006
0.5	0.508 ± 0.014	0.526 ± 0.016	0.517 ± 0.006	0.519 ± 0.006
1.0	0.533 ± 0.014	0.539 ± 0.014	0.534 ± 0.006	0.544 ± 0.007
5.0	0.723 ± 0.020	0.676 ± 0.018	0.746 ± 0.009	0.680 ± 0.008
10.0	0.874 ± 0.025	0.825 ± 0.019	0.870 ± 0.014	0.777 ± 0.018

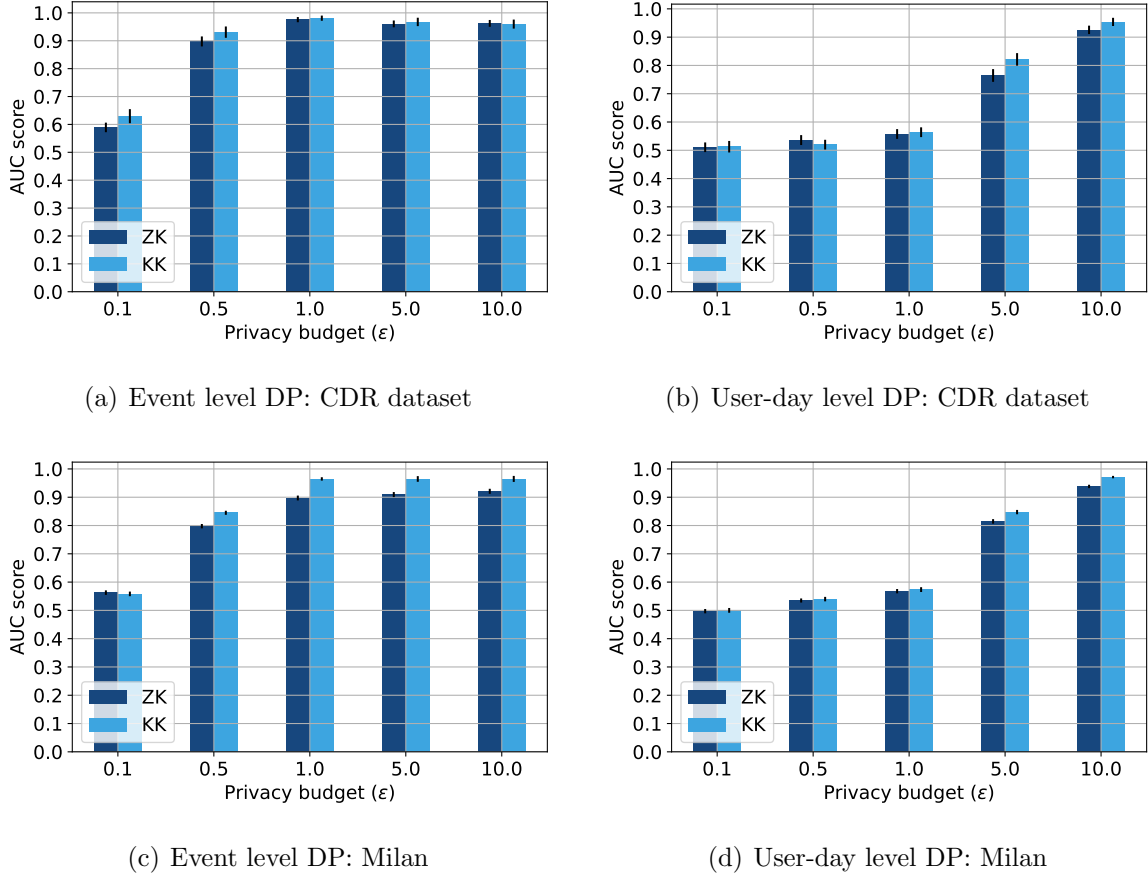


Figure 6.13: **Result comparison between our ZK-MIA and KK-MIA against DP-aggregates.** Mean AUC scores with standard error for ZK and KK on size 1000 aggregates with various privacy units and budgets ϵ .

6.6.3 Partial knowledge of the target trace

We now relax the assumption that Adv knows the full target trace L^{u^*} . This is in line with our ZK threat model, and we expand KK MIA for this setting to be able to compare methods. To simulate a weaker adversary, we suppose that Adv only knows a subset of the target u^* 's visits. We assume that Adv only knows a random fraction $p_{u^*} \in \{0.1, 0.25, 0.5, 0.75, 1.0\}$ of the trace L^{u^*} . The number of retained visits is rounded up to the next integer to prevent cases where Adv knows 0 visits. For example, if a target has 4 visits and $p_{u^*} = 0.1$, then this would correspond to Adv knowing 1 visit. This partial trace is used instead of the full trace when creating IN training and validation aggregates. The full trace L^{u^*} is still used for IN test aggregates.

We perform this experiment in the setting where the data collector applies event-level DP with $\epsilon = 1$, followed by $k = 1$ suppression. We choose this setting for a couple of reasons. First,

to study the degradation of the MIAs with decreasing information about the target, we choose a setting where *Adv* would succeed given the full target trace. Previous experiments revealed that $\epsilon = 1$ -DP at the event level and $k = 1$ suppression were not effective in preventing the MIAs by themselves, as the MIAs achieved $\text{AUC} > 0.97$ on the CDR dataset and $\text{AUC} > 0.9$ for Milan. Second, we combine the two defense mechanisms to see if suppression has an observable mitigation effect when applied following DP noise addition. By the post-processing property of DP, this would not alter the theoretical performance bound. However, zeroing all counts ≤ 1 might add a layer of complexity that affect MIAs in practice.

Results. First, we note that applying $k = 1$ SSC on top of $\epsilon = 1.0$ event level DP has an insignificant effect on the MIAs. For the full target trace, we continue to observe $\text{AUC} > 0.97$ on the CDR dataset and $\text{AUC} > 0.9$ on the Milan dataset. The only MIA with a noticeable decline was KK MIA on the Milan dataset, which dropped from 0.97 AUC to 0.9 AUC, as it can be seen in Figure 6.14.

Although decreasing the fraction of the target trace known to the adversary from 1 to 0.1 decreases the performance of the MIAs, the corresponding degradation is relatively gradual. All AUCs are captured within a range of 0.13 on the CDR dataset, and within a range of 0.22 on the Milan dataset. Even the lowest observed AUC by ZK MIA on the CDR dataset (0.84 when 10% of the target trace is known) achieves high discrimination. We note that the 50 targets in both datasets have a wide variation in trace size, as shown in Figure B 1 of the Appendix. For some targets, *Adv* will only know one of the target’s visits, whereas for others, they will still know dozens and be able to infer membership easily. Interestingly, we note that knowing a single visit from a target trace can still train a classifier that is better than random. For one CDR target with 9 visits in their full trace, we observed ZK achieve an AUC of 0.660 across 100 aggregates when only 1 random visit was known. Although far from perfect, we found this surprising, as it shows that even a single visit by the target can inform an MIA against a noisy aggregate over 1000 users.

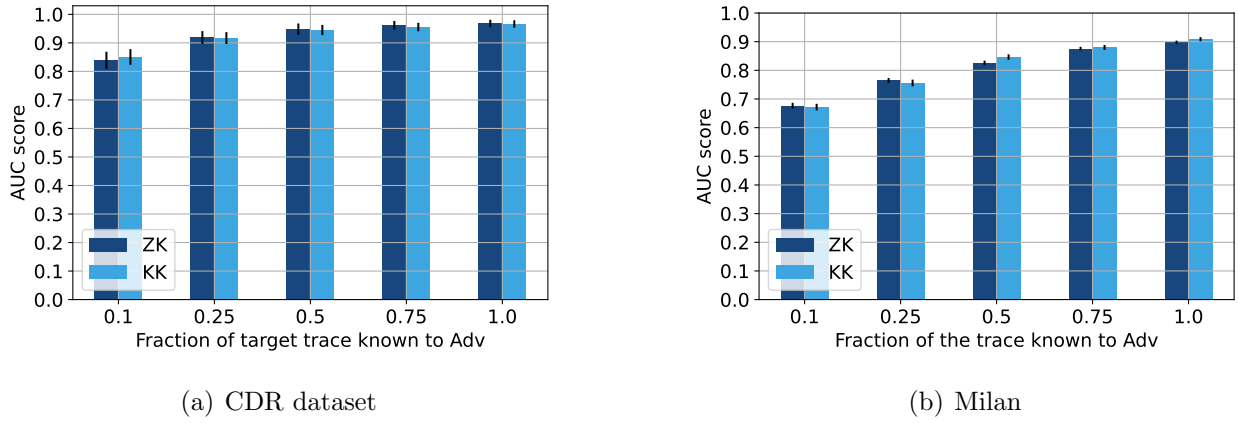


Figure 6.14: **Result comparison between our ZK-MIA and KK-MIA with only the knowledge of a fraction of the trace.** Mean AUC scores with standard error for ZK and KK on size 1000 aggregates with $\epsilon = 1$ event DP protection and $k = 1$ suppression for varying fractions of the known target trace.

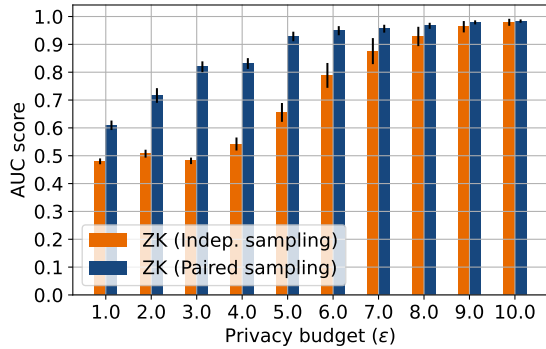
6.6.4 Paired Sampling vs. Independent Sampling

In this section, we study the performance of KK MIA and ZK MIA when we vary the sampling technique used for creating their training aggregates, i.e., paired sampling (PS) or independent sampling (IS). To test this, we consider the implementation of user-day ϵ - DP on the Milan dataset across the privacy budgets $\epsilon = 1, 2, 3, \dots, 10$ for all four possible MIAs: KK (PS), KK (IS), ZK (PS), ZK (IS).

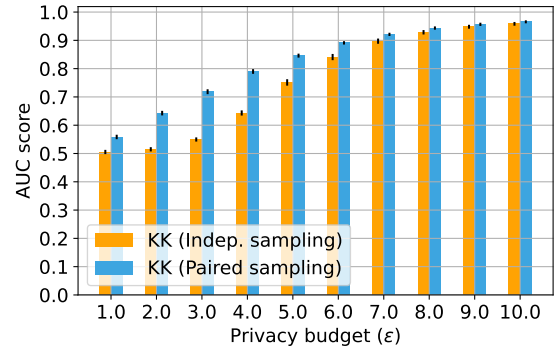
Results. From Figure 6.15, we see that the paired sampling MIA always outperforms its independent sampling equivalent across all privacy budgets $\epsilon = 1, 2, 3, \dots, 10$. Paired sampling provides the largest boost when the inference task is challenging but not intractable. In particular, we notice a few striking examples of ZK (PS) drastically outperforming ZK (IS) in the middle of the graph. For $\epsilon = 4$, ZK (IS) is basically random ($AUC = 0.54$), yet simply switching to paired sampling enables the classifier to achieve an AUC of 0.83.

The improvement achieved by switching from independent sampling to paired sampling is less significant for KK in this experiment. This suggests that using training aggregates sampled from a reference dataset of real traces may introduce less randomness to the membership classifier's decision boundary, compared to when we use a synthetic reference. This is intuitive because of

ZK’s probabilistic generation method, which relies on sampling from three different estimated distributions. However, the MIAs have indistinguishable performance when both attacks use paired sampling, with the difference in AUC always staying within 0.02 of one another across the 10 privacy settings. This suggests that paired sampling effectively eliminates the noise contributed by coincidental patterns in random entries, and enables the membership classifier to form a suitable decision boundary.



(a) Zero Auxiliary Knowledge



(b) Knock-Knock

Figure 6.15: **Study of the impact of paired sampling on both attack settings ZK and KK.** Mean AUC scores with standard error for ZK and KK on size 1000 aggregates from the Milan dataset across different privacy budgets ϵ under user-day level ϵ -DP, for MIAs using independent sampling vs. paired sampling.

6.6.5 Removing the time knowledge over traces

In this experiment, we now assume that the adversary only has access to some of the locations that the target has visited, without knowing the epochs during which the visits were done. For example, the adversary may know the target’s home and work. To model this attack setting, we suppose that the adversary either knows the target’s top- K most visited ROIs, for $K = 1, 2, 3$, or the full set of the target’s visited ROIs during the observation period. In one implementation, which we call ”greedy”, the adversary assumes that the target visits each known ROI during every epoch in the observation period. This ensures that the visits to these ROIs are reflected in the target trace, but it also sets many incorrect visits. Results are presented in Table 6.4. In our second implementation, which we call ”random sampling”, the adversary distributes the target’s visits uniformly across the known ROIs. For example, if the adversary knows the top-3

ROIs, and their estimate for the mean number of visits per user is μ , then they would sample $\frac{\mu}{3}$ visits for each of the top-3 ROIs. The corresponding epochs for each visit are sampled from the estimated time marginal. For simplicity, we assume that μ is the true mean number of visits and that the estimated time marginal is the true one. Results on raw aggregates of size $m = 1000$ are presented in Table 6.5.

Results. Table 6.5 shows that both MIAs perform poorly ($AUC < 0.63$) when the adversary uses random sampling to approximate the target trace. This suggests that random sampling fails to estimate the target trace, due to the omission of true target visits, and the inclusion of incorrect visits.

In contrast, Table 6.4 shows that KK was able to perform significantly better than random when the adversary knew more than 2 of the target’s most visited ROIs and used the greedy implementation (ex. $AUC = 0.86$ on Milan when knowing all visited ROIs). This suggests that, although the greedy implementation includes many incorrect visits, the guaranteed inclusion of some of the target’s actual visits enables membership inference to an extent. ZK, on the other hand, fails to attain 0.6 AUC. Since ZK already replaces real individual traces with synthetic traces, we hypothesize that membership inference becomes too difficult if the estimated target trace contains significantly incorrect information.

We however note that our current implementation for sampling the visits under this prior knowledge might be suboptimal and that better implementations might exist. For example, [ZZZ20] uses a synthetic target trace, using social network information and the traces of the target’s friends. We leave this exploration for future work.

Table 6.4: **Mean AUC scores with standard error for KK and ZK without time knowledge and greedy approach.** This results are computed on raw aggregates of size $m = 1000$ when the adversary only knows some of the target’s visited ROIs and employs the greedy approach of assuming that the target visits each known ROI during every epoch.

Dataset	Knock Knock				Zero Auxiliary Knowledge			
	Top 1	Top 2	Top 3	All	Top 1	Top 2	Top 3	All
CDR	0.571 ± 0.05	0.611 ± 0.05	0.647 ± 0.06	0.825 ± 0.05	0.512 ± 0.02	0.527 ± 0.01	0.514 ± 0.02	0.516 ± 0.01
Milan	0.682 ± 0.03	0.764 ± 0.03	0.822 ± 0.03	0.860 ± 0.02	0.542 ± 0.02	0.524 ± 0.02	0.542 ± 0.02	0.545 ± 0.02

Table 6.5: **Mean AUC scores with standard error for KK and ZK without time knowledge and random sampling approach.** This results are computed on raw aggregates of size $m = 1000$ when the adversary only knows some of the target’s visited ROIs and employs the random sampling approach of distributing random visits uniformly across each known ROI.

Dataset	Knock Knock				Zero Auxiliary Knowledge			
	Top 1	Top 2	Top 3	All	Top 1	Top 2	Top 3	All
CDR	0.542 ± 0.04	0.538 ± 0.03	0.531 ± 0.02	0.525 ± 0.02	0.524 ± 0.03	0.528 ± 0.02	0.515 ± 0.02	0.510 ± 0.02
Milan	0.576 ± 0.02	0.628 ± 0.03	0.614 ± 0.03	0.560 ± 0.03	0.518 ± 0.02	0.553 ± 0.02	0.568 ± 0.02	0.556 ± 0.04

6.7 Discussion

We first provide a critical analysis of the experimental results, followed by a discussion of mitigation strategies and their practicality. We then consider limitations in our methods and evaluations, before discussing how our methodology may be generalized to MIAs beyond the setting of aggregate location data.

6.7.1 Analysis of results: practical risk of MIA

In Section 6.6, we observed that our ZK MIA achieves approximately the same performance as the KK MIA across all experiments, and that both MIAs performed effectively across a range of common privacy settings. This has several important implications.

First, the ZK MIA significantly increases the attack surface of aggregate location data, since no auxiliary dataset is needed for the MIA. Although previous MIAs on aggregate location data have been successful, the strong assumption of a large auxiliary dataset prevents these adversaries from attempting the MIA in most real-life cases. The auxiliary dataset comprises sensitive user-level information collected from the same dataset that is being aggregated. However, aggregation is applied to prevent the release of personal information. Moreover, *Adv* would be restricted by the size of their auxiliary dataset, since they would not be able to perform MIAs on aggregates computed over more users than there are in their reference. In contrast, we demonstrated that our Zero Auxiliary Knowledge *Adv* can create an arbitrary number of synthetic traces upon seeing the released aggregate, without an auxiliary dataset. This offers *Adv* the flexibility to attack aggregates of any size.

Since ZK MIA requires the estimation of statistics from the aggregate, there may be concerns about its performance when the aggregate size is small. However, like previous MIAs, ZK MIA performs more effectively on smaller-scale aggregates compared to larger aggregates. This is shown in Figure 6.16 for aggregate sizes $m = 100, 250, 500, 1000$ and different privacy budgets ϵ .

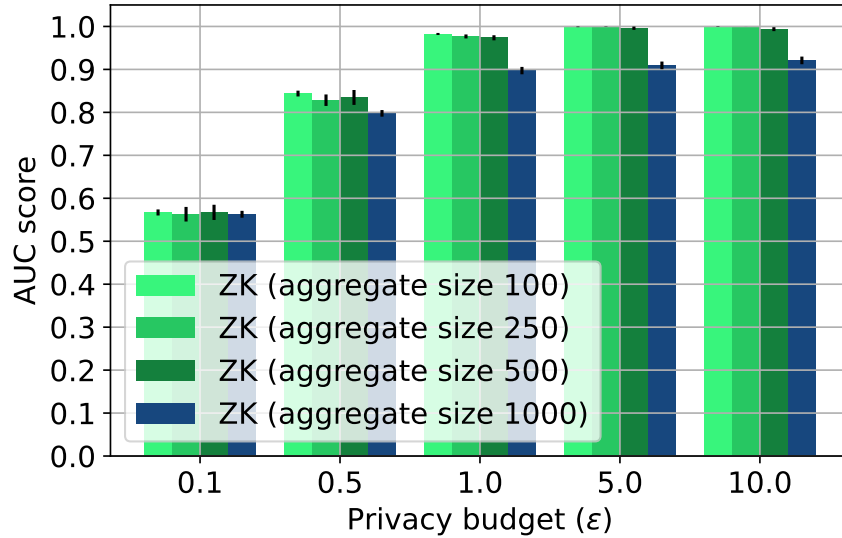


Figure 6.16: **Mean AUC scores with standard error for ZK and various aggregate sizes.** Aggregates from the Milan dataset are computed with event level ϵ -DP for varying privacy budgets ϵ and for varying sizes of aggregates.

To further understand how MIA performance scales with aggregate size m , we also consider $m > 1000$ in the next experiment. To this end, we vary $m = 100, 500, 1000, 2000, 3000$ and compare the performance of KK MIA and ZK MIA on raw ($k = 0$) and suppressed ($k = 1$) aggregates. Results on the CDR dataset are reported in Table 6.6 and results on the Milan dataset are reported in Table 6.7. $m = 3000$ was not run on the Milan dataset due size limitations.

In these settings with mild privacy protection, the attacks always succeed regardless of the value of m . We also observe a few intuitive trends. First, when raw aggregates ($k = 0$) are attacked, increasing the size of the aggregates slowly decreases the performance of the attack. On the CDR dataset, KK and ZK attain AUCs 0.999 and 1.0 for $m = 100$, which decreases to 0.919 and 0.977 for $m = 3000$. Second, when we apply suppression $k = 1$, the attacks

Table 6.6: Mean AUCs of KK and ZK MIAs on the CDR dataset with varying m .

	k=0		k=1	
m	KK	ZK	KK	ZK
100	1.000 ± 0.000	1.000 ± 0.000	0.856 ± 0.019	0.779 ± 0.041
500	1.000 ± 0.000	1.000 ± 0.000	0.961 ± 0.008	0.987 ± 0.003
1000	1.000 ± 0.000	0.999 ± 0.001	0.981 ± 0.007	0.976 ± 0.010
2000	0.997 ± 0.003	0.994 ± 0.006	0.979 ± 0.010	0.965 ± 0.013
3000	0.988 ± 0.011	0.977 ± 0.021	0.973 ± 0.011	0.938 ± 0.016

Table 6.7: Mean AUCs of KK and ZK MIAs on the Milan dataset with varying m .

	k=0		k=1	
m	KK	ZK	KK	ZK
100	1.000 ± 0.000	1.000 ± 0.000	0.756 ± 0.020	0.701 ± 0.046
500	1.000 ± 0.000	1.000 ± 0.000	0.889 ± 0.018	0.885 ± 0.025
1000	0.995 ± 0.002	1.000 ± 0.000	0.916 ± 0.015	0.919 ± 0.016
2000	0.977 ± 0.011	1.000 ± 0.000	0.981 ± 0.009	0.972 ± 0.007

initially perform poorly when the aggregate size is small. We hypothesize this to be due to a larger percentage of entries being suppressed when fewer traces are aggregated, leaving less information in the release. This effect gradually decrease as aggregate size increases. It is then counterbalanced by the first effect, that increasing the size of the aggregates slowly decreases the performance of the attack, when aggregate sizes increase. This is visible for $m \leq 1000$ in the CDR dataset. For the Milan dataset, AUC however still monotonically increases even beyond $m \leq 1000$ as the dataset is more sensitive to suppression with the average user has approximately 6 times less visits, as shown in Table 1(b).

We have also shown that MIAs on aggregate location data are more powerful than previously known. By incorporating paired sampling for training, we have demonstrated more effective MIA results on size 1000 aggregates than previously reported [PTDC20], particularly when protected by differential privacy. Our results therefore demonstrate that MIAs on aggregate location data are easily performed without auxiliary data, more effective than previously believed, and that common privacy measures fail to protect against the risk.

6.7.2 Proposed mitigations

Our results show that aggregated location data requires more stringent privacy safeguards to protect against MIAs. This is an inherently challenging task because our ZK MIA was able to succeed by using the aggregate to estimate where the population moves (space marginal), when the population moves (time marginal), and how frequently the population moves (activity marginal). However, location aggregates naturally leak this information. In fact, much of its utility is derived from these marginal statistics. Therefore, while one can mitigate the ZK MIA by perturbing the aggregate to the point where the basic mobility patterns of its population are unrecoverable, doing so may also destroy the aggregates' utility.

In light of these results, we advise that data practitioners be mindful of the parameters that they select for ϵ -DP, because DP does not guarantee sufficient protection from MIAs if the parameters are chosen too loosely. Since data practitioners often prioritize utility, it is common to pick more relaxed parameters for the privacy unit (e.g. event or user-day instead of user) and budget ϵ (e.g. $\epsilon > 1$). For example, Kohli et al. [KAB23] studied $\epsilon \in \{0.1, 0.5, 1.0\}$ at the event level in the context of aggregate O-D mobility matrices, Facebook used $\epsilon = 0.45$ at the event level when collecting data about URLs shared on the site, and Apple uses ϵ between 2 and 16 at the user-day level when collecting IOS data [Des21]. Recall that ZK and KK achieved $AUC \geq 0.9$ on the CDR dataset whenever the noise scale $\frac{\Delta}{\epsilon}$ was 2 or less. This corresponds to $\epsilon \geq 0.5$ for event level DP and $\epsilon \geq 10$ for user-day level DP with up to $\Delta = 20$ daily visits. ZK and KK therefore both achieved high discrimination on privacy settings that are in line with many real-life applications.

However, we do observe DP mitigating the MIAs when we pick sufficiently strict parameters. For example, no MIA achieved better than random performance for $\epsilon = 0.5$ in the user-day setting. We also note that we did not evaluate using the user level setting, which would achieve the strongest privacy protection. We note that the suitability of privacy parameters depends on the desired utility and sensitivity of the dataset. A stricter parameter choice is particularly relevant if the aggregate is publicly released and/or pertains to sensitive data.

Although ε -DP always offers privacy guarantees, our experimental results emphasize the importance of picking appropriate parameters. In particular, we observed that event-level DP was largely ineffective in preventing MIAs from both strong and weak adversaries. We instead encourage the use of user-day or user-level DP with carefully selected privacy budgets ε to mitigate the practical threat of an MIA.

6.7.3 Limitations

We have so far taken the Knock-Knock MIA to refer to the Subset of Locations setting [PTDC17]. We now address why we do not consider the Knock-Knock Participation in Past Groups [PTDC17] threat model in this chapter. Under the Participation in Past Groups setting, the adversary has access to a set of past aggregates $\{\bar{A}^{\mathcal{U}_1}, \dots, \bar{A}^{\mathcal{U}_N}\}$, collected over the same ROIs $\mathcal{S}$ as the released aggregate $\bar{A}^{\mathcal{U}}$. Moreover, Adv is assumed to know the membership status of the target u^* in each of these aggregates. That is, Adv knows whether or not $u^* \in \mathcal{U}_i$ for all $i = 1, \dots, N$. This last assumption is crucial because Adv directly uses $\{\bar{A}^{\mathcal{U}_1}, \dots, \bar{A}^{\mathcal{U}_N}\}$ as their training data for the membership classifier in this setting. This is unrealistic for multiple reasons. First, to train an effective membership classifier, there would need to be hundreds of labeled aggregates to have sufficient training data. More importantly, there would be no reason for the membership status of an individual within an aggregate to be released in practice. We argue that the only plausible scenario in which Adv would know the membership status of each aggregate is if they created the aggregates themselves. This reduces to the Subset of Locations setting that we have assumed in this chapter.

Delaunay triangulation We note that the Delaunay triangulation of the ROIs, $DT(\mathcal{S})$, is the only non-probabilistic parameter used to generate synthetic traces for the ZK MIA. The triangulation only depends on the locations of the ROIs, which we have so far assumed to be shared as part of the aggregate release (Section 6.2.5). We believe this to be a realistic assumption, as omitting the locations of the ROIs would strongly diminish the utility of aggregate location data. Nonetheless, there might exist cases where released location aggregates

do not relay the positions of the ROIs. For example, Google binned ROIs into categories, ex. restaurants, parks, and hospitals, when publicly releasing their mobility report during COVID <https://www.google.com/covid19/mobility/>. In this setting, the adversary would proceed without knowing where the ROIs are situated with respect to one another. The privacy risk under this setting is not known, and we identify it as an area of future research.

Removing the time knowledge Similarly, there might exist cases where the adversary knows the ROIs that were visited by the target (ex. home and work), but not the visitation times. We show in Section 6.6.5 that only knowing the visited ROIs substantially reduces the effectiveness of both MIAs.

Small aggregate sizes. ZK MIA also requires that we estimate statistical parameters from the released aggregate. It may be difficult to estimate these precisely if the aggregate size is small or if the collected location data is not regular. However, we still observe strong performance by ZK MIA on both datasets for small aggregate sizes (see results in Section 6.7.1).

Pattern in distribution. Furthermore, aggregate location data collected over large metropolitan populations are known to obey high regularity across different cities and time periods. These patterns include log-normal activity distributions [SBC⁺13, SMS⁺08, FHdM21] and periodic "circadian rhythm" time marginals [CBT⁺13, SQBB10, SMS⁺08, FHdM21]. This suggests that our statistical parameter estimation should be highly transferable across sufficiently regular datasets. However, we acknowledge that there are scenarios where the observed population is not regular (e.g. taxi drivers).

6.7.4 Generalizations to MIAs in other settings

In this chapter, we have proposed a new methodology to perform membership inference attacks on aggregate data, by training the attack on synthetic records, generated from the released aggregate. We believe that this approach can be adapted for MIAs in settings beyond aggregate

location data. Our methodology can be broken down into two main steps: 1) extracting noiseless global statistics from the released aggregate, 2) use these statistics to create individual-level records to train the MIA.

In the setting of location data, the relevant statistics pertain to the mobility trends of large-scale human populations [SBC⁺13, SMS⁺08, FHdM21, CBT⁺13, SQBB10, SMS⁺08], and individual location [KG17, KTVG18, OSRY18, KLbv07, LHK⁺09] which have both been well established in the literature. This facilitates both steps of our methodology, as we know in advance what location data should look like at both the global and individual level.

Although the trends will be distinct from aggregate location data, aggregate releases for other types of data will generally reveal global statistics. For instance, categorical tabular data is modeled by discrete random variables, whereas location data is modeled by continuous random variables, and approximated by high-dimensional discrete data. Our methods for denoising and debiasing statistics from differentially private and suppressed aggregates are however not specific to location data, and should generalize to other data releases. Regarding the second step, using the statistics to create individual records for training, the probabilistic method used for our ZK MIA, drawing from the Delaunay triangulation and the relevant marginal distributions, is partially specific to location data. One would thus need to carefully consider the statistical properties of the type of data to create high quality individual records.

6.8 Conclusion

Aggregate location data is widely shared and used by governments [Sav21, Hop21, Oli21], companies [App17, ABC⁺20, HD21], and researchers [TC23, JWA⁺20, KAB23] because of its insights into human behaviour and its presumed security against reidentification.

In this chapter, we demonstrated that aggregate location data is susceptible to MIAs by realistic adversaries, who only know some of their target’s location history. With ZK MIA, we introduced the first MIA on aggregate location data that does not require an auxiliary dataset. We accomplished this by generating appropriate synthetic traces, using statistics that are estimated

from the released aggregate. We also equipped our parameter estimation with techniques that automatically correct for bias and noise from popular privacy mechanisms like suppression of small counts and ε -DP noise.

We then showed that MIAs on aggregate location data are significantly improved by incorporating a paired sampling technique, which helps isolate the effect of the target trace within a high dimensional aggregate. Hence, the vulnerability of aggregate location data is further heightened by these improved attacks.

Our evaluations over two large datasets demonstrate that, despite the absence of an auxiliary dataset, ZK MIA performs as well as the state-of-the-art KK MIA, with both MIAs achieving high discrimination when commonly used privacy settings are applied. ZK MIA remains effective in realistic privacy settings, even when only a small fraction (10%) of the target trace is known. These results emphasize the need for strict differential privacy guarantees on released aggregate location data.

Taken together, our findings show that membership inference attacks are not merely a theoretical privacy threat posed by unrealistically strong adversaries, but also a realistic threat to contend with in practice. The next chapter will demonstrate that this is true, even when the release is made more abstract, through the use of a machine learning model.

Chapter 7

Correlation inference attacks against machine learning models.

The work presented in this chapter has been published as: Crețu, A. M.^{*}, Guépin, F.^{*} & de Montjoye, Y. A. (2024, July). Correlation inference attacks against machine learning models. In Science Advances (vol. 10, issue 28, eadj9260). <https://www.science.org/doi/full/10.1126/sciadv.adj9260>

Statement of contributions: Ana-Maria Crețu and I are joint first authors on this paper. We collaborated in the design of the methods, experiments, writing of the code and the paper. We equally contributed to the writing of the code, running experiments, production of the figures and writing of the paper.

7.1 Introduction

Machine learning (ML) is widely used today by companies, governments, and researchers to learn from data how to automate processes and make decisions. Machine learning models are, for example, used for medical diagnosis [QKS⁺19], language translation [WSC⁺16], speech recognition and synthesis [aA17], chatbots, content moderation [Rek], drug discov-

ery [CEW⁺18], content retrieval [GARL16], and personalized services.

In light of previous chapters, ML models as data release mechanisms can be seen as generic functions fine-tuned on finite training datasets, to answer specific questions [GBC16], e.g., detecting early-stage cancer in young patients. The understanding of the relationship between a model and the specific dataset it was trained on is both a fundamental question, from a learning theory perspective, and a very important one in practice, for instance to evaluate the circumstances in which a model can be used. On one hand, the two extreme cases would be “remembering” nothing and “remembering” everything from the training dataset. The first extreme would be a model that always outputs the same prediction, without taking into account the input. The second extreme would be a model that encodes the entire dataset into the model, and always outputs the label of the closest record of its training dataset. The ideal case on the other hand would be a model that always outputs the perfect probability based on the training dataset, e.g., $\mathcal{M}(x) \simeq P(Y|X = x)$. In practice, models won’t be ideal or extreme, but somewhere in between. A model that has a smaller number of parameters than records in his training dataset, suggest that some level of learning happens. In contrast, models with millions of parameters have become the standard for large language models [AK21]. This suggests that models might remember- even encode- quite a lot about their training dataset.

The intuitive way to study the relationship between a model and its training dataset could be to track how the model “came to be”, i.e., how the final parameters of the model are computed, step by step, from the training dataset. While this is doable in some specific cases such as linear regression, doing it today is increasingly difficult. As we have seen in the previous chapter, studying the different sources of randomness from a model, learning from a dataset involves performing a complex, iterative, and stochastic procedure, over random subsets of the dataset (so-called mini-batches) to update sometimes billions of parameters using gradient descent. In practice, the resulting model is closer to a “black box”: a set of parameters whose precise relationship to the training dataset is largely untraceable.

Instead, trained models have been studied in an adversarial manner, where an adversary is trying to reveal “remembered” information about the training dataset of a model [VBE18].

Inference attacks, taking the form of games, have been used to see if specific pieces of information can be extracted from a trained model [AMS⁺15, FLJ⁺14, FJR15, SSSS17, CTW⁺21, BCH22].

So far, those inference attacks have however mostly focused on inferring the presence or absence of a given record in the dataset, a membership inference attack (MIA) [HSR⁺08, SSSS17]. In an MIA, the *adversary* aims to infer, better than random, whether a specific record was part of a dataset used to train the model. A large body of work has used MIAs to build a robust understanding of the conditions under which information about individual records is more likely to leak, e.g. when a model is overfitted or trained on a small dataset [SZH⁺18, YGFJ18, NSH19, TLG⁺19, JE19, LF20, CCTCP21, CCN⁺22].

In this chapter, we exclude the hypothetical case where an MIA would achieve a perfect accuracy, leading to the whole dataset to be reconstructed. Indeed, current MIAs are not perfect and cannot achieve such a task. We also do not believe that near-future attacks are likely to reach close to perfect accuracy in general either. Membership is, however, only one of the characteristics of the training dataset that can be “remembered” by a model, and because datasets are hard to reconstruct, other attack could aim for other characteristics.

In particular, there has been little work studying whether models encode dataset-level characteristics. An important line of research, for instance, aims to audit machine learning models to study biases or to better understand the range of values within which a model can operate, a crucial question in practice [BG18]. From a privacy perspective, dataset-level characteristics could reveal sensitive information [AMS⁺15, GWY⁺18, ZTO21]. For instance, a scoring model for depression could remember that patients living in the inner city are more likely to have used illegal substances. The answer to the question of what a model does or even should remember from its training dataset is not simple. One could argue that the ideal “fully generalizable” model should learn nothing more than the intended probability of the output label given an input sample $\mathcal{M}(x) \simeq P(Y|X = x)$. Models today and in the near future, however, result from minimizing a loss function computed over a specific finite dataset of samples from the underlying dataset distribution $P(X, Y)$ [GBC16]. The dataset distribution $P(X, Y)$ and, through marginalization, $P(X)$, may very well end up influencing the final model in practice.

Contribution. In this work, we study a new type of leakage in ML models: the leakage of correlations between input variables of a tabular training dataset by proposing what is, to our knowledge, the first correlation inference attack (CIA) against ML models. We argue that (a) the leakage of correlations by an ML model can be seen as a privacy violation which goes beyond the model’s intended purpose and can lead to individual harm (see Section 7.3 for examples), (b) correlations can be used as building blocks for other attacks on individual privacy such as attribute inference attacks, in particular enabling weaker adversaries (see Section 7.2.4), and (c) CIAs are different from previously studied property inference attacks (PIA), since PIAs target something else and rely on strong assumptions that make them inapplicable to CIAs (see Section 7.5 for a detailed comparison).

We propose two different correlation inference attacks. First, a model-less attack that uses knowledge of the correlations between the input variables X and the output variable Y , which we call correlation constraints, to predict the correlation between two input variables $\rho(X_1, X_2)$. We argue that the correlation constraints are typically known to or inferable by the adversary. Our attack exploits the spherical parametrization of correlation matrices, imposing bounds on the correlation coefficients, to make an informed guess. This first attack also acts as a strong baseline, allowing us to correctly quantify the leakage caused by the machine learning model, and not by the adversary’s prior knowledge about the dataset. We then propose a second, model-based attack which exploits access to an ML model trained on the dataset of interest. Our model-based attack uses a meta-classifier relying only on synthetic data, dramatically reducing the strength of the adversary compared to the literature. To generate synthetic data, we combine Gaussian copula-based generative modeling as our prior with carefully adapted procedures for sampling correlation matrices under constraints.

We then study the performance of our model-based attack on synthetic data under three different attack scenarios. To illustrate the scenarios, we consider the same example as before, the scoring model about depression released by an hospital. By order of increasing adversary knowledge, we consider an adversary that knows: (1) the correlation constraints relating only to the target variables, which would mean the adversary knows the correlation between the use of illegal substances and depression, as well as the correlation between the living place and

depression, (2) all the input variables, which would mean how every single input variable would be correlated with the depression score or (3) the correlations between all the variables except for the target correlation, $\rho(X_1, X_2)$. In this last case, it would mean that only the correlation between the living place and the use of substance would be unknown from the adversary. Across scenarios, we first show our model-based attack to consistently strongly outperform the model-less attack, showing that ML models such as Logistic Regressions (LR) and Multilayer Perceptrons (MLP) leak information about the correlations between their input variables X . Second, we show how mitigations such as limiting the number of queries that can be made to the model or their precision fail to prevent the attack. Third, we show that, unsurprisingly, ensuring that the model is differentially private (DP) does not prevent our attack and extend our attack to and validate it on three real-world datasets. We finally argue that dataset-level statistics can be building blocks for other attacks, in particular enabling weaker adversaries. To illustrate our argument, we propose a novel attribute inference attack exploiting the correlations extracted from the model, and show it to significantly outperform previous attacks on real-world datasets.

We make our source code allowing to reproduce the results of our experiments available at: <https://github.com/computationalprivacy/ml-correlation-inference>.

7.2 Results

7.2.1 Correlation inference threat model

We consider a machine learning model $\mathcal{M}_T$ trained on a *target dataset* D_T consisting of m records $z^1, \dots, z^m$. Each z^i is a sample of $n - 1$ input variables $X_1, \dots, X_{n-1}$ and one output variable Y . We assume variables to be continuous and real-valued and denote by $F_i(x) = \Pr(X_i \leq x)$ the one-way marginal of the i -th input variable and by $F_n(y)$ the one-way marginal of the output variable. For simplicity, we consider $\mathcal{M}_T$ to be a binary classifier whose goal is to infer if $Y > c$, where c is a task-specific threshold such as the average or the median value.

Our attack can, however, be easily generalized to multi-class classification and regression tasks.

Our adversary aims to infer the Pearson correlation coefficient $\rho(X_1, X_2)$ between two input variables X_1 and X_2 given access to the machine learning model $\mathcal{M}_T$. Also called *linear correlation*, the Pearson correlation coefficient measures the strength of the linear association between two variables and is widely used by researchers and data scientists across many domains [Wal19, ZDXF16]. We call $\rho(X_1, X_2)$ the *target correlation* and X_1 and X_2 the *target variables*.

We make two assumptions on the adversary knowledge on the data distribution:

1. **Knowledge of one-way marginals.** First, we assume the adversary to have knowledge of the one-way marginals of all the variables $F_i(x), i = 1, \dots, n-1$ and $F_n(y)$. These are indeed typically released, e.g., in research papers or to comply with legal requirements such as the European Union AI Act [Com20] and their knowledge is a common assumption in the literature [FLJ⁺14, FJR15, ZTO21].
2. **Knowledge of some correlations.** Second, we assume the adversary to have access to the linear correlations between some of the variables. More specifically, we define the *partial knowledge* of the adversary as a set P of pairs such that for every $(i, j) \in P$ the adversary has the knowledge of the ground-truth values $\rho(X_i, X_j)$ (with $X_n = Y$). As $\rho(X_i, X_j) = \rho(X_j, X_i)$, by symmetry we assume that P consists of ordered pairs, i.e., if $(i, j) \in P$, then necessarily $i < j$.

We consider three different attack scenarios:

- (S1) The adversary knows the correlations between the target variables and the output variable, namely $\rho(X_1, Y)$ and $\rho(X_2, Y)$, i.e., $P = \{(1, n), (2, n)\}$, but not $\rho(X_i, X_j)$ for $1 \leq i < j \leq n-1$ nor $\rho(X_i, Y)$ for $i \geq 3$.
- (S2) The adversary knows the correlations between all the input variables and the output variable: $(\rho(X_i, Y))_{i=1, \dots, n-1}$, i.e., $P = \{(i, n) : i = 1, \dots, n-1\}$, but the adversary does not know $\rho(X_i, X_j)$ for $1 \leq i < j \leq n-1$.
- (S3) The adversary knows all the correlations between the variables, except for the target correlation $\rho(X_1, X_2)$. More specifically, the adversary knows $\rho(X_i, X_j), \forall 1 \leq i <$

$j \leq n$ such that $(i, j) \neq (1, 2)$, i.e., $P = \{(i, j) : 1 \leq i < j \leq n\} \setminus \{(1, 2)\}$.

Note that the three scenarios are identical for $n = 3$ variables.

We will henceforth call the known coefficients the *correlation constraints*.

S2 is the default scenario considered in this chapter. This choice is motivated by the fact that the relationship between the input and output variables – encoded through $\rho(X_i, Y)$ – is unlikely to be considered a secret, since the model’s goal is to infer Y given X . We furthermore show in Section 7.3 how these correlation constraints can be extracted from the model.

S1 is a weak adversary, motivated by our goal to measure the *leakage from the model*, rather than the leakage derived from constraints imposed on X_1 and X_2 via their association with the output variable Y . Indeed, our analysis shows that the correlation coefficients $\rho(X_1, Y)$ and $\rho(X_2, Y)$ restrict the range of values that the target coefficient $\rho(X_1, X_2)$ can take, without the adversary even accessing the model.

S3 is the strongest, “worst-case” scenario, corresponding to an adversary having complete knowledge of the correlations, except for the target. We mostly study this scenario in order to evaluate how much information a very strong adversary would be able to infer.

We consider two different assumptions on the adversary access to the target machine learning model $\mathcal{M}_T$.

1. **Model-less attack.** Under this attack scenario, the adversary does not have access to the target model $\mathcal{M}_T$. This scenario mostly serves as a stronger than random baseline allowing to measure the adversary’s uncertainty on the target correlation under the prior knowledge. It is necessary to consider this scenario properly evaluate the leakage from the model. A method for this scenario might also become an attack in situations where the model is not released, yet information about the marginals and the correlations between the input variables and the target variables is made available, e.g., as part of a research study.

2. **Model-based attack.** Under this attack scenario, we assume the adversary (1) knows the model architecture and training details, e.g., number of training epochs for a neural network, allowing them to train from scratch a similar model and (2) has black-box query access to the target model $\mathcal{M}_T$, allowing to retrieve the output probabilities for each class $\mathcal{M}_T(x)$ on inputs x .

7.2.2 Attack methodology

Overview. We frame the *correlation inference* task as a classification task by dividing the range of correlations $[-1, 1]$ into N_B bins of equal length. The adversary’s goal is to infer the correlation bin to which $\rho(X_1, X_2)$ belongs. When $N_B = 3$, the bins are $[-1, -1/3)$, $[-1/3, 1/3)$ and $[1/3, 1]$, which we refer to as *negative*, *low* and *positive* correlations, respectively. We opt for a classification rather than a regression task because it is easier to analyze the attack performance by comparing its accuracy with a random guess baseline (at $1/N_B$). Unless otherwise specified, we use $N_B = 3$ classification bins in our empirical evaluation, as the correlation ranges can be easily interpreted. We also discuss in Section 7.2.3 and Section 7.2.3 results of our attack for $N_B = 5$ bins.

At a high level, our methodology analyzes the behavior of the target model through the confidence scores returned on specific inputs, searching for patterns allowing to infer the correlations between input variables. More specifically, we train a meta-classifier to infer the correlation $\rho(X_1, X_2)$ between two variables of interest in a dataset D based on features extracted from a model $\mathcal{M}$ trained on D . Training the meta-classifier requires a large number of models $\mathcal{M}$ derived from datasets D spanning a wide variety of values for the unknown correlations $\rho(X_i, X_j)$, where $(i, j) \notin P$, while matching the correlation constraints $\rho(X_i, X_j)$ for every $(i, j) \in P$. Generating datasets satisfying this requirement on their correlations is a key technical challenge our work addresses.

Correlation matrices. To formalize this requirement on the linear relationships between the n variables in a dataset, we use the concept of *correlation matrix*. This is the $n \times n$ real-

valued matrix C of Pearson correlation coefficients between all the pairs of variables: $c_{i,j} = \rho(X_i, X_j), i, j = 1, \dots, n$. Note that we use lowercase notation to denote matrix coefficients. Let $\mathcal{C}$ be the set of *valid correlation matrices*. $\mathcal{C}$ consists of all $n \times n$ real-valued matrices satisfying the following properties:

- (P1) All elements are valid correlations, i.e., real values between -1 and 1 : $-1 \leq c_{i,j} \leq 1, \forall 1 \leq i, j \leq n$.
- (P2) Diagonal entries are equal to 1 , i.e., there is perfect correlation between a variable and itself: $c_{i,i} = 1, \forall 1 \leq i \leq n$.
- (P3) The matrix is symmetric, i.e., the correlation between X_i and X_j is the same as the correlation between X_j and X_i : $c_{i,j} = c_{j,i}$.
- (P4) The matrix is positive semi-definite: $x^T C x \geq 0, \forall x \in \mathbb{R}^n$.

The **spherical parametrization** of correlation matrices [PB96] provides a principled and effective approach for sampling valid correlation matrices C . This approach crucially relies on the Cholesky decomposition of $C = BB^T$ which, as we describe next, can be used to generate valid correlation matrices by sampling its correlation coefficients one-by-one, conditionally on the correlation constraints. More specifically, since C is positive semi-definite, it has a Cholesky decomposition $C = BB^T$, where B is lower triangular with non-negative diagonal entries: $b_{i,j} = 0, 1 \leq i < j \leq n$ and $b_{i,i} \geq 0, i = 1, \dots, n$. Furthermore, the coefficients of matrix B can be expressed using spherical coordinates, as follows:

$$B = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ \cos \theta_{21} & \sin \theta_{21} & \ddots & \ddots & \vdots \\ \cos \theta_{31} & \cos \theta_{32} \sin \theta_{31} & \sin \theta_{32} \sin \theta_{31} & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \cos \theta_{n1} & \cos \theta_{n2} \sin \theta_{n1} & \cos \theta_{n3} \sin \theta_{n2} \cdot \sin \theta_{n1} & \cdots & \prod_{j=1}^{n-1} \sin \theta_{nj} \end{pmatrix} \quad (7.1)$$

This is because the diagonal of C consists of ones, i.e., $1 = c_{i,i} = (BB^T)_{i,i} = \sum_{j=1}^n b_{i,j}^2 = \sum_{j=1}^i b_{i,j}^2$, which means that each vector $(b_{i,1}, \dots, b_{i,i}) \in \mathbb{R}^i$ has an $\mathbb{L}_2$ norm of 1 and can

therefore be expressed using spherical coordinates, i.e., $\exists \theta_{i,j} \in [0, \pi), 1 \leq i < j < n$. Note that as expected for the symmetric matrix C , only $\frac{n \times (n-1)}{2}$ parameters, called angles, are sufficient to characterize the set of valid correlation matrices.

A key property exploited by both our model-less and model-based attacks is that the first column of B is identical to the first column of C . By reordering—without loss of generality—the variables as $Y, X_1, \dots, X_{n-1}$, all the correlation constraints appear on the first column as free parameters under scenarios **S1** and **S2**. We will use this property to sample the other correlation coefficients conditionally on the correlation constraints (Algorithm 8 and 9). We will further show how sampling can be done even in scenario **S3** (Algorithm 10), where the correlation constraints cannot be expressed using only the free parameters of the first column. We start by describing the model-less attack, to build intuition into how this property is useful for sampling correlation matrices under constraints.

Model-less attack. Our *model-less attack* leverages the intuition that the adversary’s knowledge of some correlations allows to derive theoretical limits on the range spanned by the other values. This allows the adversary to make a better than random guess about the target correlation $\rho(X_1, X_2)$ without access to the model. The model-less attack acts as a strong baseline for any attack using access to the model, allowing to correctly quantify the leakage caused by the model.

Given the correlation constraints $(c_{i,j})_{(i,j) \in P}$, we call *matching set* C_P the set of valid correlation matrices satisfying them: $C_P = \{C' \in \mathcal{C} : \forall (k,l) \in P, c'_{k,l} = c_{k,l}\}$. Due to the correlation constraints, the range of values R that can be taken by $c_{1,2}$ is a subinterval of $[-1, 1]$: $R = [\inf_{C' \in C_P} c'_{1,2}, \sup_{C' \in C_P} c'_{1,2}] := [m_1, m_2]$. Our model-less attack outputs the majority bin over this interval. To avoid introducing biases in the evaluation, we assume the uniform prior over this interval and return the classification bin which is covered in the largest proportion by $[m_1, m_2]$. We make our choice of classification bin as follows:

With the following definition of N_B , the classification bins, as the following intervals:

$$\begin{aligned} & \left[\frac{2(b-1) - N_B}{N_B}, \frac{2b - N_B}{N_B} \right), \text{ for } b = 1, \dots, N_B - 1 \\ & \left[\frac{2(b-1) - N_B}{N_B}, \frac{2b - N_B}{N_B} \right], \text{ for } b = N_B \end{aligned} \quad (7.2)$$

When predicting the majority bin over the interval $[m_1, m_2]$, we can distinguish between those three cases:

- (C1) The interval is fully included inside a bin, i.e., $(2(b-1) - N_B)/N_B \leq m_1 \leq m_2 < (2b - N_B)/N_B$ for some $b = 1, \dots, N_B$. We predict the bin b .
- (C2) The interval $[m_1, m_2]$ partially covers two bins, but none entirely, i.e., $m_1 < (2b - N_B)/N_B \leq m_2 < (2(b+1) - N_B)/N_B$, for some $b = 1, \dots, N_B - 1$. We predict the bin b if it has higher coverage than $b+1$ (i.e., if $(2b - N_B)/N_B - m_1 > m_2 - (2(b+1) - N_B)/N_B$), and $b+1$ otherwise.
- (C3) At least one bin is fully covered by the interval $[m_1, m_2]$. We predict one of the bins that we sample uniformly at random.

As an example, consider the case $n = 3$ and an adversary having knowledge of $\rho(X_1, Y)$ and $\rho(X_2, Y)$ (corresponding to $P = \{(1, 2), (1, 3)\}$). We assume, for simplicity, that the correlation matrix is computed over Y, X_1, X_2 in this order. Developing the computation $C = BB^T$ yields the following correlation coefficients:

$$\begin{aligned} c_{2,1} &= \cos \theta_{2,1} = \rho(X_1, Y) \\ c_{3,1} &= \cos \theta_{3,1} = \rho(X_2, Y) \\ c_{3,2} &= \cos \theta_{2,1} \cos \theta_{3,1} + \sin \theta_{2,1} \sin \theta_{3,1} \cos \theta_{3,2} = \rho(X_1, X_2) \end{aligned} \quad (7.3)$$

Since $\theta_{2,1}$ and $\theta_{3,1}$ are fixed, the only degree of freedom left on the unknown $\rho(X_1, X_2)$ is $\theta_{3,2}$. Any choice of $\theta_{3,2} \in [0, \pi)$ yields a valid correlation matrix. The unknown $\rho(X_1, X_2)$ can thus take any value between $\cos \theta_{2,1} \cos \theta_{3,1} - \sin \theta_{2,1} \sin \theta_{3,1} = \cos(\theta_{2,1} + \theta_{3,1})$ and

$\cos \theta_{2,1} \cos \theta_{3,1} + \sin \theta_{2,1} \sin \theta_{3,1} = \cos(\theta_{2,1} - \theta_{3,1})$, but cannot fall outside the interval determined by these endpoints. Depending on the values of the constraints, this interval might be much smaller than $[-1, 1]$, thereby reducing the uncertainty of the adversary.

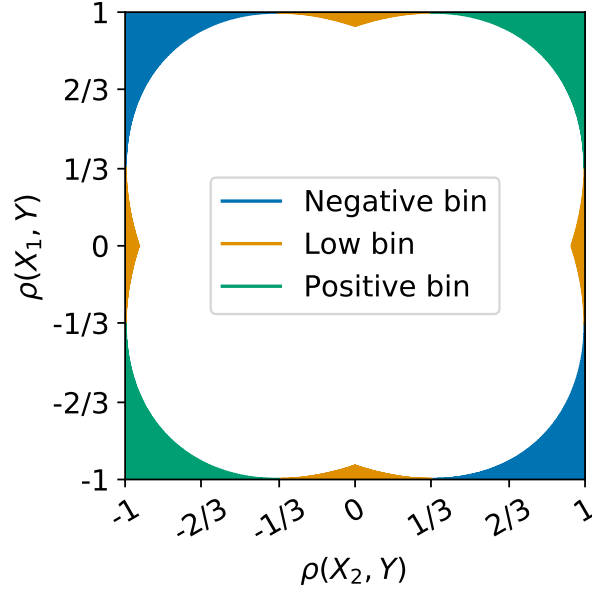


Figure 7.1: **Illustration of correlation constraints $(\rho(X_1, Y), \rho(X_2, Y))$ that lead to only one possible bin for the unknown correlation $\rho(X_1, X_2)$.** Here, we show results for $N_B = 3$ possible bins.

Fig. 7.1 shows that some constraints yield a theoretical interval R which is strictly included in one of the $N_B = 3$ classification bins. For such constraints, the adversary can infer the target correlation bin with certainty. These constraints are of the two types. First, constraints that are close to 1 in absolute value result in high $\rho(X_1, X_2)$ belonging to the positive, green bin, when they have the same sign and to the negative, blue bin, when they have opposite signs. Second, when one of the constraints is close to 0 while the other is close to 1 in absolute value, $\rho(X_1, X_2)$ is necessarily in the low, orange bin.

Analysis of the regions We analyze in detail the sets of constraints $(\rho(X_1, Y), \rho(X_2, Y))$ satisfying that there is only one possible bin for $\rho(X_1, X_2)$. We consider $0 \leq \theta_1, \theta_2 \leq \pi$ such that $\rho(X_1, Y) = \cos \theta_1$ and $\rho(X_2, Y) = \cos \theta_2$. As a reminder, the $N_B = 3$ classification bins are $[-1, -1/3)$, $[-1/3, 1/3)$, and $[1/3, 1]$ and the interval attainable by $\rho(X_1, X_2)$ is $[\cos(\theta_1 + \theta_2), \cos(\theta_1 - \theta_2)]$.

- **Negative bin:** $\cos(\theta_1 - \theta_2) \leq -\frac{1}{3}$. It follows that $\theta_1 - \theta_2 \geq \arccos(-\frac{1}{3})$ if $\theta_1 \geq \theta_2$ and $\theta_2 - \theta_1 \geq \arccos(-\frac{1}{3})$ otherwise.
- **Low bin:** $-\frac{1}{3} \leq \cos(\theta_1 + \theta_2) \leq \cos(\theta_1 - \theta_2) < \frac{1}{3}$. We distinguish 4 subcases:
 - (I) $\theta_1 \geq \theta_2$ and $0 \leq \theta_1 + \theta_2 \leq \pi$. It follows that $\theta_1 + \theta_2 \leq \arccos(-\frac{1}{3})$ and $\theta_1 - \theta_2 > \arccos(\frac{1}{3})$.
 - (II) $\theta_1 < \theta_2$ and $0 \leq \theta_1 + \theta_2 \leq \pi$. It follows that $\theta_1 + \theta_2 \leq \arccos(-\frac{1}{3})$ and $\theta_2 - \theta_1 > \arccos(\frac{1}{3})$.
 - (III) $\theta_1 \geq \theta_2$ and $\pi \leq \theta_1 + \theta_2 \leq 2\pi$. It follows that $\theta_1 + \theta_2 \geq \pi + \arccos(\frac{1}{3})$ and $\theta_1 - \theta_2 > \arccos(\frac{1}{3})$.
 - (IV) $\theta_1 < \theta_2$ and $\pi \leq \theta_1 + \theta_2 \leq 2\pi$. It follows that $\theta_1 + \theta_2 \geq \pi + \arccos(\frac{1}{3})$ and $\theta_2 - \theta_1 > \arccos(\frac{1}{3})$.
- **Positive bin:** $\frac{1}{3} < \cos(\theta_1 + \theta_2)$ It follows that $\theta_1 + \theta_2 < \arccos(\frac{1}{3})$ if $\theta_1 + \theta_2 \leq \pi$ and to $\theta_1 + \theta_2 > \pi + \arccos(-\frac{1}{3})$ if $\pi \leq \theta_1 + \theta_2 \leq 2\pi$.

Impact of the number of samples. This interval can be computed analytically or empirically, by sampling a sufficient number of correlation matrices satisfying the correlation constraints. We here use the latter approach, as our model-based attack also requires generating valid correlation matrices satisfying the constraints given by the adversary knowledge.

Throughout the experiments, for a fair comparison between the model-less and model-based attacks, we have used the same number of samples for the model-less and model-based attack, K . We have opted for relatively large values of K , in the order of thousands (see Section C.5 for the values used). This is because the model-based attack, being machine learning-based, needs more samples to converge. We here examine how the performance of the model-less attack varies with the number of samples K . We compute the accuracy of the model-less attack over $T = 1000$ correlation matrices of $n = 3$ variables using $K \in \{1, 2, 5, \dots, 100, 200, 500, 1000\}$.

Figure 7.2 shows that the accuracy of the model-less attack converges very quickly as a function of the number of samples. This is because the model-less attack predicts which of the $N_B = 3$

bins is most covered by the segment $[\min(samples), \max(samples)]$, as described in Section 7.2.2 and this prediction is very robust to the number of samples used.

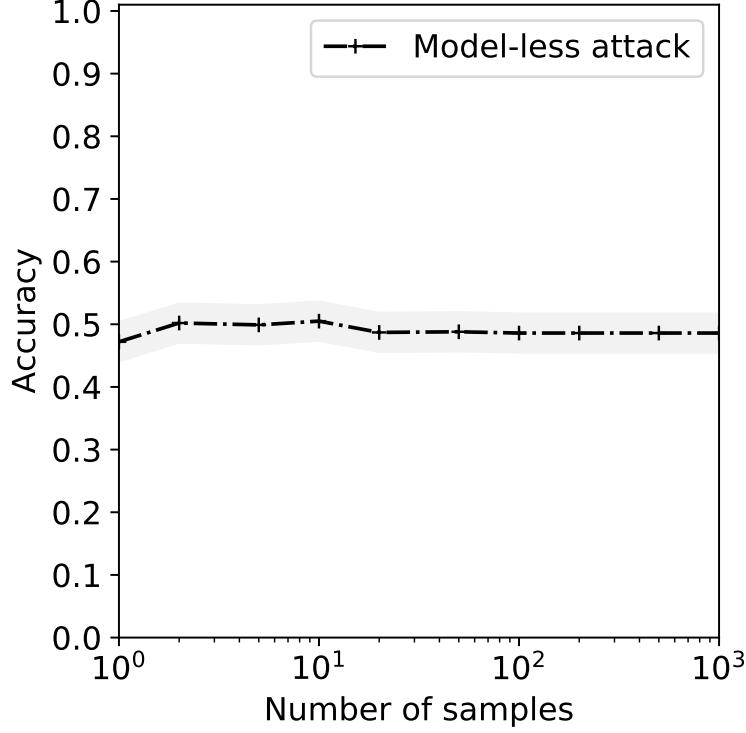


Figure 7.2: **Impact of the number of samples K on the model-less attack performance.** We report the accuracy (with 95% confidence interval) over 1000 target models.

Generating correlation matrices under constraints. To generate valid correlation matrices satisfying the correlation constraints, we adapt an algorithm developed by Numpacharoen and Atsawarungruangkit [NA12]. The goal of the original algorithm, (which can be found in Appendix A.1), is to sample valid correlation matrices (under no constraints). The insight of the authors is that each correlation coefficients $c_{i,j}$ can be expressed as $c_{i,j} = m_{i,j} + \cos \theta_{i,j} l_{i,j}$, where $m_{i,j}$ and $l_{i,j}$ only depend on $\theta_{p,q}$ for $1 \leq p \leq i$ and $1 \leq q < j$. Algorithm in Appendix A.1 uses this insight to generate valid correlation matrices by sampling the correlations in order, from top to bottom ($i = 1, \dots, n$) and from left to right ($j = 1, \dots, n$), uniformly within the bounds $m_{i,j} \pm l_{i,j}$ derived from the values previously sampled. The formulas on $m_{i,j}$ and $l_{i,j}$ can be derived (supposing without loss of generality that $i \geq j$) by developing the computation of $c_{i,j} = (BB^T)_{i,j} = B_{i,1:j-1}(B_{j,1:j-1})^T + b_{i,j}b_{j,j}$ and setting $m_{i,j} = \sum_{p=1}^{j-1} (\cos \theta_{i,p} \cos \theta_{j,p} \prod_{q=1}^{p-1} \sin \theta_{i,q} \sin \theta_{j,q})$ and $l_{i,j} = \prod_{q=1}^{j-1} \sin \theta_{i,q} \sin \theta_{j,q}$.

We adapt this algorithm to our problem by noticing that the coefficients of the first column of B ($\cos \theta_{i,1}$ for $i = 2, \dots, n-1$) are identical to those of C and are free parameters. We set these coefficients equal to the correlation constraints and sample the others uniformly within their bounds. We propose two adaptations of the original algorithm for attack scenarios **S1** and **S2**, Algorithm 9 and Algorithm 8, respectively. Note that correctly sampling the correlation matrices without biasing the values requires carefully re-ordering the variables before and after the sampling, an intricate operation that we detail in Section 7.4.1. We further propose an algorithm to sample the unknown value $\rho(X_1, X_2)$ for attack scenario **S3**, when all the other correlations are known (Algorithm 10).

Model-based attack. We finally develop a *model-based attack*, which trains a meta-classifier to infer the bin to which $\rho(X_1, X_2)$ belongs on a labelled dataset of model and correlation pairs $(M(D), \rho(X_1(D), X_2(D)))$. We create this labelled dataset by training models on synthetic datasets generated using the Gaussian copula generative model [Skl59]. We select the Gaussian copula model because it is parameterized by one-way marginals and a covariance matrix, allowing us to generate synthetic datasets that match the adversary’s knowledge about the target dataset D_T , while spanning the entire range of values attainable by the unknown correlations. Using our notation, the Gaussian copula model is parameterized by (1) the marginals of the variables $(F_i)_{1 \leq i \leq n}$ and (2) a positive semi-definite covariance matrix Σ yielding the following joint distribution:

$$\Pr(X_1 \leq x_1, \dots, X_n \leq x_n) = \Phi_\Sigma(\Phi^{-1}(F_1(x_1)), \dots, \Phi^{-1}(F_n(x_n))), \quad (7.4)$$

where Φ_Σ is the cumulative distribution function (CDF) of the Gaussian multivariate distribution $\mathcal{N}(0, \Sigma)$: $\Phi_\Sigma(x_1, \dots, x_n) = \int_{-\infty}^{x_1} \dots \int_{-\infty}^{x_n} \frac{e^{-\frac{1}{2}x^T \Sigma^{-1}x}}{\sqrt{(2\pi)^n \det(\Sigma)}}$ and Φ is the CDF of a one-dimensional standard normal $\mathcal{N}(0, 1)$. Algorithm A 2 in Appendix A.1 describes the procedure to sample from the Gaussian copulas.

Consistent with the rest of our methodology, we use correlation, rather than covariance matrices to parametrize the Gaussian copulas. Correlation matrices are a subset of covariance matrices

that have only values of 1 on the diagonal.

To generate synthetic datasets D matching the correlation constraints and the one-way marginals of the private dataset D_T , we distinguish between two cases:

1. When the variables are all standard normals, i.e., $F_i = \Phi, i = 1, \dots, n$, the correlations of the distribution output by the Gaussian copula are identical to the parameter Σ . We first generate a correlation matrix C using the algorithm corresponding to the attack scenario (e.g., Algorithm 8 for **S2**), then set $\Sigma = C$ to sample a dataset whose empirical correlation approximately match C .
2. When the variables have arbitrary one-way marginals, the correlations of the distribution output by the Gaussian copula parameterized by these marginals are not necessarily equal to the parameter Σ [Xia16]. This means that the adversary must not directly parameterize the Gaussian copula by setting $\Sigma = C$, but instead *must modify the correlation constraints* $V = \{(c_T)_{i,j}, (i, j) \in P\}$ (with C_T denoting the correlation matrix of private dataset D_T) to a new set of constraints V' , generate a correlation matrix C' under the constraints given by V' (instead of V), and only then set $\Sigma = C'$. V' is chosen such that the empirical correlations computed on the synthetic datasets are approximately equal to V and thus match the correlation constraints. Algorithm 11 details a simple, yet very effective heuristic to modify the correlation constraints.

To train our meta-classifier, we first generate K synthetic datasets $D^k, k = 1, \dots, K$. Second, we train a model $\mathcal{M}^k$ on each dataset using the same architecture and hyperparameters as the target model, but a different seed (we refer the reader to Section 7.3 for results of our attack when the adversary knows the seed). Finally, we extract a set of features from each model in the form of confidence scores computed on a synthetic query dataset D_{query} generated in the same way as the other datasets, i.e., to match the adversary's knowledge. We train the meta-classifier to infer the correlation $\rho(X_1(D^k), X_2(D^k))$, discretized over B bins, given as input the features extracted from the model $\mathcal{M}^k$. The complete procedure is described in Section 7.4.1.

7.2.3 Empirical evaluation

Unless otherwise specified, we assume the adversary to have black-box access to a target model $\mathcal{M}_T$ trained on a target dataset D_T , with the goal of inferring the correlation $\rho(X_1, X_2)$, and we consider the default adversary **(S2)**, i.e., the adversary knows $\rho(X_i, Y)_{i=1, \dots, n-1}$.

Across experiments, we run our correlation inference attacks against the target model by, first, training a meta-classifier on features extracted from models trained on synthetic datasets satisfying the adversary’s knowledge about the target dataset D_T . Second, we apply the meta-classifier to features extracted from the target model $\mathcal{M}_T$ to infer the bin to which the target correlation $\rho(X_1, X_2)$ belongs. We repeat the experiment multiple times and compute the attack accuracy as the fraction of correct guesses.

Target models. We apply our attack to Logistic Regression (LR) and Multilayer Perceptron (MLP) models, which are standard choices for learning tasks on tabular training datasets. We train the models on synthetic and real-world datasets, as described in the following sections.

For the LR models, we use the implementation provided by the Scikit-learn 0.24.1 [PVG⁺11] library with default parameters. For the MLP models, we implement in Pytorch 1.10.0 [PGM⁺19] an architecture consisting of two hidden layers of sizes 20 and 10 with ReLU nonlinearity. In the experiments on synthetic data (Sections 7.2.3, 7.2.3 and 7.2.3), we train the models using gradient descent. In the experiments on real-world datasets (Section 7.2.3), we instead train the models using mini-batch gradient descent with a batch size of 128, as this leads to better target model accuracy. The models are trained on 90% of the samples for up to 100 epochs, stopping the models after 5 epochs of non-improving accuracy on the remaining 10%. We use a learning rate of $\eta = 0.05$.

To quantify how much information is leaked by different types of models, we first compare the performances of our model-based and model-less attacks in a controlled setting. More specifically, we train the target models $\mathcal{M}_T$ on synthetic data generated using the Gaussian copula generative model. Recall that this model is parameterized by a correlation matrix,

which allows us to explore the entire range that can be taken by the correlation constraints $\rho(X_i, Y)_{i=1, \dots, n-1}$ and characterize their impact on the performance of the attack.

Impact of constraints for $n = 3$ variables

To characterize the impact of the correlation constraints on the attack accuracy, we study a simple setting in which there are only two input variables and one output variable ($n = 3$). We consider all possible ranges for the correlation constraints $\rho(X_1, Y)$ and $\rho(X_2, Y)$ by dividing the grid $[-1, 1] \times [-1, 1]$ into 200×200 equal cells. We report one accuracy for every cell $[a_1, a_2] \times [b_1, b_2]$, evaluating the performance of an adversary (model-less or model-based) at inferring the target correlation $\rho(X_1, X_2)$ for datasets subject to correlation constraints belonging to that cell, i.e., such that $a_1 \leq \rho(X_1, Y) < a_2$ and $b_1 \leq \rho(X_2, Y) < b_2$.

In every cell, we draw T' pairs of constraints $\rho(X_1, Y)$ and $\rho(X_2, Y)$ uniformly in $[a_1, a_2]$ and $[b_1, b_2]$, respectively. Each pair is used to generate, in order: (1) a target correlation matrix C_T using Algorithm 8, (2) a target dataset D_T of 1000 samples using Gaussian copulas parametrized by C_T and standard normal one-way marginals, and (3) a target model $\mathcal{M}_T$ trained on D_T to perform the binary classification task $Y > 0$. We make three observations regarding (1). First, since there are only $n = 3$ variables, Algorithm 8 and Algorithm 9 are equivalent. Second, $\rho(X_1, X_2)$ is the only correlation coefficient being generated. Third, this value is sampled uniformly over the range defined by Equation 7.3 and correlation constraints $\rho(X_1, Y)$ and $\rho(X_2, Y)$. For every pair of constraints, given access to it and to $\mathcal{M}_T$, the adversary's goal is to infer $(c_T)_{1,2} = \rho(X_1, X_2)$. Executing our attack independently on all the T' targets in every cell would require us to generate $K \times T' \times 200 \times 200$ models in order to train meta-classifiers, which is computationally infeasible. Instead, in every cell, we (1) run the model-based attack by training one meta-classifier and (2) run the model-less attack by empirically estimating, from the samples, the range of values that $\rho(X_1, X_2)$ can take when the correlation constraints belong to that cell. We evaluate the attacks using 5-fold cross-validation and setting $T' = 1500$.

Figure 7.3A shows an adversary with no access to the model can infer the target correlation $\rho(X_1, X_2)$ 56.0% of the time on average over all the cells, much better than random (33.3%).

These results confirm the theoretical analysis presented in Section 7.2.2 with our model-less attack achieving perfect accuracy in the corners of the grid and the top center region (with its symmetric counterparts). They also highlight why it is essential to have a strong baseline different from the random guess baseline in order to quantify the leakage from the model. The accuracy of the model-less attack then slowly decreases as the constraints move to regions of higher uncertainty, in which more than one bin is possible.

Figure 7.3B shows that black-box access to a Logistic Regression (LR) allows an adversary to greatly improve the accuracy of the attack, reaching 95.6% on average over all the cells. As expected, the model-based attack achieves perfect accuracy in the regions where the model-less attack also does. Outside these regions, the model-based attack vastly outperforms the model-less attack. Its accuracy exceeds 90% everywhere, with the exception of the cross-like region centered in the origin where the models leak less information than elsewhere. At the center of the cross ($|\rho(X_i, Y)| \leq 0.02, i = 1, 2$), where the input variables X_1 and X_2 are uncorrelated with the output variable Y , the accuracy of our attack is 47.2%. When there is little to no correlation between the inputs and the output variable Y ($\rho(X_i, Y) \simeq 0$), the target models do not learn much about the dataset. As for the arms of the cross, where $\rho(X_1, Y) \simeq \rho(X_2, Y) \ll 1$, we believe that the lower attack performance is due to the theoretical interval attainable by the target correlation $\rho(X_1, X_2)$ being close to $[-1, 1]$ and hence the adversary having maximum uncertainty. Even in this scenario, the leakage from the model is much higher than the baseline.

Figure 7.3C shows that multilayer perceptron (MLP) models are also vulnerable to our attack, achieving an accuracy of 82.3% on average over the cells. This shows that access to an MLP model greatly improves over the model-less attack, but slightly less than the LR. This may seem surprising at first, since our MLP, with its 292 learnable weights, has a much higher capacity than the LR with its 3 weights and thus a higher ability to retain information about the dataset. We believe that this gap is due to the fact that the non-convex loss used to train MLP models has a (potentially large) number of local minima [CHM⁺15]. This means that the models trained by an adversary's in order to generate the meta-classifier's training dataset are unlikely to reach the same local minima as the target model, increasing the adversary's uncertainty. We refer the reader to Section 7.3 for results of our attack when we remove the

uncertainty caused by the difference in randomness. We also find that the cross-like behavior is exacerbated on the MLP: the center of the cross is larger, reaching an average accuracy of 48.1% for $|\rho(X_1, Y)| \leq 0.1$.

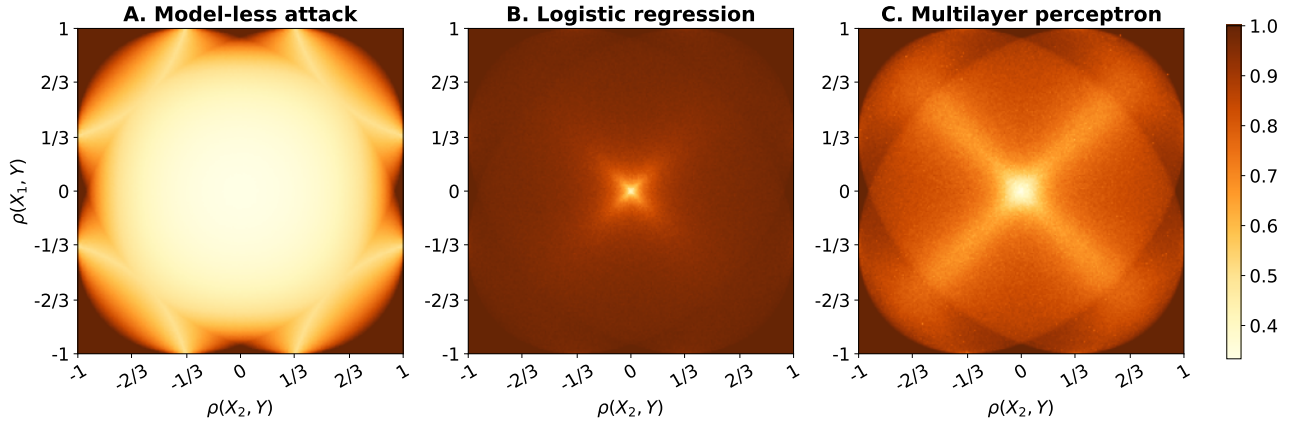


Figure 7.3: **Accuracy of the correlation inference attack on $n = 3$ variables as a function of the correlation constraints $\rho(X_1, Y)$ and $\rho(X_2, Y)$.** We show results of (A) the model-less attack and of the model-based against two different target models: (B) logistic regression and (C) multilayer perceptron. The color of each cell indicates the accuracy of our correlation inference attack applied to models trained on synthetic datasets whose correlations $\rho(X_1, Y)$ and $\rho(X_2, Y)$ belong to the region defined by the cell. There are 200×200 cells in total.

White-box results. Figures 7.4 and 7.5 show that white-box access to the model, where the features extracted from the model include its weights, does not improve the performance of our attack.

We evaluate the performance of white-box correlation inference attacks using the model weights as features. The weights of the Logistic Regression are its coefficients (including the bias). For the MLP, we explore two options: (1) using the *raw weights* (coefficients of the linear layers, including the bias), flattened and concatenated into a single vector and (2) the *canonical weights*, extracted in the same way from the model after sorting the neurons in each layer according to the sum of weights [GWY⁺18].

Figure 7.4 shows that the weights of Logistic Regression models yield similar performance to the confidence scores (95.1% vs 95.6%). This is likely due to the small number of weights (three in total), whose information we believe to be redundant with the one contained in the confidence scores extracted from $Q = 100$ records.

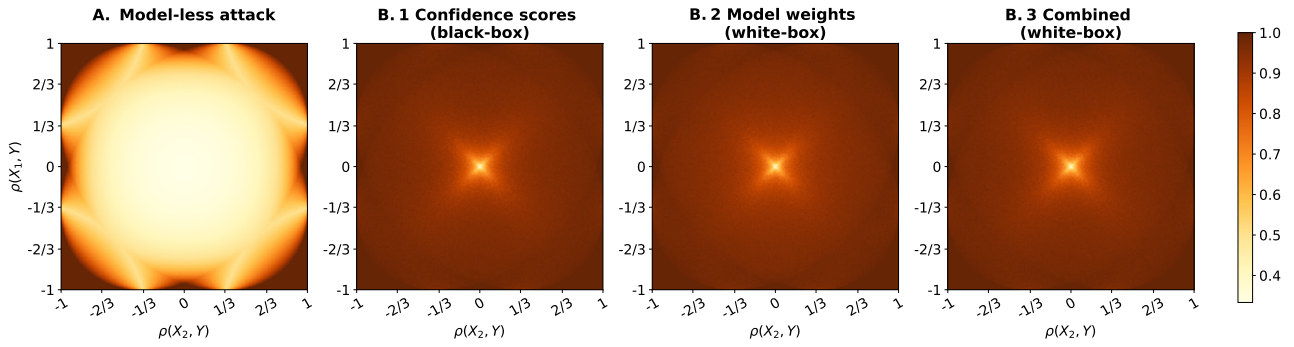


Figure 7.4: **Logistic regression: Comparison between black-box and white-box correlation inference attacks.** We report results for our model-less attack (A), our black-box model-based attack using as features the confidence scores (B. 1), and our white-box model-based attacks using as features the model weights (B. 2) or the combined weights and confidence scores (B. 3).

Figure 7.5 shows that the raw weights of MLP models do not leak more information than the model-less attack (55.0%). This negative result is in line with previous findings from the property inference literature [GWY⁺18, ZTO21] and can be attributed to the permutation equivalence property of MLPs [GWY⁺18], whereby neurons of internal layers can be permuted without changing the underlying function. As a result, a meta-classifier operating on inputs in their default ordering (with $k!$ equivalent orderings being possible in each layer, where k is the number of neurons in the layer) needs to learn a more difficult task. We show that the canonical weights yield much better performance (65.5%), albeit lower than the confidence scores.

On both models, combining the model weights and the confidence scores (Figure 7.4-B.3 and Figure 7.5-C.4) does not yield better results than using the confidence scores alone.

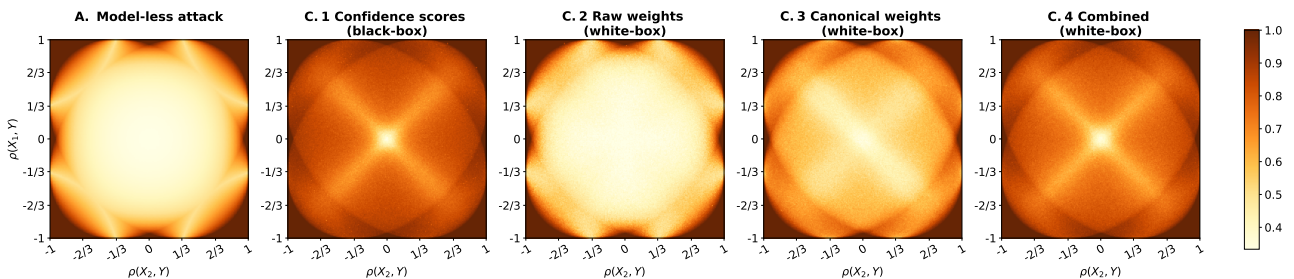


Figure 7.5: **Multilayer perceptron (MLP): Comparison between black-box and white-box correlation inference attacks.** We report results for our model-less attack (A), our black-box model-based attack using as features the confidence scores (C. 1), and our white-box model-less attacks using as features the raw model weights (C. 2), the canonical weights (i.e., the raw weights mapped to a canonical representation [GWY⁺18], C. 3), or the combined canonical weights and confidence scores (C. 4).

Increasing the number of variables n

Next, we compare the performances of our attack against Logistic Regression and MLP models when increasing the number of variables n , under three different attack scenarios.

For each $n = 3, \dots, 10$, we sample 1000 target correlation matrices $C_T \in \mathcal{C}$ using Algorithm A 1 in Appendix A.1. To obtain a balanced distribution over the target correlations, we generate the value $\rho(X_1, X_2)$ first, as explained in Section 7.4.1. Then for each C_T , we sample a synthetic target dataset D_T (parametrized by C_T and standard normal marginals) and train a target model $\mathcal{M}_T$ on it.

Figure 7.8 **S2** shows that the adversary is consistently and significantly better than the model-less attack. The performance of our attack decreases slowly with the number of variables in the dataset, from 96.1% ($n = 3$) to 70.4% ($n = 6$) and 58.2% ($n = 10$) when applied to LR models, while the model-less attack has a roughly constant accuracy of 49.3% for all values of n . This decrease when n increases is expected. Indeed, first, the number of unknown correlations increases quadratically with n , which increases the uncertainty of the adversary. Second, the more variables a model is trained on, the less likely our target variables X_1 and X_2 are to be useful for the prediction task at hand. This in turn means the model might learn less information about $\rho(X_1, X_2)$.

Figure 7.6 shows that when at least one of the constraints $\rho(X_1, Y)$ or $\rho(X_2, Y)$ is large in absolute value (meaning that X_1 , X_2 , or both are good predictors of the output variable), our attack performs better (see Appendix D.1 for completeness of the results). For instance, on $n = 6$ variables our attack reaches 90.6% when $\max(|\rho(X_1, Y)|, |\rho(X_2, Y)|) \geq 0.8$, much better than 62.7% when $\max(|\rho(X_1, Y)|, |\rho(X_2, Y)|) \leq 0.2$. Interestingly, even when both constraints are small, e.g., when $\max(|\rho(X_1, Y)|, |\rho(X_2, Y)|) \leq 0.2$, the model-based attack can still infer $\rho(X_1, X_2)$ much better than random, especially for a smaller number of variables. We discuss potential reasons for this behavior in Section 7.3.

Interestingly, the difference in vulnerability between Logistic Regression and MLP models reduces as the number of variables in the dataset n increases, with our attack achieving similar

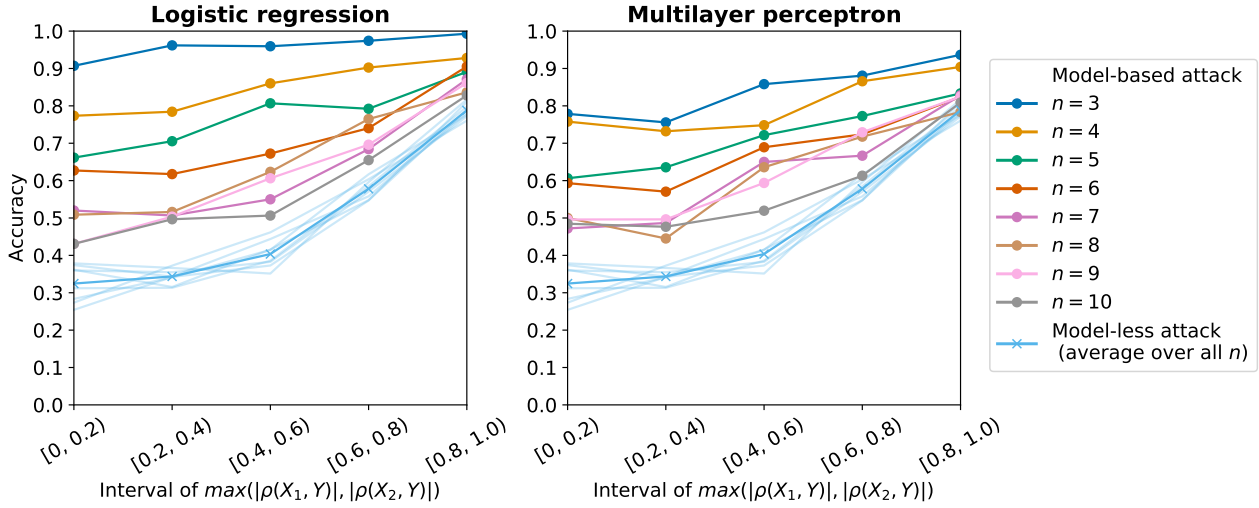


Figure 7.6: **Impact of the largest constraint $\max(|\rho(X_1, Y)|, |\rho(X_2, Y)|)$ on the attack accuracy for different number of variables n .** We compare the results of our model-based and model-less attacks on logistic regression (left) and MLP models (right).

accuracy against the two for $n \geq 6$. We rule out the possibility of models learning the same decision boundary, as the MLP models are, on average, 1 to 4% less accurate, with the gap in accuracy increasing with n . We believe this could instead be due to MLP models' training algorithm becoming more stable as n increases, i.e., the loss having fewer local minima, as more input variables are likely to be predictive of the output Y .

We obtain similar trends on $N_B = 5$ classification bins, and refer the reader to Figure 7.7 for detailed results.

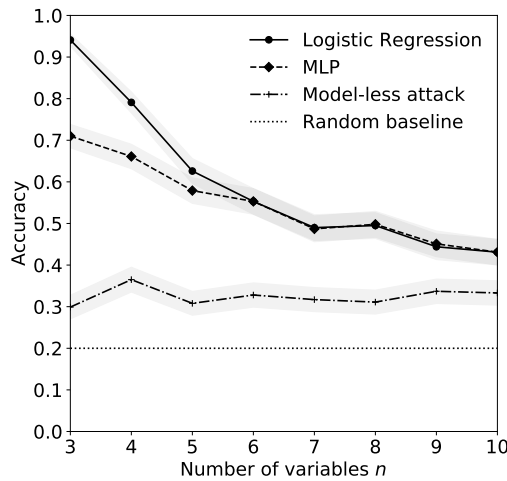


Figure 7.7: **Attack accuracy under scenario S2 and $N_B = 5$ correlation bins for different number of variables in the dataset n .** We report the accuracy (with 95% confidence interval) over 1000 target models.

Figure 7.8 **S1** shows that reducing the knowledge of correlations to only $\rho(X_1, Y)$ and $\rho(X_2, Y)$, only slightly decreases the attack accuracy. More specifically, the accuracy is 78.5% when $n = 3$ vs 85.1% for **S2**, 63.4% when $n = 6$ vs. 70.4% for **S2**, and 55.5% when $n = 10$ vs. 58.2% for **S2**. This shows that correlation inference attacks are a concern even when facing a weak adversary.

We complete our understanding of the impact of uncertainty on the attack by additionally considering a very strong adversary, who has knowledge of all the correlations but the target $\rho(X_1, X_2)$. Figure 7.8 **S3** shows that the accuracy of our attack now increases as n increases. We believe this is because the fraction of unknown correlations now reduces quadratically with n . More specifically, our attack improves as n increases, from 95.5% ($n = 3$) to 99.5% ($n = 6$) and 99.9% ($n = 10$) on Logistic Regression models. We obtain similar results on MLP models, as our attack improves from 83.1% ($n = 3$) to 95.6% ($n = 6$) and 99.9% ($n = 10$). Interestingly, the strongest adversary can again exploit a higher vulnerability in LR models compared to MLP models. The model-less attack also increases with n , but achieves a lower accuracy than our model-based attack for $n \leq 8$.

Impact of mitigations

We analyze the effectiveness of three possible mitigations in preventing our correlation inference attack.

Limiting the number of queries. It could be put in place, e.g. in the machine learning as a service case, by the model developer in order to prevent the user from extracting too much information. We vary the number of queries Q made to an LR from 1 to 5000 and report the attack accuracy against the same targets as before. Figure 7.9A shows that this measure does not prevent our attack. Indeed only 5 queries are necessary to reach close to optimal accuracy for $n = 3$, 50 queries for $n = 4$ and 100 for $n = 5$. Even a single query is sufficient for the attack to reach 79.1% on $n = 3$ variables, 62.2% on $n = 4$, and 59.0% on $n = 5$, significantly higher than the model-less attack which only reaches 49.6%.

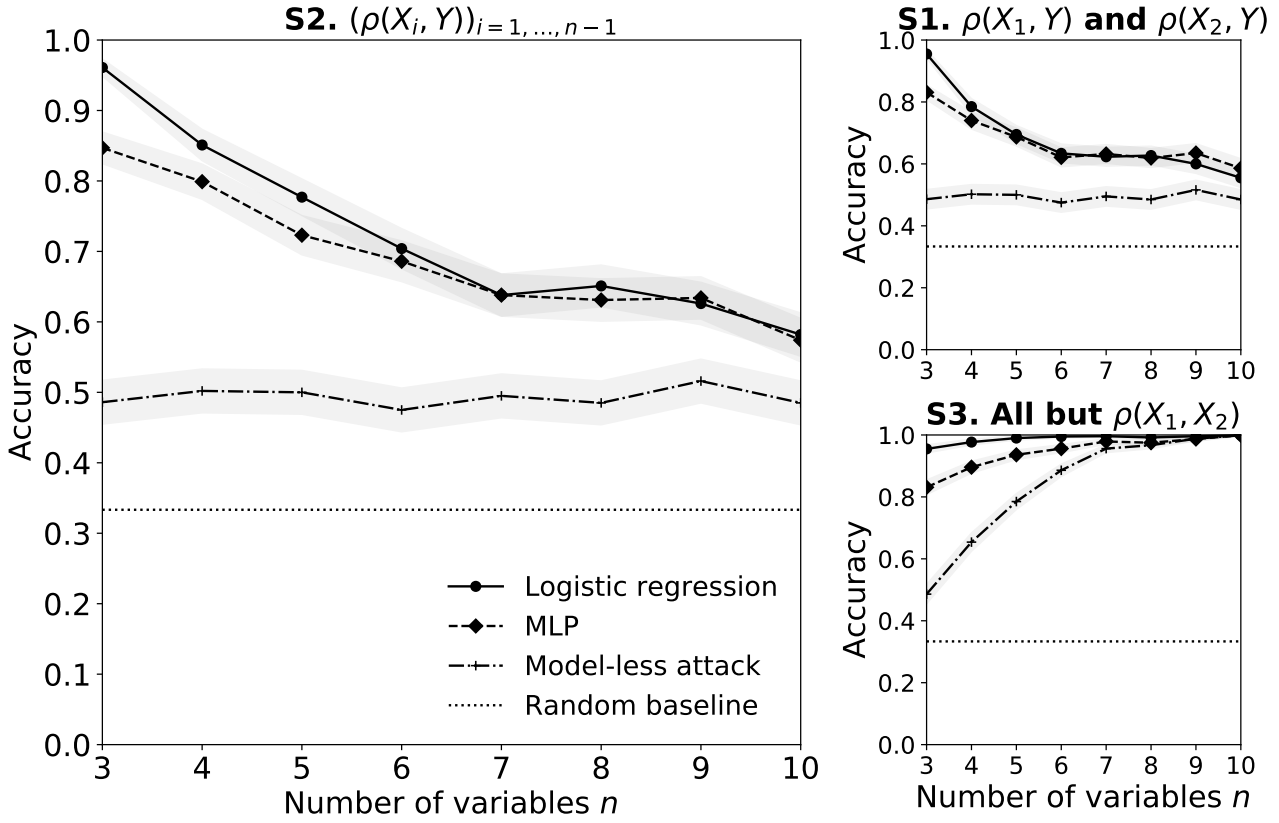


Figure 7.8: **Attack accuracy for different scenarios and number of variables in the dataset n .** We consider three different attack scenarios: **(S1)** the adversary knows the correlations between the target variables and the output variable $\rho(X_1, Y)$ and $\rho(X_2, Y)$, **(S2)** the adversary knows the correlations between all the input variables and the output variable $\rho(X_i, Y)_{i=1, \dots, n-1}$, and **(S3)** the adversary knows all the correlations between the variables except for the target correlation $\rho(X_1, X_2)$. We compute the accuracy (with 95% confidence interval) over 1000 target models.

Lowering the precision of confidence scores. It is another popular mitigation deployed when releasing models, mainly by allowing the model to only output the class label [CCTCP21]. Figure 7.9B shows that this measure has a very small impact on our attack, only decreasing its accuracy from 96.0% when all digits are made available to 90.8% when only the label is, on $n = 3$ variables (similarly, from 84.9% to 77.2% when $n = 4$ and from 78.4% to 71.9% when $n = 5$). We obtain similar results on MLP models, as shown on Figure D 3 in Appendix D.1.

Differential privacy (DP). It is a popular privacy definition requiring that the model should not depend too strongly on any one record, as controlled by ϵ [DMNS06] (see Chapter 2.3 for further details). DP protects individual privacy and ML models trained with DP guarantees have been shown to prevent individual-level attacks, including membership inference attacks.

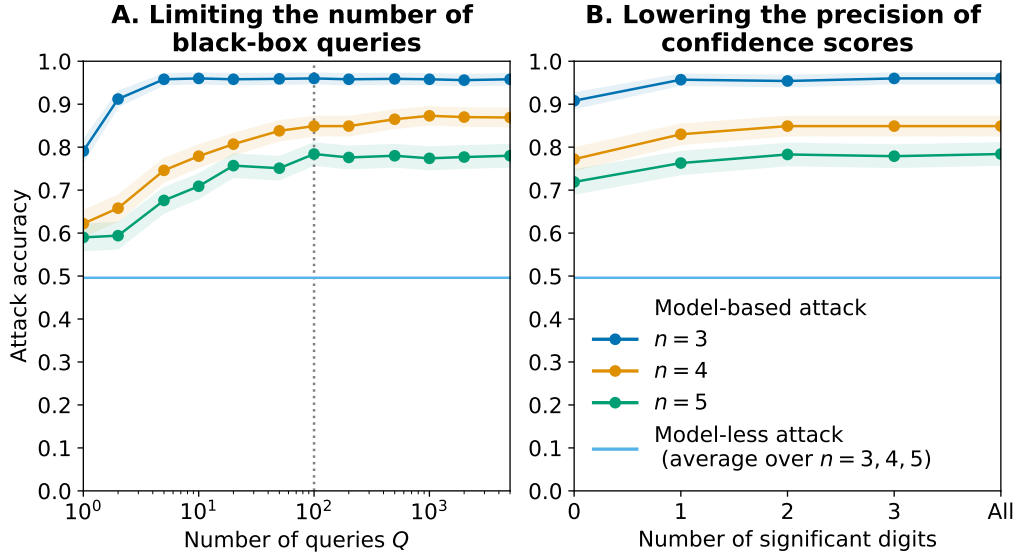


Figure 7.9: **Impact of mitigations on the accuracy of our attack against Logistic regression models.** We report results for two mitigations: limiting the number of black-box queries (A) and lowering the precision of confidence scores (B). We report the accuracy (with 95% confidence interval) of our model-less and model-based attacks over 1000 targets models for $n \in \{3, 4, 5\}$ variables.

Although DP is not meant to protect population-level information such as correlations, the noise added might perturb the model enough to mitigate our attack. We train 1000 DP Logistic Regression models with objective perturbation [CMS11] using the Diffprivlib library [HBMAL19] and different values of ϵ .

Figure 7.10 shows that the vulnerability of a model to our attack is strongly linked to its utility, i.e., accuracy. Indeed, for large value of ϵ , models are both useful and vulnerable. As we decrease ϵ , more noise is added which reduces both the vulnerability of models to our attack but also their utility. For small value of ϵ , our attack does not perform better than the model-less one, but at the cost of models losing any utility.

Taken together, our results suggests these mitigations to be ineffective in preventing our attack.

Real-world dataset evaluation

We evaluate the performance of our attack on the following three real-world tabular datasets: Fifa19 [Gad19], Communities and Crime [Red11], and Musk [Die98], presented in Appendix B.3.

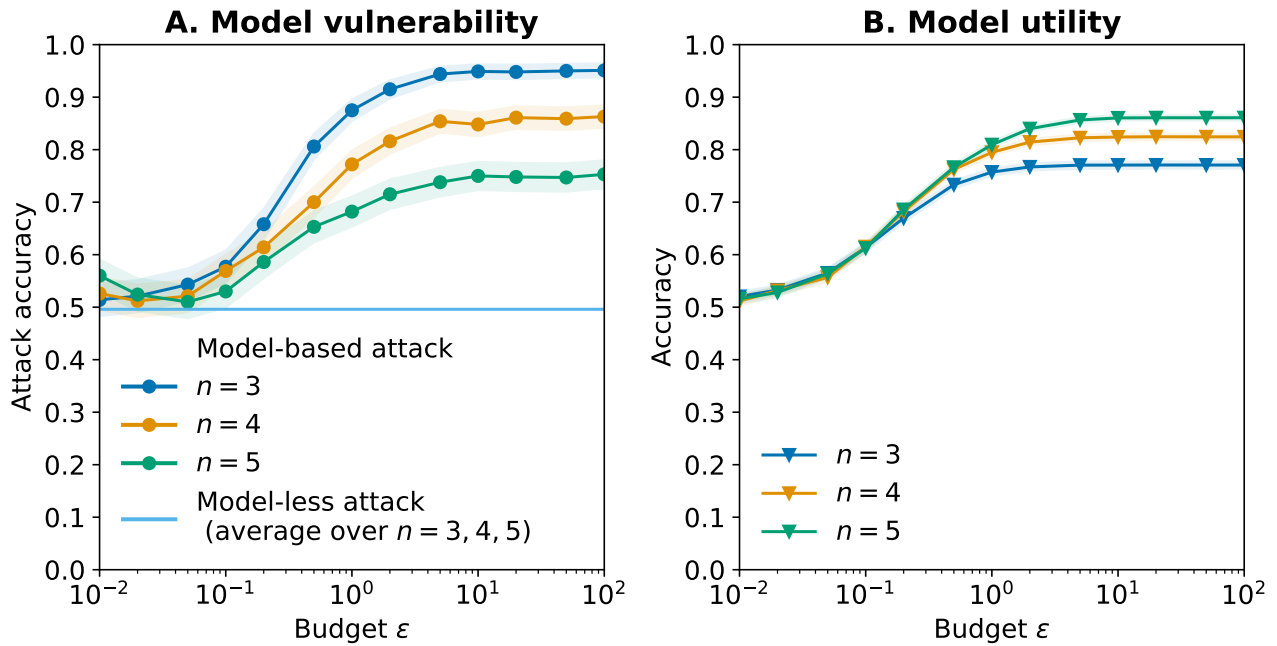


Figure 7.10: **Trade-off between the vulnerability of DP Logistic regression models to our attack (A) and their utility (B).** We compute the metrics (with 95% confidence interval) over 1000 target models for $n \in \{3, 4, 5\}$ variables.

For each dataset, we select 100 data collections consisting of $n = 4$ columns, by sampling three different variables X_1, X_2, X_3 uniformly at random without replacement among the set of all triplets then adding the output variable Y . We report the mean and standard deviation of the attack accuracy averaged over 10 runs of 100 data collections each.

Contrary to the synthetic results above, the correlations in our real-world datasets are not uniform. Instead they are, to a different extent depending on the dataset, skewed towards zero. This means that a large fraction of the target correlations belong to the “low” bin $[-1/3, 1/3]$. Beyond this it also means that the correlation constraints $\rho(X_1, Y) = \cos \theta_1$ and $\rho(X_2, Y) = \cos \theta_2$ are very small, thus leading to intervals $R = [\cos(\theta_1 + \theta_2), \cos(\theta_1 - \theta_2)]$ that are close to, but not equal to $[-1, 1]$ (e.g., $R = [-0.99, 0.99]$). These intervals thus encompass the “low” bin fully, and the other bins almost fully. For these two reasons, the model-less attack achieves on real-world datasets an “artificially” high accuracy. One solution would be to balance the test distribution. However, we instead prefer to show results for $N_B = 5$ bins. This indeed reduces the artificial advantage of the model-less attack while maintaining the generality of the evaluation.

Table 7.1 shows that, despite its advantage, the model-less attack is outperformed by our model-based attack in five out of six cases. In Fifa19, it achieves an accuracy of 91.2% against LR models and 77.9% against MLP models, strongly outperforming the model-less attack which reaches 60.2%. Our attack similarly outperforms the model-less attack in the Communities and Crime dataset, achieving an accuracy of 86.0% against the LR models and 75.6% against the MLP models compared to the 73.6% reached by the model-less attack. Finally, while our attack achieves an accuracy of 82.0% on LR models, it is outperformed by the model-less attack when it comes to MLP models in the Musk dataset (56.3% vs 67.8%).

Table 7.1: **Results of our correlation inference attacks on three real-world datasets.**

Number of bins	Dataset	Random guess	Model-less attack	Model-based attack	
				Logistic Regression	MLP
$N_B = 3$	Fifa19	33.3	60.2 (5.0)	91.2 (3.6)	78.8 (4.5)
	Communities and Crime	33.3	73.6 (2.7)	86.0 (3.6)	75.6 (4.0)
	Musk	33.3	67.8 (5.6)	82.0 (3.2)	56.3 (6.2)
$N_B = 5$	Fifa19	20.0	29.4 (5.1)	79.1 (3.6)	61.2 (6.1)
	Communities and Crime	20.0	27.6 (3.2)	70.6 (3.8)	56.0 (5.3)
	Musk	20.0	28.7 (4.5)	72.0 (5.5)	41.7 (6.4)

However, as soon as we increase the number of bins, removing the artificial advantage of the model-less attack, our attack vastly and consistently outperforms the model-less attack. On Fifa19, it achieves an accuracy of 79.1% against LR models and 61.2% against MLP models, strongly outperforming the model-less attack which reaches 29.4%. It similarly outperforms the model-less attack on the Communities and Crime and Musk datasets. On Communities and Crime it achieves an accuracy of 70.6% against the LR models and 56.0% against the MLP models compared to the 27.6% reached by the model-less attack. On Musk, it achieves 72.0% against the LR models and 41.7% against the MLP models compared to the 28.7% reached by the model-less attack.

7.2.4 Attribute inference attack

We have so far shown that dataset correlations, which are fundamental summary statistics about a dataset, can be extracted from a machine learning model trained on it. We here

argue that the ability of an adversary to accurately extract statistical information can enable other attacks, in particular allowing weaker adversaries to infer membership or attributes from trained machine learning models. More specifically, we present an attribute inference attack that uses as building blocks the correlations between input variables extracted from the model using our attack.

We assume the first variable in the dataset X_1 to be sensitive and consider an adversary that aims to recover the sensitive attribute value x_1 of a *target record* $(x_1, x_2, \dots, x_{n-1}, y) \in D_T$ using black-box access to a model $\mathcal{M}_T$ trained on D_T . In line with the literature [FLJ⁺14, MDDV⁺22, JE22], we assume that the adversary knows the partial record $(x_2, \dots, x_{n-1})$, the label y , and the one-way marginals of the attributes.

Correlation Inference based Attribute Inference Attack. The intuition behind our attack is that the correlation between the sensitive variable X_1 and the other variables, when combined with knowledge of the partial record $(x_2, \dots, x_{n-1}, y)$, can lead to accurate inference of the sensitive attribute x_1 . We propose an attack that extracts the correlations between the input variables $\rho(X_i, X_j), 1 \leq i < j \leq n-1$ from the model using our model-based attack, under default scenario **S2**, and uses them to generate synthetic data satisfying these correlations. We then search for (approximate) matches of the partial record $(x_2, \dots, x_{n-1}, y)$ in the synthetic data and return as prediction $\hat{x}_1$ the average value of the sensitive attribute values of matched records. We refer the reader to Section 7.4.2 for the details.

We implement CI-AIA using the same parameters as in Section C.5, along with the following parameters: $S' = 1000$ synthetic datasets, $G = 100$ bins, and $m_i = 2$ and $\delta_i = 0.5$ for every $i = 1, \dots, n-1$.

Other Attribute Inference Attack considered. We now describe the five attacks from the previous works against which we compare our CI-AIA method. To perform attribute inference over a continuous sensitive attribute, we divide the attribute range into G bins of equal length, infer the most likely bin using a given attack, then return a uniform sample from the bin as the final prediction. Given a candidate record $(x_1^g, x_2, \dots, x_{n-1}, y)$ where the value of the unknown

sensitive attribute x_1 is substituted with a sample x_1^g of the g -th candidate bin, we will denote by V_y^g the confidence that the candidate record is classified with the target label y , i.e., the l -th element of output vector $\mathcal{M}_T(x_1^g, x_2, \dots, x_{n-1})$.

1. **Fredrikson et al. [FLJ⁺14]** The attack infers the most likely bin $g = 1, \dots, G$ as per the probability $\Pr(x_1^g) \Pr(\hat{y} = y | \arg \max_{l=1}^L \mathcal{M}_T(x_1^g, x_2, \dots, x_{n-1}) = \hat{y})$, where x_1^g is a value sampled uniformly at random in the g -th sub-interval. The second term is approximated using the confusion matrix of the target model, assumed by Fredrikson et al. [FLJ⁺14] to be known by the adversary.
2. **CSMIA (Mehnaz et al. [MDDV⁺22])** The attack sets the sensitive attribute value of the partial record to each possible value (here, one of G bins) and retrieves the model prediction $\hat{y}^g = \mathcal{M}_T(x_1^g, x_2, \dots, x_{n-1})$, $g = 1, \dots, G$. Note that we represent a bin using a uniform sample in the bin x_1^g . Then, the attack distinguishes three cases: (1) if $\hat{y}^g = y$ for only one value of g , the approach returns x_1^g , (2) if $\hat{y}^g = y$ for more than one value of g , the approach returns x_1^g on which the model is the most confident, and (3) if $\hat{y}^g \neq y$ for all values of g , the approach returns x_1^g on which the model is the least confident.
3. **Yeom et al. [YGFJ18]** The attack returns the sensitive attribute value with the largest prior $\Pr(x_1^g)$ that passes a membership inference test. More specifically, the attack uses a membership oracle $\mathbb{O}$ that returns 1 if $(x_1^g, x_2, \dots, x_{n-1}, y) \in D_T$ and 0 otherwise. The attack returns $\arg \max_g \Pr(x_1^g) \cdot \mathbb{O}(\mathcal{M}_T, (x_1^g, x_2, \dots, x_{n-1}))$ (where x_1^g is sampled uniformly at random in the g -th bin). We use the threshold-based membership oracle of Yeom et al. [YGFJ18], that returns 1 if and only if the model's confidence on the target record is larger than a threshold τ . We use $\tau = 0.5$.
4. **Jayaraman and Evans [JE22]** removes the need for an oracle in the attack by Yeom et al. [YGFJ18], proposing two variants. The first one is CAI, which returns $\arg \max_g V_y^g$. The second one is WCAI, which returns $\arg \max_g \Pr(x_1^g | x_2, \dots, x_{n-1}) V_y^g$, making use of the conditional probability of the sensitive attribute given the non-sensitive attributes. In their work, this quantity is estimated empirically by fitting a machine learning model

to infer X_1 based on $X_2, \dots, X_{n-1}$. The model is fitted on auxiliary data drawn from a similar distribution as the private dataset. As we here study a weaker adversary lacking access to auxiliary data, under our assumptions the attack would simply become $\arg \max_g \Pr(x_1^g) V_y^g$.

Results. We evaluate our correlation-inference-based attribute inference attack (CI-AIA) on the Fifa19 dataset [Gad19], randomly selecting 1000 data collections consisting of $n = 4$ columns. For each data collection, we train a Logistic Regression model and run the attack against 500 randomly selected records (with the parameters described in Section 7.2.4). To evaluate the attack performance, we divide the range of the sensitive attribute X_1 into three bins corresponding to the tertiles of the distribution. Thus, a majority baseline attains an accuracy of roughly 33% (denoted as acc_{ref}). The improvement of the different attacks studied over the majority baseline, referred to as normalized gain ($G(acc, acc_{ref})$), is computed using the following formula: $G(acc, acc_{ref}) = 100 \times \frac{acc - acc_{ref}}{100 - acc_{ref}}$.

We compare our attack with the five attacks from previous works: the first AIA of Fredrikson et al. [FLJ⁺14], CSMIA by Mehnaz et al. [MDDV⁺22], the threshold-based attack of Yeom et al. [YGFJ18], and the CAI and WCAI methods of Jayaraman and Evans [JE22]. We further compare our approach with two other baselines: a naïve approach that returns a random sample from the one-way marginal of the sensitive attribute X_1 , which is known to the adversary, and another approach, referred to as $X_1 - Y$ **correlation**, that runs our CI-AIA approach on the variables X_1 and Y alone. Its goal is to quantify the leakage from the adversary knowledge of $\rho(X_1, Y)$ without access to the model. Indeed, knowledge of the correlation $\rho(X_1, Y)$ can lead to better than random inference of the attribute x_1 since y is known. This baseline acts as a lower bound for CI-AIA, as the latter additionally assumes access to the model.

Table 7.2 shows that our novel correlation inference-based attribute inference attack, referred to as CI-AIA, outperforms previous works and baselines, demonstrating how correlations can be used as building blocks for other attacks. Our attack achieves an accuracy of 49.7%, significantly better than CSMIA (Mehnaz et al. [MDDV⁺22]), which only reaches 46.5%, on par with CAI (Jayaraman and Evans [JE22]), and the method of Fredrikson et al. [FLJ⁺14], which reaches

Table 7.2: **Comparison between CI-AIA and other attribute inference attacks on Fifa19.** We report the attack accuracy averaged over 1000 runs (with 95% confidence interval).

Method	Accuracy	Normalized accuracy gain over majority voting
(Ours) CI-AIA	49.7 $\pm$ 1.0	24.9 $\pm$ 1.5
Fredrikson et al. [FLJ ⁺ 14]	38.4 $\pm$ 0.8	08.1 $\pm$ 1.2
CSMIA (Mehnaz et al. [MDDV ⁺ 22])	46.5 $\pm$ 0.9	20.1 $\pm$ 1.3
Yeom et al. [YGFJ18]	38.5 $\pm$ 0.8	08.2 $\pm$ 1.2
CAI (Jayaraman and Evans [JE22])	46.5 $\pm$ 0.9	20.1 $\pm$ 1.3
WCAI (Jayaraman and Evans [JE22])	41.7 $\pm$ 0.8	13.0 $\pm$ 1.2
$X_1 - Y$ correlation	41.7 $\pm$ 0.8	13.0 $\pm$ 1.2
Marginal prior	35.3 $\pm$ 0.6	03.4 $\pm$ 0.9

38.4% and is on par with Yeom et al. [YGFJ18].

Our attack also outperforms the $X_1 - Y$ correlation baseline, confirming our intuition that the additional correlations $\rho(X_1, X_i), i > 1$ extracted from the model using our attack lead to more accurate attribute inference.

Differences with CAI and WCAI. Our correlation inference-based attribute inference attack (CI-AIA) is fundamentally different from Jayaraman and Evans’s AIA [JE22]. This is because we do not assume the adversary to know the data distribution via access to auxiliary data. Instead, we extract information about the distribution from the model (the correlation between the input variables) and use it to generate synthetic data. In spite of this fundamental difference, the WCAI formula for inferring the sensitive attribute given the partial record is similar to step 3 of our CI-AIA attack. Recall that in our step 3, we retrieve the average sensitive attribute value x_1^g among all synthetic records which approximately match the partial record $(x_2, x_3, \dots, x_{n-1})$ and the label y . Our step 3 can thus be seen as equivalent to estimating $E(x_1|x_2, \dots, x_{n-1}, y)$, while WCAI returns $\arg \max_g p(x_1^g|x_2, \dots, x_{n-1})V_y^g$.

We study if our CI-AIA can be further improved by applying the WCAI formula instead of our step 3 (described in Section 7.4.2) to the synthetic data generated conditionally on the correlations inferred using our attack. For completeness, we explore different variants to the WCAI formula.

1. Variant 1: We trained a regression model for inferring x_1 given $x_2, \dots, x_{n-1}$, like [JE22]. The model is trained on the same synthetic data – generated conditionally on the correlations inferred from the model – as our attack, instead of auxiliary data [JE22] which our adversary does not have.
2. Variant 2: This is the same as variant 1, except that to understand the impact of conditioning on y – which our step 3 does while [JE22] does not – we trained a regression model for inferring x_1 given $x_2, x_3, \dots, x_{n-1}$, **and** y .
3. Variant 3: This is the same as variant 1, except that we multiply $p(x_1^g|x_2, \dots, x_{n-1})$ by the model’s confidence on label y , V_y^g , to understand if V_y^g helps. This is the closest to the formula of [JE22], the only difference being how $p(x_1^g|x_2, \dots, x_{n-1})$ is estimated. Jayaraman and Evans [JE22] uses a neural network classifier which outputs probabilities, while we use a regression since our attribute x_1 is continuous. As the regression model does not output a probability, we model it as the probability of error using a standard normal distribution $p(x_1^g|x_2, \dots, x_{n-1}) \sim e^{-\frac{(x_1^* - x_1^g)^2}{2}}$, where x_1^* denotes the model’s prediction on the partial record $(x_2, \dots, x_{n-1})$.
4. Variant 4: This is the same as variant 3, except that to understand the impact of conditioning on y – which we do while [JE22] does not – we trained a regression model for inferring x_1 given $x_2, x_3, \dots, x_{n-1}$ **and** y .

Table 7.3 shows that none of the variants of our CI-AIA attack achieve better performance than our original attack. In particular, our CI-AIA achieves the same accuracy (49.7%) as the WCAI formula [JE22] applied to the synthetic data generated using steps 1 and 2 of our attack (49.8% - Variant 3). This means that once correlations have been extracted and synthetic data has been generated conditionally on these correlations using steps 1 and 2 of our attack, either our step 3 or Jayaraman and Evans’s [JE22] formula can be used to achieve similar results. Another interesting finding is that using y in the prediction leads to better performance than not using y at all. The AIA accuracy is 49.5% when x_1 is inferred conditionally on y in addition to the other variables (Variant 2), 49.8% when the prediction uses the model’s confidence on label y

(Variant 3) and 46.7% when y is not used at all (Variant 1). Using y by both conditioning on it and via the model’s confidence (Variant 4) does not significantly improve AIA performance.

Table 7.3: **Comparison between our CI-AIA and attacks obtained by substituting Jayaraman and Evans’s [JE22] prediction rule to step 3 of CI-AIA.** We report the attack accuracy averaged over 1000 runs (with 95% confidence interval) on the Fifa19 dataset.

Method	Accuracy	Normalized accuracy gain over majority voting
(Ours) CI-AIA	49.7 ± 1.0	24.9 ± 1.5
Variant 1	46.7 ± 1.0	20.4 ± 1.5
Variant 2	49.5 ± 1.0	24.6 ± 1.5
Variant 3 (closest formula to [JE22])	49.8 ± 1.0	25.1 ± 1.5
Variant 4	50.5 ± 0.9	26.1 ± 1.3

7.3 Discussion

In this work, we study a new type of leakage in machine learning (ML) models, the leakage of correlations between input variables, proposing the first correlation inference attack against ML models. Our results show that a model remembers more information than previously thought, raising privacy and confidentiality concerns whenever dataset correlations are sensitive information. We develop two different attacks, *model-less* and *model-based*, that together allow to correctly quantify the leakage from the model. We perform an extensive evaluation of our attacks, showing that models leak correlations of their training dataset with high accuracy, in both synthetic and real-world datasets. We also show that mitigations such as limiting the number of queries that can be made to the model, their precision, or training the models with differential privacy guarantees do not help. Finally, we show how correlations extracted using our attack can be used as building blocks for attribute inference attacks, in particular enabling weak adversaries to develop more powerful attacks than previously thought.

Attack motivation. To further emphasize the privacy interest of correlation information, we here describe several use cases where correlation leakage might represent a privacy violation or could lead to individual harm. First, we consider the example presented in the introduction:

a scoring model for depression which leaks that patients living in the inner city are more likely to have used illegal substances, an information that could be used to target them. Second, we consider a model controlling the signalization on the highway. We assume that it was trained on data collected from vehicles in a specific area and that there is a positive correlation between the darkness of the car and the probability of accident in the area. If the model leaks this information to an insurance company operating in the area, this could lead to higher insurance premium for darker cars in the area. Third, we consider a model that leaks the correlation between age and income of people in its training dataset. Furthermore, assume that the model developer claims the model to be trained on representative data. Our correlation inference attack can be used as part of an auditing process, to surface a situation where although the training dataset is balanced along each individual attribute (e.g., age and income taken in isolation), it is not balanced on their intersection, e.g., *in the training dataset* being older is excessively correlated with income.

When does the leakage happen? Our results (Figure 7.3 and Figure 7.6) suggest that (1) models leak the correlation between input variables *more* when these variables are highly correlated (positively or negatively) with the output variable Y than when these variables are not correlated with Y , but that (2) leakage still occurs even when there is little to no correlation between these variables and the target variable. The second finding might seem particularly surprising in the case of $n = 3$ variables, where the resulting model makes random predictions since all its input variables (X_1 and X_2) are statistically independent to the target variable Y (uncorrelation implies statistical independence for Gaussian variables). The apparent contradiction can however be explained by the fact that a model making random predictions is not the same as the model not having any statistical dependency on its training dataset. Indeed, while the optimal model is only intended to learn $P(Y|X = x)$, current learning techniques such as stochastic gradient descent optimize the expected loss computed over records of the joint distribution $P(X, Y)$. Thus, the distribution of the input variables $P(X)$ influences the model. Whether the leakage we observe is a necessary condition for learning to happen and thus, whether robust defenses might exist, remains an open problem.

We now analyze some of the assumptions made on the adversary knowledge.

Access to auxiliary data. Our approach does not require access to the private dataset D_T , nor to a dataset drawn from a similar distribution. Such detailed knowledge of the data is a common assumption in the literature on inference attacks and has led to strong results as the adversary uncertainty is reduced, e.g., [GWY⁺18, ZTO21, CCN⁺22]. We however believe that real-world adversaries are unlikely to have detailed knowledge of the data. Our goal is thus to explicitly extract information about the data distribution and use it to infer attributes of individual records. While in this work we have assumed a Gaussian copula prior on the underlying distribution (allowing us to control for correlations), future work could explore different priors to improve our results.

Knowledge of the correlation constraints. Our default scenario **S2** assumes the adversary to know $(\rho(X_i, Y))_{i=1, \dots, n-1}$. We show that these can be extracted from the model using a small number of queries $\tilde{Q}$. More specifically, we sample $\tilde{Q}$ records $(x_1, \dots, x_{n-1})$ independently from the marginals $F_1, \dots, F_{n-1}$, then query the model to retrieve the predicted label y . We then estimate $(\hat{\rho}(X_i, Y))_{i=1, \dots, n-1}$ on these records. We compute the $\mathbb{L}_1$ error of the predictions, defined as $\frac{1}{n} \sum_{i=1}^n |\hat{\rho}(X_i, Y) - \rho(X_i, Y)|$, over 500 data collections of $n = 4$ columns, and compare it with the $\mathbb{L}_1$ error of a random guess in $[-1, 1]$. Figure 7.11 shows our approach to perform much better than the random guess, reducing the $\mathbb{L}_1$ error from 0.584 to 0.214 with as little as 100 queries to Logistic Regression models trained on the Fifa19 dataset. The performance of our approach improves as the number of queries $\tilde{Q}$ increases, but stabilizes early at $\tilde{Q} = 100$.

Knowledge of precise marginals. We have so far assumed the adversary to have precise knowledge of the marginals, discretizing them for simplicity into $G = 100$ sub-intervals of equal size. Figure D 4 in the Appendix D.1 shows that decreasing G and, by doing so, reducing the adversary’s ability to sample from the marginals with fine granularity, has a very small impact on the performance of our attack. The accuracy on Fifa19 decreases very slowly from 91.2% ($G = 100$) to 82.9% ($G = 5$), much higher than the model-less attack at 60.2%.

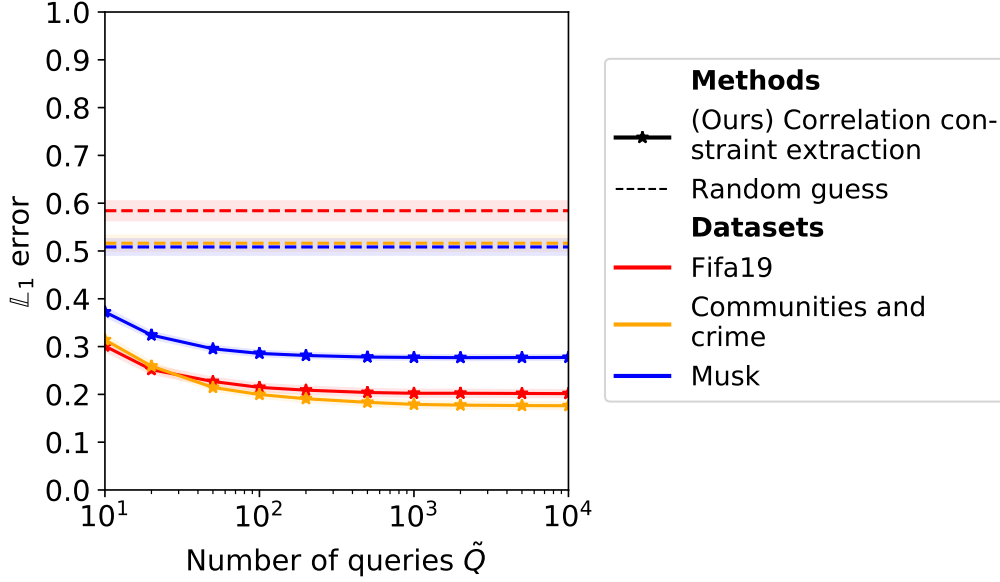


Figure 7.11: **Correlation constraint extraction from LR models for different number of queries $\tilde{Q}$.** We compute the $\mathbb{L}_1$ error between the inferred and the ground-truth values of the correlations: $\frac{1}{n} \sum_{i=1}^n |\hat{\rho}(X_i, Y) - \rho(X_i, Y)|$ (mean with 95% confidence interval).

Knowledge of the model seed. We believe it more realistic to assume that the adversary does not know the seed used to initialize the training algorithm of MLP models, including their weights. This assumption leads to more uncertainty, as the models may end up in different local minima compared to the target model and may behave less similarly than expected. This is not an issue for LR models, as they minimize a convex objective having a global optimum [BCH22]. Figure 7.12A and B compare the performances of black-box and various white-box correlation inference attacks against MLP models, when changing, respectively keeping the seed constant between models used to generate the meta-classifier’s training dataset and target models. We refer the reader Section 7.2.3 for implementation details of the white-box attacks. While the black-box attack gains little from the seed knowledge (up to 4.6 p.p), the white-box attack reaches an astonishing 94.5% for $n = 3$ variables, 88.4 for $n = 6$, and 78.3% for $n = 10$ variables, the last being 29.8 p.p. above the model-less attack. It now also outperforms the attack on LR models for all $n \geq 4$, confirming our hypothesis that more weights (e.g. 432 for the MLP vs 10 for the LR when $n = 10$) can indeed encode more information. Finally, the canonical sorting of weights, proposed by Ganju et al. [GWY⁺18] to enhance the performance of the meta-classifier is, in this setting, somewhat effective when the seed changes. However, its performance is still much lower than the black-box attack. It however achieves the opposite

effect of destroying some of the information present in the raw weights when the seed is kept constant.

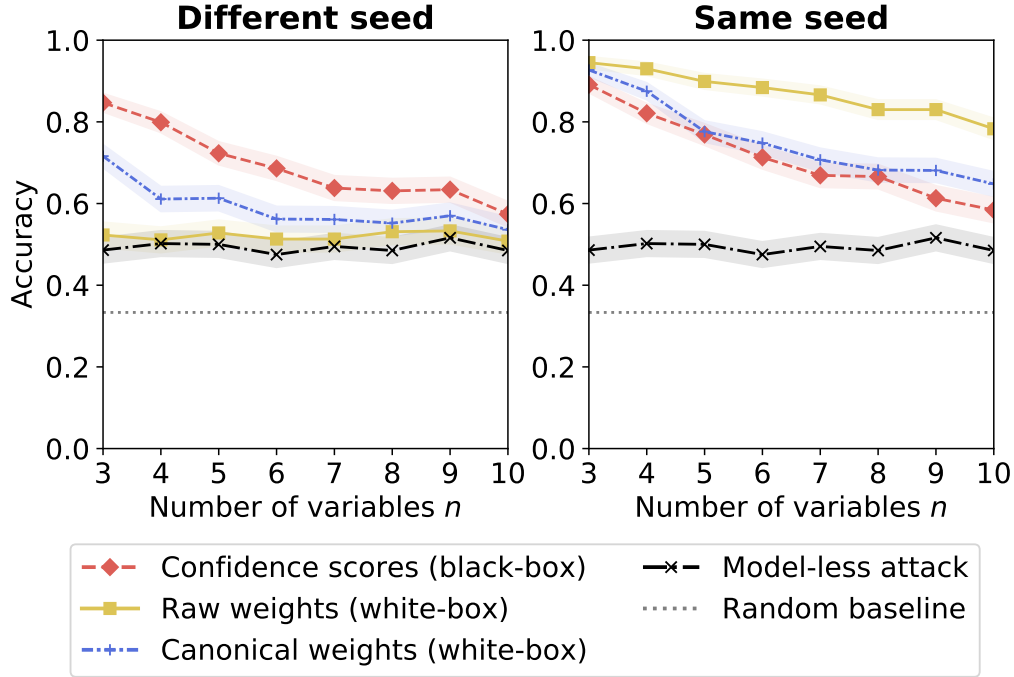


Figure 7.12: Comparison between black-box and white-box CIAs against MLPs when the adversary uses different seeds (left) or the same seed (right) as the target model for training the models used to generate the meta-classifier’s training dataset. We report the mean attack accuracy (with 95% confidence interval).

7.4 Materials and methods

In this section, we describe in detail our methodology for our correlation inference attack (Section 7.4.1) and our correlation inference-based attribute inference attack (Section 7.4.2).

7.4.1 Correlation inference attack

We describe in detail the various building blocks of our methodology for the model-less and model-based attacks. More specifically, we first describe our algorithms for generating correlation matrices under constraints (Section 7.4.1). Second, we describe how we generate synthetic dataset conditionally on a correlation matrix for arbitrary one-way marginals (Section 7.4.1).

Third, we describe the details of our model-based attack (Section 7.4.1).

Generating correlation matrices under constraints

We adapt the algorithm of Numpacharoen and Atsawarungruangkit [NA12], which we include in the Appendix A.1 (Algorithm A 1), to our problem of generating matrices under constraints given by the adversary’s knowledge. We notice that the coefficients of the first column of B ($\cos \theta_{i,1}$ for $i = 2, \dots, n - 1$) are identical to those of C and are free parameters. We start by reordering the variables to $Y, X_1, \dots, X_{n-1}$ to generate $\rho(X_1, X_2)$ first (and explain below why). Under **S2**, we set the coefficients of the first column of B (and C) equal to the correlation constraints $(\rho(X_i, Y))_{i=1, \dots, n-1}$.

This amounts to replacing line 6 from Algorithm A 1 (highlighted in red) with $c_{i,1} \leftarrow \rho(X_{i-1}, Y)$. Under **S1**, we will set the first two coefficients equal to the correlation constraints $\rho(X_1, Y)$ and $\rho(X_2, Y)$, initializing the others uniformly at random within $[-1, 1]$. We replace line 6 from Algorithm A 1 correspondingly.

We then shuffle the variables 3 to $n - 1$ by applying a random permutation. Indeed, naïvely applying Algorithm A 1 would generate the correlations using the default variable ordering (in our case, $Y, X_1, \dots, X_{n-1}$). While the correlation generated first, $\rho(X_1, X_2)$, is uniformly distributed within its boundaries, the correlation generated last, $\rho(X_{n-2}, X_{n-1})$, is not, because the previously sampled correlations restrict its range. More specifically, its conditional distribution follows a bell-shaped curve. Shuffling the variables reduces this bias [NA12]. We choose to keep the target pair X_1 and X_2 in the first position, to guarantee a uniform prior over the distribution of the target correlation $\rho(X_1, X_2)$. Indeed, we assume the adversary does not know how $\rho(X_1, X_2)$ is distributed within its boundaries. Furthermore, we do not want to bias the values towards the center of the interval, as would be the case if the value was not generated first. Algorithm 8 details how we carefully sequence the shuffling, the column reordering, and the inversion of the shuffling for scenario **S2** (similarly, see Algorithm 9 for scenario **S1**).

Algorithm 10 details our procedure to sample the unknown value $\rho(X_1, X_2)$ for scenario **S3**,

Algorithm 8 SAMPLECORRMATRIX-ATTACKS2

```

1: Inputs:
    $n$ : Number of variables.
    $constraints = (\rho(X_i, Y))_{i=1, \dots, n-1}$ : Correlation constraints imposed by the adversary's
   knowledge.
2: Output:
    $C \in \mathbb{R}^{n \times n}$ : A valid correlation matrix satisfying  $c_{i,n} = \rho(X_i, Y)$ , for  $i = 1, \dots, n-1$ .
3: // Randomly shuffle the constraints.
4:  $\sigma \leftarrow [1, 2] + \text{random\_permutation}([3, \dots, n-1])$ 
5:  $constraints \leftarrow \sigma(constraints)$ 
6:  $C \leftarrow \text{SAMPLECORRMATRIX}(n, constraints)$ 
7: // Reorder the variables as  $X_1, \dots, X_{n-1}, Y$ .
8:  $\sigma' \leftarrow [2, \dots, n] + [1]$ 
9:  $C \leftarrow \text{reorder\_columns}(\text{reorder\_rows}(C, \sigma'), \sigma')$ 
10: // Revert the shuffling of constraints.
11:  $\sigma_n \leftarrow \sigma^{-1} + [n]$ 
12:  $C \leftarrow \text{reorder\_columns}(\text{reorder\_rows}(C, \sigma_n), \sigma_n)$ 

```

Algorithm 9 SAMPLECORRMATRIX-ATTACKS1. Code comments are written in blue.

```

1: Inputs:
    $n$ : Number of variables.
    $constraints = (\rho(X_1, Y), \rho(X_2, Y))$ : Correlation constraints imposed by the adversary's
   knowledge.
2: Output:
    $C \in \mathbb{R}^{n \times n}$ : A valid correlation matrix satisfying  $c_{1,n} = \rho(X_1, Y)$  and  $c_{2,n} = \rho(X_2, Y)$ .
3:  $C \leftarrow \text{SAMPLECORRMATRIX}(n, constraints)$ 
4: // Reorder variables as  $X_1, X_2, X_{\sigma(3)}, \dots, X_{\sigma(n-1)}, Y$ .
5:  $\sigma \leftarrow [2, 3] + \text{random\_permutation}([4, \dots, n]) + [1]$ 
6:  $C \leftarrow \text{reorder\_columns}(\text{reorder\_rows}(C, \sigma), \sigma)$ 

```

when all the other correlations are known. We start by placing the target pair (X_1, X_2) in the last position. Denote by C' the partial correlation matrix after reordering the variables (filling the unknown $c'_{n,n-1}$ with a dummy value). We sample the last coefficient $c'_{n,n-1}$ uniformly within its bounds, which we determine by reverse-engineering part of the spherical parametrization B . More specifically, we compute all the low triangular coefficients except for $b_{n,n-1}$ and $b_{n,n}$, which cannot be determined absent knowledge of $c'_{n,n-1}$. This allows us to compute the bounds of $c'_{n,n-1}$ and sample it uniformly within them.

Algorithm 10 SAMPLECORRMATRIX-ATTACKS3

```

1: Inputs:
    $n$ : Number of variables.
    $C$ : Adversary knowledge having  $c_{1,2}$  and  $c_{2,1}$  filled with a dummy value.
2: Output:
    $D \in \mathbb{R}^{n \times n}$ : A valid correlation matrix satisfying  $d_{i,j} = c_{i,j}$ ,
    $\forall (i,j) \in \{1, \dots, n\}^2 \setminus \{(1,2), (2,1)\}$ .
3: // Reorder the variables as  $Y, X_{n-1}, \dots, X_1$ .
4:  $\sigma \leftarrow [n, \dots, 1]$ 
5:  $C' \leftarrow \text{reorder\_columns}(\text{reorder\_rows}(C, \sigma), \sigma)$ 
6:  $B' B'^T \leftarrow \text{CholeskiDecomposition}(C'_{1:n-1, 1:n-1})$  //  $B' \in \mathbb{R}^{(n-1) \times (n-1)}$ 
7:  $B \leftarrow 0$ ;  $B_{1:n-1, 1:n-1} \leftarrow B'$  //  $B \in \mathbb{R}^{n \times n}$ 
8:  $b_{n,1} \leftarrow c'_{n,1}$  // Start filling the last row of  $B$ .
9: for  $i \in \{2, \dots, n-2\}$  do
10:    $b_{n,i} \leftarrow b'_{n,i} - B_{i,1:i}(B_{n,1:i})^T / b_{i,i}$ 
11: end for
12:  $b_{n,n-1} \leftarrow \sqrt{1 - \sum_{j=1}^{n-2} b_{n,j}^2}$ 
13: // Compute the boundaries  $m \pm l$  of  $c'_{n,n-1}$ .
14:  $m, l \leftarrow B_{n-1, 1:n-1}(B_{n,1:n-1})^T, b_{n-1,n-1} b_{n,n-1}$ 
15: // Sample uniformly within the bounds.
16:  $c'_{n,n-1} = c'_{n-1,n} \leftarrow \mathcal{U}(m-l, m+l)$ 
17: // Reorder the variables as  $X_1, \dots, X_{n-1}, Y$ .
18:  $D \leftarrow \text{reorder\_columns}(\text{reorder\_rows}(C', \sigma))$ 

```

Generating synthetic data given a correlation matrix for arbitrary one-way marginals

Algorithm 11 details our heuristic to modify the correlation constraints. More specifically, it determines V' such that, on average, synthetic datasets D generated from matrices matching V' minimize the difference $\max_{(i,j) \in P} |E[C(D)_{i,j}] - (c_T)_{i,j}|$ between their empirical correlations $C(D)$ and the correlation constraints. For conciseness, we focus on the default scenario **S2**, where $V = (\rho(X_i, Y))_{i=1, \dots, n-1}$ (of which scenario **S1** is a particular case).

Model-based attack

Our model-based attack exploits access to a machine learning model $\mathcal{M}_T$ trained on the target dataset D_T in order to extract the target correlation.

First, we generate K correlation matrices $C^1, \dots, C^K$ satisfying V when the marginals are standard normals (and V' for arbitrary marginals). In each scenario, the correlation matrices are sampled using the corresponding algorithm. For each correlation matrix C^k , we generate a

synthetic dataset D^k whose correlations approximately match V and whose marginals match $F_i, i = 1, \dots, n$ as described in Section 7.4.1.

Algorithm 11 MODIFYCORRELATIONCONSTRAINTS

```

1: Inputs:
    $n$ : Number of variables.
    $V = (\rho(X_i, Y))_{i=1, \dots, n-1} \in \mathbb{R}^{n-1}$ : Correlation constraints.
    $(F_i)_{i=1, \dots, n}$ : One-way marginals.
    $S$ : Number of synthetic datasets used to estimate the gap.
    $N_D$ : Size of the synthetic datasets.
    $e$ : Error tolerance.
    $M$ : Maximum number of iterations.

2: Output:
    $V' \in \mathbb{R}^{n-1}$ : Modified correlation constraints.

3: Initialize:
    $V' \leftarrow V$ 

4: for  $m = 1, \dots, M$  do
5:    $\bar{V} \leftarrow \text{zeros}(n-1)$ 
6:   for  $s = 1, \dots, S$  do
7:      $C \leftarrow \text{SAMPLECORRMATRIX-ATTACKS2}(n, V')$ 
8:     // Synthetic dataset whose correlations approximately match  $V'$ .
9:      $D^s \leftarrow [\text{SAMPLEFROMGAUSSIANCOPULAS}(\$ 
10:  $n, C, F_1, \dots, F_n)$  for  $d = 1, \dots, N_D]$ 
11:      $C^s \leftarrow \text{correlation\_matrix}(D^s)$ 
12:      $\bar{V}_i \leftarrow \bar{V}_i + \frac{C_{i,n}^s}{S}$  for  $i = 1, \dots, n-1$ 
13:   end for
14:    $\text{gap} \leftarrow \max_{i=1, \dots, n-1} |\bar{V}_i - V_i|$ 
15:   if  $\text{gap} < e$  then
16:     break
17:   else
18:     // Update the constraints.
19:      $V'_i \leftarrow \max(-1, \min(1, V'_i + \frac{V_i - \bar{V}_i}{2}))$  for  $i = 1, \dots, n-1$ 
20:   end if
21: end for

```

Second, we train a model $\mathcal{M}^k$ on dataset D^k , for every $k = 1, \dots, K$. The models are trained using the same algorithm and hyperparameters as the target model, but a different seed. We present in Section 7.3 results of our attack when the adversary knows the seed.

Third, we generate a synthetic *query dataset* D_{query} in the same way as above, and use it to extract black-box features from the models. More specifically, after the K models have been trained, we pass the records of D_{query} through the model $\mathcal{M}^k$ for $k = 1, \dots, K$, and retrieve the confidence scores for the first $L-1$ classes (the last one is redundant with the others as they all

sum to 1). Denoting by Q the size of D_{query} , we thus extract a total of $Q \times (L - 1)$ features from each model. In this chapter, we assume for simplicity that the models are binary classifiers ($L = 2$), meaning that we extract Q features from each model. We concatenate them to form the *output feature vectors* $O_k, l = 1, \dots, K$. Our hypothesis is that these features are likely to encode information about the dataset correlations, e.g. that a model makes more confident predictions on records resembling its own training dataset.

Fourth, given a dataset D , we denote by $C(D)$ the correlation matrix computed on the dataset. We label each feature vector O_k with the correlation $L_k = C(D^k)_{1,2}$ of the corresponding synthetic dataset, discretized over B classes. We then train a meta-classifier $\mathcal{A} : \mathbb{R}^Q \rightarrow \{1, \dots, B\}$ on $D_{\text{meta}} = \{(O_k, L_k), k = 1, \dots, K\}$ to infer the correlation given features extracted from the model.

Finally, we deploy the meta-classifier on the target model $\mathcal{M}_T$. The adversary's guess is the output class $\mathcal{A}(O_T)$ predicted by the meta-classifier on the feature vector O_T , consisting of the outputs of $\mathcal{M}_T$ on the query dataset D_{query} .

7.4.2 Attribute inference attack

We describe our attribute inference attack aiming to infer the sensitive attribute value x_1 of a target record $(x_1, \dots, x_{n-1}, y) \in D_T$ using black-box access to a model $\mathcal{M}_T$ trained on a private dataset D_T . Our method consists of three steps:

Step 1: We execute our correlation inference attack independently against each target pair $(X_i, X_j), 1 \leq i < j \leq n - 1$ under default scenario **S2**. The output of the attack is a correlation bin $b_{i,j} \in \{1, \dots, N_B\}$. To evaluate our attack on real-world datasets having arbitrary marginals, we run the attack using the `MODIFYCORRELATIONCONSTRAINTS` heuristic. We denote by $V'_{i,j} \in \mathbb{R}^{n-1}$ the modified correlation constraints returned by the heuristic.

Step 2: We sample synthetic datasets whose empirical correlations belong to the inferred bins, while also matching the correlation constraints $(\rho(X_i, Y))_{i=1, \dots, n-1}$. Sampling synthetic data under constraints given by the adversary knowledge requires to modify the correlation con-

straints $\rho(X_i, Y), i = 1, \dots, n - 1$, as described in Section 7.4.1. We here use the *average* of modified constraints used previously to attack each of the pairs: $V' = \frac{2}{(n-1)(n-2)} \sum_{1 \leq i < j \leq n-1} V'_{i,j}$. We generate S' synthetic datasets under these modified constraints, then select among them the datasets D whose empirical correlations $C(D)_{i,j}$ belong to the inferred bin $b_{i,j}$, for every $1 \leq i < j \leq n - 1$, discarding the rest. We concatenate the datasets together into a larger dataset D_{synth} .

Step 3: We select all the values of the sensitive attribute $x_1^1, \dots, x_1^H$ belonging to records that approximately match the partial record $(x_2, \dots, x_{n-1}, y)$. We search for approximate matches as follows. First, we discretize the marginals of non-sensitive variables $X_2, \dots, X_{n-1}$ into G sub-intervals of same size and denote by $b_2, \dots, b_{n-1}$ the size of the sub-intervals. Second, we retrieve all the records $(x'_1, \dots, x'_{n-1}, y')$ in D_{synth} such that: (1) $|x_i - x'_i| \leq m_i b_i, i = 2, \dots, n - 1$ and (2) $y = y'$, where m_i denotes the resolution of the search around the partial record's attribute x_i . For instance, for $m_2 = 1$ and $b_2 = 0.2$ we only select records having $x'_2 \in [x_2 - 0.2, x_2 + 0.2]$. If no record is found, we increase the resolution parameters by increments δ_i : $m_i \leftarrow m_i + \delta_i, i = 2, \dots, n - 1$, until at least one record is found. We return the average sensitive attribute of matched records $\hat{x}_1 = \frac{1}{H}(\sum_{h=1}^H x_1^h)$ as our prediction. We also explore ablations of our attack using different formulas for predicting x_1 given the partial record $(x_2, \dots, x_{n-1}, y)$ and the synthetic data generated under Step 2, including variations of Jayaraman and Evans's [JE22] formula, none of which significantly improve the attack's performance. Summary of the results can be found in Tables 7.2 and 7.3.

As for the $X_1 - Y$ **correlation** baseline, we run steps 2 and 3 of our CI-AIA approach, described above, on variables X_1 and Y , but not step 1 as there is no need to extract additional correlations from the model.

7.5 Related work

Information leakage by ML models. It has, so far, been mostly studied through the lens of *individual records*. This includes membership inference attacks [SSSS17, SZH⁺18, YGFJ18,

NSH19, CCTCP21, CCN⁺22] aiming to infer if a record was used to train a model – and to a lesser extent, attribute inference [FLJ⁺14, FJR15, MDDV⁺22] and reconstruction [SBB⁺20, BCH22] attacks – aiming to reconstruct respectively, partially and entirely, a record. While potentially very invasive from a privacy perspective, attacks against individual-level records commonly rely on powerful adversaries, with partial [NSH19] or near-complete [BCH22] knowledge of the dataset, or with access to a dataset sampled from the same distribution. We here focus on an adversary aiming to infer correlations without access to real records, which requires rethinking the current design of inference attacks. In particular, we develop a model-less attack as a strong baseline and use Gaussian copulas to sample synthetic tabular datasets that we use to generate the meta-classifier’s training dataset.

Property inference attacks (PIA). PIAs study the leakage of macro properties of the dataset. Ateniese et al. [AMS⁺15] introduced property inference attacks (PIAs), albeit by another name, and applied them to Hidden Markov Models and Support Vector Machines. They show, for instance, that it is possible to distinguish the dialect on which a speech recognition model was trained. To this end, they train a meta-classifier on *white-box* features extracted from models trained in the same way as the target model, a methodology now known as *shadow modeling* [SSSS17]. Ganju et al. [GWY⁺18] showed that extending this attack to multilayer perceptrons (MLP) is more challenging due to their permutation equivalence property and propose to use weight-based neuron sorting and set-based meta-classifiers; we use the former in our white-box experiments on MLPs. Zhang et al. [ZTO21] proposed the first *black-box* property inference attack, making use of the model’s output probabilities computed on a dataset of real records from the same distribution. Finally, Suri and Evans [SE22] proposed a formal definition of distribution inference attacks that unifies the property inference attacks (PIA) of previous works, which focused on the leakage of the proportion of samples along a certain attribute. The authors extend the definition to other properties such as the average node degree of a graph.

Our work differs substantially from previous work in PIAs. First, we focus, by design, on explicit micro characteristics of the dataset X rather than on external macro properties of the record (e.g., the gender label of an image). Second, we demonstrate how dataset-level information, here

correlations, can be used as building blocks for more accurate attribute inference attacks. Third, our attack, contrary to PIAs, does not require access to a subset of the dataset or, at least, a dataset sampled from a similar distribution. This would indeed mean that the adversary could readily compute the correlations or, at least, have a very good prior on them. We here do not assume access to private data for approximating the distribution, developing a methodology for inferring correlations which relies on synthetic data instead of private data. More specifically, the models we use to generate the training dataset of the meta-classifier are trained exclusively on synthetic records. This makes the attack accessible to weaker adversaries than those of previous works. Fourth, previous works mainly focus on a categorical choice between *very distinct* properties. For instance, Ganju et al. [GWY⁺18] aim to infer whether the proportion of older faces is equal to (precisely) 23% or 37% or if exactly 0% or 87% of the faces were from white people [GWY⁺18]. By contrast, we frame the correlation inference task as a prediction task *covering the entire range of possible values*. While Zhou et al. [ZCSZ21] aim to address this shortcoming by framing property inference as a regression task, they only apply their attack to Generative Adversarial Networks which would, by design, reflect the statistical properties of the training dataset making a correlation attack trivial.

Attribute inference attacks (AIA). AIAs aim to recover the sensitive attribute of a target record. Friedrikson et al. [FLJ⁺14] proposed the first AIA against ML models. They applied their attack to linear regression models available as a black-box and aimed to infer a patient’s genotype. Subsequent work [FJR15] extended this attack to make use of confidence scores and applied it to decision trees. Yeom et al. [YGFJ18] explored the connection between membership and attribute inference and proposed a black-box attack relying on a membership inference oracle. Mehnaz et al. [MDDV⁺22] then proposed new black-box attacks: (1) a confidence-based attack dropping the prior work’s assumption that the adversary knows the one-way marginals of the dataset and (2) a label-based attack assuming the adversary has access to a large fraction of the training dataset (excluding the sensitive attribute).

In contemporaneous work, Jayaraman and Evans [JE22] propose two extensions of the black-box AIA of Yeom et al. [YGFJ18] and a white-box AIA. Our black-box correlation inference-based

AIA (CI-AIA) is fundamentally different from their black-box AIAs, because we study a weaker adversary who does not know the data distribution via access to auxiliary data. Instead, we (1) extract information about the distribution from the model (the correlation between the input variables), (2) use this information to generate synthetic data, and (3) predict the sensitive attribute using the synthetic data. Table 7.2 shows our CI-AIA to outperform their black-box AIAs when compared fairly in our weak adversary setting. We further note that our step (3) applied to synthetic data bears similarity with Jayaraman and Evans’s [JE22] prediction rule applied to auxiliary data. To understand if their prediction rule can further improve our CI-AIA performance, we substituted their prediction rule to step (3) of our attack while leaving our steps (1) and (2) unchanged. We show in Table 7.3 that this does not lead to better performance than our CI-AIA.

Synthetic data-based shadow modeling. It is an under-explored research area. Shokri et al. [SSSS17] were the first to use synthetic data to train shadow models. They developed a hill-climbing algorithm to generate synthetic records. Their approach is based on the intuition that a model would be more confident on records that are similar to the training dataset. Salem et al. [SZH⁺18] argued that this method is only efficient for datasets of binary records and explored the use of shadow datasets from different sources. This is however not a suitable option when aiming to infer summary statistics of a dataset as it requires shadow datasets to span the entire range of possible values for the statistics.

Chapter 8

Conclusion

8.1 Summary of thesis achievements

In the adversarial definition of regulation laws, such as GDPR [Eur16], it is stated that any means “reasonably likely” used to de-identify someone’s data have to be taken into consideration when releasing an anonymous dataset. So far, it was tempting to dismiss privacy attacks based on the unrealisticness of their core assumptions. This thesis aims to provide a more realistic privacy evaluation. We first provided an identification of vulnerable individuals in the context of synthetic datasets, second we provided an evaluation setup to assess the privacy risk presented by a specific data release, and third we constructed attacks without access to auxiliary datasets in different settings.

In Chapter 2 we present the necessary information to formally define the notions used throughout the thesis.

In Chapter 3, we provide a better method for outlier identification in the context of synthetic datasets. Our method is, to the best of our knowledge, the first metric based at-risk record detection. To provide empirical evidence that this result holds for various datasets, attacks and generators, we also introduce a new attention-based attack against synthetic generators. After extensive evaluation, we show our results to be consistent across attacks, datasets and

generators.

In Chapter 4, we first explain why the common method to evaluate the privacy of various mechanisms can provide an inaccurate estimation of the privacy risk. Indeed, traditional evaluation uses dataset sampling to ensure a good evaluation, as it is naturally an extension from ML models evaluation to MIAs. We present evidences that this provides an inaccurate estimation of the privacy risk because it averages the privacy risk across the datasets. We provide a different, *specific* evaluation setup. To provide a risk estimate specific to the risk tied to a given data release. We extensively evaluate this setup, across different attacks, datasets, and data release mechanisms.

In the remaining chapters, we show that we can remove the assumption that the adversary has access to an auxiliary dataset with the same underlying distribution as the target dataset. This unrealistic assumption has been tackled in three different setups.

First, in Chapter 5 we study synthetic data generators, and their vulnerabilities. We create three different adversaries, classified by the strength of their assumptions. The first uses access to the generator to query as many records as needed, while the second uses the produced and released synthetic dataset only. We then identify the *double counting problem*, which explains why using only the released synthetic dataset leads to a small decrease in performances. After identification, we developed a third adversary that avoids this problem, and we show that it outperforms the other adversaries. This chapter presents, to the best of our knowledge, the first attack based solely on the output of one synthetic generator.

Second, in Chapter 6 we consider the case when a data release mechanism provides indirect access to the target record, using an aggregation mechanism to “hide” each record’s contribution. We show that by leveraging only access to the aggregate, the adversary is able to achieve the same performance as the state-of-the-art attack, which is based upon the unrealistic assumption of the attacker having access to auxiliary data. We also show that even if the aggregate is released under privacy preserving defenses, our attack is able to denoise the aggregate to achieve great performances.

Third, in Chapter 7 we consider the case when a machine learning model is released. We show that not having access to any kind of auxiliary data enables new kinds of privacy attacks, such as correlation inference attacks. The goal of this attack is to infer the correlation matrix of the target dataset from a machine learning model trained on it. We show that our attack is successful across three different scenarios that vary the level of auxiliary knowledge available to the adversary. We then use the inferred information to construct a better attribute inference attack. We were able to use the inferred information about the correlation matrix, to build a better prior for the adversary and greater performances. We show that privacy of dataset level properties matters, both for the privacy of dataset-level properties but also for the record-level privacy, as dataset level properties can be used to further improve the prior of adversaries.

8.2 Limitations and future work

In this thesis, we carefully design experiments and methodologies to construct more realistic privacy adversaries against three different types of data release mechanisms. We here discuss the potential limitations and future research that can be built upon the work presented in this thesis.

Limitations:

1. **Most vulnerable records.** In this work, we study the identification and attack of at-risk records. Further work could build on this to analyze the effect that the exclusion of these records from the data could have. Previous work on MIAs against ML models tends to show that when we remove the most vulnerable records, other records take their place as most vulnerable [CJZ⁺22]. Future work could explore if this is still true in the context of synthetic datasets, as studied in Chapter 3, 4 and 5.
2. **Empirical evaluation.** One of the results we present in this thesis tells us how sometimes results are specific to the release considered. Here, one limitation of this thesis is that we validated results on widely used synthetic data generators or attacks, future work could

explore if our results hold true for different specific systems such as GANs. Similarly, we introduce an attack without auxiliary dataset against location aggregates. Future work could explore the application of our technique to different kinds of aggregate, such as survey data.

3. **Protection against privacy attacks.** Results presented in this thesis focus on attacks against different data release mechanisms. One limitation of this approach is that we did not study in depth the extent of protection available against such attacks, often restricting our analysis to the introduction of DP. Whether or not additional protection mechanisms exist to prevent the presented attacks is left for future work.

Future work:

1. **Generalization to privacy adversaries in other settings.** The work presented in this thesis focused on using generative models to relax the unrealistic assumption of the adversary having access to an auxiliary dataset from the assumptions of privacy adversaries. We developed this approach for membership inference attacks, attribute inference attacks and correlation inference attacks. It is left for future work whether this approach can be developed for further specific attacks, such as Reconstruction Attacks. Furthermore, our newly defined specific attack, introduced in Chapter 4, has thus far been studied primarily in the context of membership inference attacks. Future work could explore the extent to which this approach can be generalized to other types of privacy attacks.
2. **Metric to quantify quality of synthetic datasets.** In the context of synthetic datasets, we validate their utility by verifying the preservation of the global statistics of the dataset, and evaluating how useful the synthesized datasets are for training ML models. Although these metrics are commonly used in state-of-the-art papers on synthetic datasets [TMH⁺21, AGR24], we believe that they could still be improved. It is left for future work to explore how it is possible to create a widely applicable, computationally efficient metric to better assess the utility of synthesized datasets.

As we conclude this thesis, we hope it helps the reader to become familiarized with privacy adversaries, and convinced that those adversaries are real threats that should not be underestimated. We believe that MIAs and other privacy adversaries should be considered as reasonably likely to be deployed, and that, when releasing a dataset by any means, the data owner should be very cautious of the privacy threats that may arise.

Bibliography

- [aA17] Siri Team at Apple. Deep learning for siri’s voice: On-device deep mixture density networks for hybrid unit selection synthesis. <https://machinelearning.apple.com/research/siri-voices>, 2017.
- [ABC⁺20] Ahmet Aktay, Shailesh Bavadekar, Gwen Cossoul, John Davis, Damien Desfontaines, Alex Fabrikant, Evgeniy Gabrilovich, Krishna Gadepalli, Bryant Gipsen, Miguel Guevara, et al. Google covid-19 community mobility reports: anonymization process description (version 1.1). *arXiv preprint arXiv:2004.04145*, 2020.
- [AGR23] Meenatchi Sundaram Muthu Selva Annamalai, Andrea Gadotti, and Luc Rocher. A linear reconstruction approach for attribute inference attacks against synthetic data. *arXiv preprint arXiv:2301.10053*, 2023.
- [AGR24] Meenatchi Sundaram Muthu Selva Annamalai, Andrea Gadotti, and Luc Rocher. A linear reconstruction approach for attribute inference attacks against synthetic data. *USENIX Association*, 2024.
- [AK21] Ali Alvi and Paresh Kharya. Using deepspeed and megatron to train megatron-turing nlg 530b, the world’s largest and most powerful generative language model. <https://www.microsoft.com/en-us/research/blog/using-deepspeed-and-megatron-to-train-megatron-turing-nlg-530b-the-worlds-largest-generative-language-model/>, 2021.

- [Ala22] Alan Turing Institute. Resprosyn. <https://github.com/alan-turing-institute/reprosyn>, 2022.
- [All23] Tristan Allard. Snake challenge. <https://www.codabench.org/competitions/879/>, 2023.
- [AMS⁺15] Giuseppe Ateniese, Luigi V Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks*, 10(3):137–150, 2015.
- [App17] D Apple. Learning with privacy at scale. *Apple Machine Learning Journal*, 1(8), 2017.
- [AZT24] Michael Aerni, Jie Zhang, and Florian Tramèr. Evaluations of machine learning privacy defenses are misleading. *arXiv preprint arXiv:2404.17399*, 2024.
- [BB20] Luke A Bauer and Vincent Bindschaedler. Towards realistic membership inferences: The case of survey data. In *Annual Computer Security Applications Conference*, pages 116–128, 2020.
- [BCH22] Borja Balle, Giovanni Cherubin, and Jamie Hayes. Reconstructing training data with informed adversaries. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1138–1156. IEEE, 2022.
- [BDNL⁺15] Gianni Barlacchi, Marco De Nadai, Roberto Larcher, Antonio Casella, Cristiana Chitic, Giovanni Torrisi, Fabrizio Antonelli, Alessandro Vespignani, Alex Pentland, and Bruno Lepri. A multi-source dataset of urban life in the city of milan and the province of trentino. *Scientific data*, 2(1):1–15, 2015.
- [BDR19] Steven M Bellovin, Preetam K Dutta, and Nathan Reiter. Privacy and synthetic datasets. *Stan. Tech. L. Rev.*, 22:1, 2019.

- [BG18] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018.
- [BK23] Michelle Boorstein and Heather Kelly. Colorado catholic group bought app data that tracked gay priests. *The Washington Post*, March 2023. <https://www.washingtonpost.com/dc-md-va/2023/03/09/catholics-gay-priests-grindr-data-bishops/>.
- [BN06] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.
- [BSDM19] AG Bates, I Spakulová, I Dove, and A Meador. On methodology working paper series number 16—synthetic data pilot. <https://www.ons.gov.uk/methodology/methodologicalpublications/generalmethodology/onsworkingpapersseries/onsmethodologyworkingpapersseriesnumber16syntheticdatapilot>. 2019. Accessed on 02/06/2023.
- [BUK19] PRESTON BUKATY. *The California Consumer Privacy Act (CCPA): An implementation guide*. IT Governance Publishing, 2019. <http://www.jstor.org/stable/j.ctvjghvnn>.
- [CBT⁺13] Balázs Cs Csáji, Arnaud Browet, Vincent A Traag, Jean-Charles Delvenne, Etienne Huens, Paul Van Dooren, Zbigniew Smoreda, and Vincent D Blondel. Exploring the mobility of mobile phone users. *Physica A: statistical mechanics and its applications*, 392(6):1459–1473, 2013.
- [CCN⁺22] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914, 2022.
- [CCTCP21] Christopher A Choquette-Choo, Florian Tramèr, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International conference on machine learning*, pages 1964–1974. PMLR, 2021.

- [CEW⁺18] Hongming Chen, Ola Engkvist, Yinhai Wang, Marcus Olivecrona, and Thomas Blaschke. The rise of deep learning in drug discovery. *Drug discovery today*, 23(6):1241–1250, 2018.
- [CGdM24] Ana-Maria Crețu, Florent Guépin, and Yves-Alexandre de Montjoye. Correlation inference attacks against machine learning models. *Science Advances*, 10(28):eadj9260, 2024.
- [CHCdM22] Ana-Maria Cretu, Florimond Houssiau, Antoine Cully, and Yves-Alexandre de Montjoye. Querysnout: Automating the discovery of attribute inference attacks against query-based systems. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 623–637, 2022.
- [CHM⁺15] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Artificial intelligence and statistics*, pages 192–204. PMLR, 2015.
- [CJdMT23] Ana-Maria Cretu, Daniel Jones, Yves-Alexandre de Montjoye, and Shruti Tople. Investigating the effect of misalignment on membership privacy in the white-box setting. *arXiv preprint arXiv:2306.05093*, 2023.
- [CJZ⁺22] Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramèr. The privacy onion effect: Memorization is relative. *Advances in Neural Information Processing Systems*, 35:13263–13276, 2022.
- [CKL⁺09] Bee-Chung Chen, Daniel Kifer, Kristen LeFevre, Ashwin Machanavajjhala, et al. Privacy-preserving data publishing. *Foundations and Trends® in Databases*, 2(1–2):1–167, 2009.
- [CMS11] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.
- [Com14] European Commission. Article 29 data protection working party opinion 05/2014

- on anonymisation techniques. https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf, 2014.
- [Com20] European Commission. On artificial intelligence-a european approach to excellence and trust. https://ec.europa.eu/info/sites/default/files/commission-white-paper-artificial-intelligence-feb2020_en.pdf, 2020.
- [CS21] Hongyan Chang and Reza Shokri. On the privacy risks of algorithmic fairness. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 292–303. IEEE, 2021.
- [CTW⁺21] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *USENIX Security Symposium*, volume 6, 2021.
- [D⁺34] Boris Delaunay et al. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800):1–2, 1934.
- [DCM⁺22] Zizhuang Deng, Kai Chen, Guozhu Meng, Xiaodong Zhang, Ke Xu, and Yao Cheng. Understanding real-world threats to deep learning models in android apps. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 785–799, 2022.
- [Des21] Damien Desfontaines. A list of real-world uses of differential privacy, 2021.
- [DFRSC15] Josep Domingo-Ferrer, Sara Ricci, and Jordi Soria-Comas. Disclosure risk assessment via record linkage by a maximum-knowledge attacker. In *2015 13th Annual Conference on Privacy, Security and Trust (PST)*, pages 28–35. IEEE, 2015.
- [Die98] T. Dietterich. UCI Repository of machine learning databases: Musk (Version 2) Data Set. [https://archive.ics.uci.edu/ml/datasets/Musk+\(Version+2\)](https://archive.ics.uci.edu/ml/datasets/Musk+(Version+2)), 1998.

- [DKM⁺06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology-EUROCRYPT 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28-June 1, 2006. Proceedings 25*, pages 486–503. Springer, 2006.
- [DKM19] Cynthia Dwork, Nitin Kohli, and Deirdre Mulligan. Differential privacy in practice: Expose your epsilons! *Journal of Privacy and Confidentiality*, 9(2), 2019.
- [dMGB⁺18] Yves-Alexandre de Montjoye, Sébastien Gambs, Vincent Blondel, Geoffrey Canright, Nicolas De Cordes, Sébastien Deletaille, Kenth Engø-Monsen, Manuel Garcia-Herranz, Jake Kendall, Cameron Kerry, et al. On the privacy-conscientious use of mobile phone data. *Scientific data*, 5(1):1–6, 2018.
- [dMHVB13] 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(1):1–5, 2013.
- [DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.
- [DN03] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210, 2003.
- [Dr 23] Dr Henrike Mueller, Leo Gosland, Owen Courtney, Pavle Avramovic, Valerie Marshall. Can synthetic data enable data sharing in financial services? <https://oecd.ai/en/wonk/synthetic-data-financial-services>, 2023.
- [Dwo06] Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II 33*, pages 1–12. Springer, 2006.

- [Eur16] General data protection regulation. <https://gdpr-info.eu/>, 2016.
- [EYL⁺20] Darren Edge, Weiwei Yang, Kate Lytvynets, Harry Cook, Claire Galez-Davis, Hannah Darnton, and Christopher M White. Design of a privacy-preserving data platform for collaboration against human trafficking. *arXiv preprint arXiv:2005.05688*, 2020.
- [Fel20] Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020.
- [FHdM21] Ali Farzanehfar, Florimond Houssiau, and Yves-Alexandre de Montjoye. The risk of re-identification remains high even in country-scale location datasets. *Patterns*, 2(3):100204, 2021.
- [Fin22] Financial Conduct Authority. Synthetic data to support financial services innovation. <https://www.fca.org.uk/publication/call-for-input/synthetic-data-to-support-financial-services-innovation.pdf>, 2022. Accessed on 02/06/2023.
- [FJR15] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.
- [FLJ⁺14] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 17–32, 2014.
- [Flo24] Flowminder. Flowminder website. <https://www.flowminder.org/>, 2024.
- [FZ20] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems*, 33:2881–2891, 2020.

- [Gad19] K. Gadiya. FIFA 19 complete player dataset. <https://www.kaggle.com/datasets/karangadiya/fifa19>, 2019.
- [Gar18] Gary Benedetto, Jordan C. Stanley, and Evan Totty. The creation and use of the sipp synthetic beta v7.0. <https://www.census.gov/library/working-papers/2018/adrm/SIPP-Synthetic-Beta.html>, 2018.
- [GARL16] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. Deep image retrieval: Learning global representations for image search. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VI 14*, pages 241–257. Springer, 2016.
- [GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [GBWT22] Matteo Giomi, Franziska Boenisch, Christoph Wehmeyer, and Borbála Tasnádi. A unified framework for quantifying privacy risk in synthetic data. *arXiv preprint arXiv:2211.10459*, 2022.
- [GdBK19] Yola Georgiadou, Rolf A de By, and Ourania Kounadi. Location privacy in the wake of the gdpr. *ISPRS international journal of geo-information*, 8(3):157, 2019.
- [GF16] Qian Ge and Daisuke Fukuda. Updating origin–destination matrices with aggregated data of gps traces. *Transportation Research Part C: Emerging Technologies*, 69:291–312, 2016.
- [GHR⁺19] Andrea Gadotti, Florimond Houssiau, Luc Rocher, Benjamin Livshits, and Yves-Alexandre De Montjoye. When the signal is in the noise: Exploiting diffix’s sticky noise. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1081–1098, 2019.
- [GMC⁺20] Kyra H Grantz, Hannah R Meredith, Derek AT Cummings, C Jessica E Metcalf, Bryan T Grenfell, John R Giles, Shruti Mehta, Sunil Solomon, Alain Labrique, Nishant Kishore, et al. The use of mobile phone data to inform analysis of covid-19 pandemic epidemiology. *Nature communications*, 11(1):4961, 2020.

- [GMCdM23] 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.
- [Gre21] Graham Greenleaf. Global data privacy laws 2021: Despite covid delays, 145 laws show gdpr dominance, 2021.
- [GWY⁺18] Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 619–633, 2018.
- [Haz19] Hazy. Synthpop. <https://github.com/hazy/synthpop>, 2019.
- [HBMAL19] Naoise Holohan, Stefano Braghin, Pól Mac Aonghusa, and Killian Levacher. Diff-privlib: the IBM differential privacy library. *ArXiv e-prints*, 1907.02444 [cs.CR], July 2019.
- [HBR13] Alastair Holmes, Angela Byrne, and Jennifer Rowley. Mobile shopping behaviour: insights into attitudes, shopping process involvement and location. *International Journal of Retail & Distribution Management*, 42(1):25–39, 2013.
- [HD21] A Herdağdelen and A Dow. Protecting privacy in facebook mobility data during the covid-19 response (2020). <https://research.fb.com/blog/2020/06/protecting-privacy-in-facebook-mobility-data-during-the-covid-19-response>, 2021.
- [HJC⁺22] 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*, 2022.
- [HMDDC19] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan:

- Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019:133–152, 01 2019.
- [Hop21] Christopher Hope. Millions ‘unwittingly tracked’ by phone after vaccination to see if movements changed. <https://www.telegraph.co.uk/politics/2021/05/22/millions-unwittingly-tracked-phone-vaccination-see-movements/>, 2021.
- [HOT⁺23] Thomas Humphries, Simon Oya, Lindsey Tulloch, Matthew Rafuse, Ian Goldberg, Urs Hengartner, and Florian Kerschbaum. Investigating membership inference attacks under data dependencies. In *2023 IEEE 36th Computer Security Foundations Symposium (CSF)*, pages 473–488. IEEE, 2023.
- [HSR⁺08] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS genetics*, 4(8):e1000167, 2008.
- [HY21] Yusuke Hara and Hiromichi Yamaguchi. Japanese travel behavior trends and change under covid-19 state-of-emergency declaration: Nationwide observation by mobile phone location data. *Transportation Research Interdisciplinary Perspectives*, 9:100288, 2021.
- [HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [JE19] Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *USENIX Security Symposium*, 2019.
- [JE22] Bargav Jayaraman and David Evans. Are attribute inference attacks just imputation? In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 1569–1582, 2022.

- [JUO20] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private sgd? *Advances in Neural Information Processing Systems*, 33:22205–22216, 2020.
- [JWA⁺20] Benjamin Jeffrey, Caroline E Walters, Kylie EC Ainslie, Oliver Eales, Constanze Ciavarella, Sangeeta Bhatia, Sarah Hayes, Marc Baguelin, Adhiratha Boonyasiri, Nicholas F Brazeau, et al. Anonymised and aggregated crowd level mobility data from mobile phones suggests that initial compliance with covid-19 social distancing interventions was high and geographically consistent across the uk. *Wellcome Open Research*, 5, 2020.
- [JYVDS19] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2019.
- [JZGR16] Karim Keramat Jahromi, Matteo Zignani, Sabrina Gaito, and Gian Paolo Rossi. Simulating human mobility patterns in urban areas. *Simulation Modelling Practice and Theory*, 62:137–156, 2016.
- [KAB23] Nitin Kohli, Emily Aiken, and Joshua Blumenstock. Privacy guarantees for personal mobility data in humanitarian response. *arXiv preprint arXiv:2306.09471*, 2023.
- [Kak22] Saif Kakakhel. Optimising urban planning with location intelligence. Quadrant Blog: <https://www.quadrant.io/blog/how-to-optimize-urban-planning-with-location-intelligence>, October 2022. Accessed: 2024-03-07.
- [KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [KG17] Vaibhav Kulkarni and Benoît Garbinato. Generating synthetic mobility traffic using rnns. In *Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery*, pages 1–4, 2017.

- [KH⁺09] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. <https://www.cs.toronto.edu/~kriz/cifar.html>, 2009.
- [KLBV07] Thomas Karagiannis, Jean-Yves Le Boudec, and Milan Vojnović. Power law and exponential decay of inter contact times between mobile devices. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 183–194, 2007.
- [KTVG18] Vaibhav Kulkarni, Natasa Tagasovska, Thibault Vatter, and Benoit Garbinato. Generative models for simulating mobility trajectories. *arXiv preprint arXiv:1811.12801*, 2018.
- [LF20] Klas Leino and Matt Fredrikson. Stolen memories: Leveraging model memorization for calibrated white-box membership inference. In *29th USENIX security symposium (USENIX Security 20)*, pages 1605–1622, 2020.
- [LHK⁺09] Kyunghan Lee, Seongik Hong, Seong Joon Kim, Injong Rhee, and Song Chong. Slaw: A new mobility model for human walks. In *IEEE INFOCOM 2009*, pages 855–863. IEEE, 2009.
- [LIE⁺23] Guillaume Leclerc, Andrew Ilyas, Logan Engstrom, Sung Min Park, Hadi Salman, and Aleksander Madry. FFCV: Accelerating training by removing data bottlenecks. In *Computer Vision and Pattern Recognition (CVPR)*, 2023. <https://github.com/libffcv/ffcv/>.
- [LQS⁺13] Ninghui Li, Wahbeh Qardaji, Dong Su, Yi Wu, and Weining Yang. Membership privacy: A unifying framework for privacy definitions. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 889–900, 2013.
- [LTS⁺20] Eyal Lotan, Charlotte Tschider, Daniel K Sodickson, Arthur L Caplan, Mary Bruno, Ben Zhang, and Yvonne W Lui. Medical imaging and privacy in the era of artificial intelligence: myth, fallacy, and the future. *Journal of the American College of Radiology*, 17(9):1159–1162, 2020.

- [LWB⁺20] Yunhui Long, Lei Wang, Diyue Bu, Vincent Bindschaedler, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. A pragmatic approach to membership inferences on machine learning models. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 521–534. IEEE, 2020.
- [LWY19] Pei-Hsuan Lu, Pang-Chieh Wang, and Chia-Mu Yu. Empirical evaluation on synthetic data generation with generative adversarial network. In *Proceedings of the 9th International Conference on Web Intelligence, Mining and Semantics*, pages 1–6, 2019.
- [M15] Meinard Müller. Logarithmic compression. https://www.audiolabs-erlangen.de/resources/MIR/FMP/C3/C3S1_LogCompression.html, 2015.
- [MAR04] Bertrand David Miguel Alonso and Gaël Richard. Tempo and beat estimation of musical signals. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR), Barcelona, Spain, 2004*.
- [MDDV⁺22] Shagufta Mehnaz, Sayanton V Dibbo, Roberta De Viti, Ehsanul Kabir, Björn B Brandenburg, Stefan Mangard, Ninghui Li, Elisa Bertino, Michael Backes, Emiliano De Cristofaro, et al. Are your sensitive attributes private? novel model inversion attribute inference attacks on classification models. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4579–4596, 2022.
- [MDMZ⁺23] Orlando E Martínez-Durive, Sachit Mishra, Cezary Ziemlicki, Stefania Rubrichi, Zbigniew Smoreda, and Marco Fiore. The netmob23 dataset: A high-resolution multi-region service-level mobile data traffic cartography. *arXiv preprint arXiv:2305.06933*, 2023.
- [MGCdM23] Matthieu Meeus, Florent Guepin, Ana-Maria Crețu, and Yves-Alexandre de Montjoye. Achilles’ heels: vulnerable record identification in synthetic data publishing. In *European Symposium on Research in Computer Security*, pages 380–399. Springer, 2023.

- [ML21] Malcolm Morgan and Robin Lovelace. Travel flow aggregation: Nationally scalable methods for interactive and online visualisation of transport behaviour at the road network level. *Environment and Planning B: Urban Analytics and City Science*, 48(6):1684–1696, 2021.
- [MS20] Sasi Kumar Murakonda and Reza Shokri. Ml privacy meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning. *arXiv preprint arXiv:2007.09339*, 2020.
- [NA12] Kawee Numpacharoen and Amporn Atsawarungrangkit. Generating correlation matrices based on the boundaries of their coefficients. *PloS one*, 7(11):e48902, 2012.
- [NHS20] NHS England. A&e synthetic data. <https://data.england.nhs.uk/dataset/a-e-synthetic-data>, 2020. Accessed on 02/06/2023.
- [NRD16] Beata Nowok, Gillian M Raab, and Chris Dibben. synthpop: Bespoke creation of synthetic data in r. *Journal of statistical software*, 74:1–26, 2016.
- [NSH19] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 739–753. IEEE, 2019.
- [NST⁺21] Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlin. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on security and privacy (SP)*, pages 866–882. IEEE, 2021.
- [O219] O2. O2 transport smart steps product sheet. <https://static-www.o2.co.uk/sites/default/files/2019-04/o2-transport-smart-steps-product-sheet.pdf>, 2019. [Online].
- [Off11] Office for National Statistics. Census microdata teaching files. <https://www.ons.gov.uk/census>

[//www.ons.gov.uk/census/2011census/2011censusdata/censusmicrodata/microdatateachingfile](https://www.ons.gov.uk/census/2011census/2011censusdata/censusmicrodata/microdatateachingfile), 2011.

- [Off23] Office of the Privacy Commissioner of Canada. Investigation into the collection and use of de-identified mobility data in the course of the covid-19 pandemic. https://www.priv.gc.ca/en/opc-actions-and-decisions/investigations/investigations-into-federal-institutions/2022-23/pa_20230529_phac/, 2023. Accessed: 2023-09-14.
- [OJGdM19] Axel Oehmichen, Shubham Jain, Andrea Gadotti, and Yves-Alexandre de Montjoye. Opal: High performance platform for large-scale privacy-preserving location data analytics. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1332–1342. IEEE, 2019.
- [Oli21] Swikar Oli. Canada’s public health agency admits it tracked 33 million mobile devices during lockdown. *National Post*, 24, 2021.
- [OSRY18] Kun Ouyang, Reza Shokri, David S Rosenblum, and Wenzhuo Yang. A non-parametric generative model for human trajectories. In *IJCAI*, volume 18, pages 3812–3817, 2018.
- [PB96] José C Pinheiro and Douglas M Bates. Unconstrained parametrizations for variance-covariance matrices. *Statistics and computing*, 6:289–296, 1996.
- [PBBL11] Raluca Ada Popa, Andrew J Blumberg, Hari Balakrishnan, and Frank H Li. Privacy and accountability for location-based aggregate statistics. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 653–666, 2011.
- [PGM⁺19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

- [Pre24] Precisely. Placeiq movement. <https://www.precisely.com/product/precisely-placeiq/placeiq-movement>, 2024. Accessed: 2024-02-13.
- [PTDC17] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. Knock knock, who’s there? membership inference on aggregate location data. *arXiv preprint arXiv:1708.06145*, 2017.
- [PTDC20] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. Measuring membership privacy on aggregate location time-series. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(2):1–28, 2020.
- [PVG⁺11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.
- [QKS⁺19] Sehrish Qummar, Fiaz Gul Khan, Sajid Shah, Ahmad Khan, Shahaboddin Shamshirband, Zia Ur Rehman, Iftikhar Ahmed Khan, and Waqas Jadoon. A deep learning ensemble approach for diabetic retinopathy detection. *Ieee Access*, 7:150530–150539, 2019.
- [RB96] Kohavi Ronny and Becker Barry. UCI machine learning repository: Adult data set. <https://archive.ics.uci.edu/ml/datasets/Adult>, 1996.
- [RCKW05] Da Ruan, Guoqing Chen, Etienne E Kerre, and Geert Wets. *Intelligent data mining: techniques and applications*, volume 5. Springer Science & Business Media, 2005.
- [Red11] M. Redmond. UCI Repository of machine learning databases: Communities and Crime Unnormalized Data Set. <https://archive.ics.uci.edu/ml/datasets/communities+and+crime+unnormalized>, 2011.
- [Rek] Amazon Rekognition. Moderating content. <https://docs.aws.amazon.com/rekognition/latest/dg/moderation.html>.

- [Saf24] SafeGraph. Enrich pois with aggregated transaction data. <https://www.safegraph.com/products/spend>, 2024. Accessed: 2024-02-13.
- [Sav21] Charlie Savage. Intelligence analysts use u.s. smartphone location data without warrants, memo says. *The New York Times*, 01 2021. Available at <https://www.nytimes.com/2021/01/22/us/politics/dia-surveillance-data.html>.
- [SBB⁺20] Ahmed Mohamed Gamal Salem, Apratim Bhattacharyya, Michael Backes, Mario Fritz, and Yang Zhang. Updates-leak: Data set inference and reconstruction attacks in online learning. In *29th USENIX Security Symposium*, pages 1291–1308. USENIX, 2020.
- [SBC⁺13] Christian M Schneider, Vitaly Belik, Thomas Couronné, Zbigniew Smoreda, and Marta C González. Unravelling daily human mobility motifs. *Journal of The Royal Society Interface*, 10(84):20130246, 2013.
- [SdM15] SpazioDati and DEIB Politecnico di Milano. Social Pulse - Milano. <https://doi.org/10.7910/DVN/9IZALB>, 2015.
- [SE22] Anshuman Suri and David Evans. Formalizing and estimating distribution inference risks. *Proceedings on Privacy Enhancing Technologies*, 2022.
- [Skl59] M Sklar. Fonctions de répartition à n dimensions et leurs marges. In *Annales de l'ISUP*, volume 8, pages 229–231, 1959.
- [SMS⁺08] Mukund Seshadri, Sridhar Machiraju, Ashwin Sridharan, Jean Bolot, Christos Faloutsos, and Jure Leskove. Mobile call graphs: beyond power-law and lognormal distributions. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 596–604, 2008.
- [SOJH09] Sriram Sankararaman, Guillaume Obozinski, Michael I Jordan, and Eran Halperin. Genomic privacy and limits of individual detection in a pool. *Nature genetics*, 41(9):965–967, 2009.

- [SOT22] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. Synthetic data-anonymisation groundhog day. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1451–1468, 2022.
- [SQBB10] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.
- [SSSS17] 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. IEEE, 2017.
- [Syn24] Synthetic Data Expert Group, Financial Conduct Authority. Report: Using synthetic data in financial services. <https://www.fca.org.uk/publications/corporate-documents/report-using-synthetic-data-financial-services>, 2024.
- [SZH⁺18] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*, 2018.
- [TC23] Terje Trasberg and James Cheshire. Spatial and social disparities in the decline of activities during the covid-19 lockdown in greater london. *Urban Studies*, 60(8):1427–1447, 2023.
- [TdM22] Arnaud J Tournier and Yves-Alexandre de Montjoye. Expanding the attack surface: Robust profiling attacks threaten the privacy of sparse behavioral data. *Science Advances*, 8(33):eabl6464, 2022.
- [Tel24] Telus. Telus Insights Location API. <https://docs.insights.telus.com/>, 2024. [Online].
- [TLG⁺19] Stacey Truex, Ling Liu, Mehmet Emre Gursay, Lei Yu, and Wenqi Wei. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing*, 14(6):2073–2089, 2019.

- [TMH⁺21] Yuchao Tao, Ryan McKenna, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. Benchmarking differentially private synthetic data generation algorithms. *arXiv preprint arXiv:2112.09238*, 2021.
- [TWRM20] Allan Tucker, Zhenchen Wang, Ylenia Rotalinti, and Puja Myles. Generating high-fidelity synthetic patient data for assessing machine learning healthcare software. *NPJ digital medicine*, 3(1):1–13, 2020.
- [VBE18] Michael Veale, Reuben Binns, and Lilian Edwards. Algorithms that remember: model inversion attacks and data protection law. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 376(2133):20180083, 2018.
- [VSP⁺17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [VZ16] Liesbet Van Zoonen. Privacy concerns in smart cities. *Government Information Quarterly*, 33(3):472–480, 2016.
- [Wal19] Patrik Waldmann. On the use of the pearson correlation coefficient for model evaluation in genome-wide prediction. *Frontiers in genetics*, 10:899, 2019.
- [WSC⁺16] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.
- [Xia16] Qing Xiao. Calculating correlation coefficient for gaussian copula. *arXiv preprint arXiv:1608.00738*, 2016.
- [XSCIV19a] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems*, 32, 2019.

- [XSCIV19b] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in neural information processing systems*, 32, 2019.
- [XSZ⁺15] Yang Xu, Shih-Lung Shaw, Ziliang Zhao, Ling Yin, Zhixiang Fang, and Qingquan Li. Understanding aggregate human mobility patterns using passive mobile phone location data: a home-based approach. *Transportation*, 42:625–646, 2015.
- [YDD⁺19] Andrew Yale, Saloni Dash, Ritik Dutta, Isabelle Guyon, Adrien Pavao, and Kristin P Bennett. Assessing privacy and quality of synthetic health data. In *Proceedings of the Conference on Artificial Intelligence for Data Discovery and Reuse*, pages 1–4, 2019.
- [YGFJ18] 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. IEEE, 2018.
- [YJR⁺22] Takahiro Yabe, Nicholas KW Jones, P Suresh C Rao, Marta C Gonzalez, and Satish V Ukkusuri. Mobile phone location data for disasters: A review from natural hazards and epidemics. *Computers, Environment and Urban Systems*, 94:101777, 2022.
- [ZB11] Hui Zang and Jean Bolot. Anonymization of location data does not work: A large-scale measurement study. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, pages 145–156, 2011.
- [ZCP⁺17] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Trans. Database Syst.*, 42(4), oct 2017. <https://doi.org/10.1145/3134428>.
- [ZCSZ21] Junhao Zhou, Yufei Chen, Chao Shen, and Yang Zhang. Property inference attacks against gans. *arXiv preprint arXiv:2111.07608*, 2021.
- [ZDXF16] Haomiao Zhou, Zhihong Deng, Yuanqing Xia, and Mengyin Fu. A new sampling

- method in particle filter based on pearson correlation coefficient. *Neurocomputing*, 216:208–215, 2016.
- [ZFVH22] Keyu Zhu, Ferdinando Fioretto, and Pascal Van Hentenryck. Post-processing of differentially private data: A fairness perspective. *arXiv preprint arXiv:2201.09425*, 2022.
- [Zho17] Tao Zhou. Understanding location-based services users’ privacy concern: An elaboration likelihood model perspective. *Internet Research*, 27(3):506–519, 2017.
- [ZTO21] Wanrong Zhang, Shruti Tople, and Olga Ohrimenko. Leakage of dataset properties in multi-party machine learning. In *USENIX Security Symposium*, pages 2687–2704, 2021.
- [ZZZ20] Guanglin Zhang, Anqi Zhang, and Ping Zhao. Locmia: Membership inference attacks against aggregated location data. *IEEE Internet of Things Journal*, 7(12):11778–11788, 2020.

A Algorithms

A.1 Algorithm in Chapter 7

Algorithm to sample a valid correlation matrix. Algorithm A 1 details an implementation of the algorithm to sample a valid correlation matrix by Numpacharoen and Atsawarungruangkit [NA12]. We refer the reader to the original work for a slightly modified version which correctly handles the numerically unstable cases when the range of attainable values for $c_{i,j}$ is very small.

Algorithm A 1 SAMPLECORRMATRIX [NA12]. We highlight in red the statement which we modify when calling this algorithm inside Algorithm 8 and Algorithm 9. The modified statement for each case is provided in Sec. 4.1.1 and is omitted here for brevity. Code comments are written in blue.

```

1: Inputs:
    $n$ : Number of variables.
2: Output:
    $C \in \mathbb{R}^{n \times n}$ : A valid correlation matrix.
3: Initialize:
    $C \leftarrow 0$ ;  $B \leftarrow 0$ 
4: // Randomly initialize the first column.
5: for  $i \in \{2, \dots, n\}$  do
6:    $c_{i,1} \leftarrow \mathcal{U}(-1, 1)$  //  $\cos \theta_{i,1}$ 
7:    $b_{i,1} \leftarrow c_{i,1}$ 
8:    $b_{i,j} \leftarrow \sqrt{1 - c_{i,1}^2}$  for  $j \in \{2, \dots, i\}$   $\mathcal{U}(\sin \theta_{i,1})$ 
9: end for
10: for  $i \in \{2, \dots, n\}$  do
11:   for  $j \in \{1, \dots, i\}$  do
12:     // Compute the bounds  $m_{i,j} \pm l_{i,j}$ .
13:      $m_{i,j} \leftarrow B_{i,1:j-1}(B_{j,1:j-1})^T$ 
14:      $l_{i,j} \leftarrow b_{i,j}b_{j,j}$ 
15:     // Sample uniformly within bounds.
16:      $c_{i,j} \leftarrow \mathcal{U}(m_{i,j} - l_{i,j}, m_{i,j} + l_{i,j})$ 
17:     // Update  $B$ .
18:      $aux \leftarrow \frac{c_{i,j} - m_{i,j}}{l_{i,j}}$  //  $\cos \theta_{i,j}$ .
19:      $b_{i,j} \leftarrow b_{i,j} * aux$ 
20:      $b_{i,k} \leftarrow \sqrt{1 - aux^2}$  for  $k \in \{j+1, \dots, n\}$ 
21:   end for
22: end for
23:  $C \leftarrow C + C^T + I_n$ 

```

Algorithm to sample from a Gaussian Copula. Algorithm A 2 explain how to sample synthetic dataset from Gaussian Copulas.

Algorithm A 2 SAMPLEFROMGAUSSIANCOPULAS. Code comments are written in blue.

1: **Inputs:**

n : Number of variables.

$\Sigma \in \mathbb{S}_+^d$: A valid covariance matrix.

$F_1, \dots, F_n$: One-way marginals.

2: **Output:**

A sample $z = (z_1, \dots, z_n)$ from a Gaussian copula distribution parametrized by $F_1, \dots, F_n$ and Σ .

3: $A \leftarrow \text{CholeskyDecomposition}(C)$ // $C = A^t \cdot A$

4: $Z \sim \mathcal{N}(0, I_n)$

5: $X \leftarrow A^T Z$

6: **for** $i = 1, \dots, n$ **do**

7: $z_i \leftarrow F_i^{-1}(X_i)$

8: **end for**

9: $z \leftarrow (z_1, \dots, z_n)$

B Presentation of datasets

The goal of this section of the Appendix is to present the various datasets used in the main Chapter of this thesis. Datasets are separated by the Chapter they are used in. This appendix include an ethical consideration for Chapter 6, as we used real data provided by a third party.

B.1 Presentation of the datasets used in Chapter 3,4 and 5

UK Census [Off11] is the 2011 Census Microdata Teaching File published by the UK Office for National Statistics. It contains an anonymised fraction of the 2011 Census from England and Wales File (1%), consisting of 569741 records with 17 categorical columns.

Adult [RB96] contains an anonymized set of records from the 1994 US Census database. It consists of 48842 records of 15 columns. 9 of those columns are categorical, and 6 are continuous.

B.2 Presentation of the datasets used in Chapter 6

We here describe the two location datasets used for evaluating the MIAs in Chapter 6, and discuss ethical considerations of the data collection and usage.

CDR: The first dataset, which we refer to as "CDR", is a private dataset, shared with us by Flowminder [Flo24] for the purpose of this research. The raw dataset comprises timestamped and geo-tagged call records of approximately 11,000 mobile phone users within a Latin American metropolitan area. The observation period is June 2021, with epochs defined by the 720 hourly timeslots. The ROIs are defined by the service regions of approximately 500 cellular antenna towers within the metropolitan area, which spans $\sim 150\text{km}^2$. The users were selected such that they registered at least one visit per week, to omit users who changed SIM cards, and such that the majority of their visits are within the region, to ensure that they are residents. 50 target users for the MIAs were randomly selected by Flowminder. A histogram of the number of visits over the target traces is plotted in Figure 1(a).

Milan: The second dataset is the Milan Social Pulse dataset [SdM15], made publicly available as part of the Telecom Italia Big Data Challenge [BDNL⁺15]. This dataset comprises timestamped and geo-tagged tweets from 4840 mobile phone users within the Milano region. The ROIs are defined by a grid of 100 points, each with an approximate area of 256 km^2 . We consider the location data from the first week of data, yielding 168 hourly epochs. We do not delete any users from the dataset prior to aggregation. We randomly select 50 targets among users who tweeted at least 10 times during the observation period. A histogram of the number of visits over the target traces is plotted in Figure 1(b).

Ethical Considerations: Because of the sensitivity of location data, we did not access raw individual-level data, and instead collaborated with Flowminder (FM) to develop a privacy-preserving data-sharing pipeline for the purpose of this research [dMGB⁺18]. More specifically, data sharing was restricted to pre-computed aggregate matrices (labeled according to target membership, computed by FM on the data provider server) and 50 target traces randomly

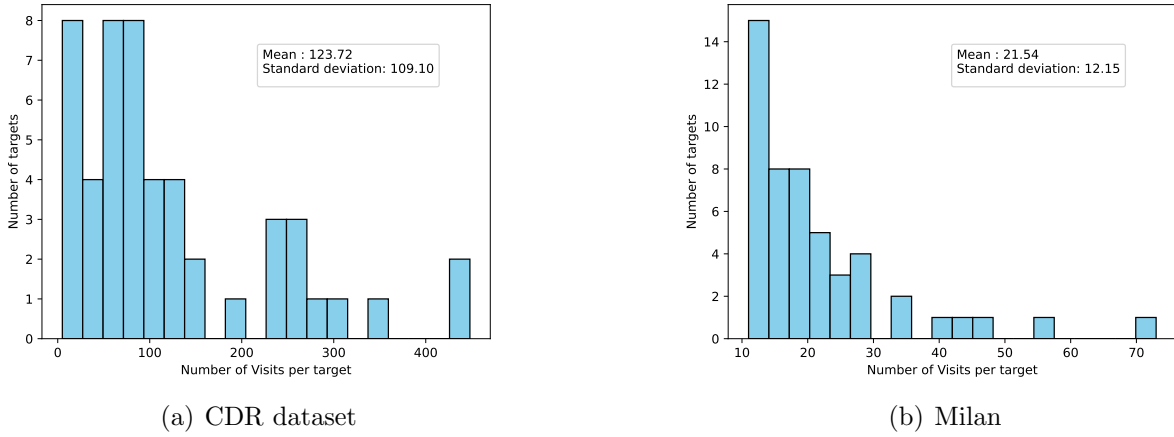


Figure B 1: **Number of visits for both CDR and Milan datasets.** The number of visits is computed per target over the 50 targets.

chosen by FM. To further mitigate the privacy risk, the ~ 500 ROIs and 720 epoch indices were randomly permuted in the shared aggregate and target trace matrices, according to a mapping known only by FM. This random permutation relabeled the space and time indices, enabling us to test the MIAs without knowing the true times or locations. The graphs of the marginal statistics (see Figures 6.5, 6.7, 6.9) were plotted by FM and shared with us. All the data shared by FM with us is subject to a research contract between FM and our institution and was kept on our segregated server. The Milan dataset, derived from geo-tagged tweets, remains publicly available, and was only used for the purpose of testing the MIAs.

B.3 Presentation of the datasets used in Chapter 7

We use three datasets in the real-world evaluation of Chapter 7.2.3.

Fifa19 [Gad19] contains 18207 records of 89 attributes each, describing physical attributes and performance statistics of football players. We discard the categorical attributes, keeping the continuous and discrete ordinal variables. We select the *Value* attribute, representing the amount of money a club paid for a player, as the output variable, discarding the other attributes predictive of it that would make classification trivial (e.g., *Wage*). We then remove the records with missing values, leaving us with 15917 records. We binarize the output variable

by mapping it to $Y = 1$ if larger than the median value ($> 0.74M$) and to $Y = 0$ otherwise. Finally, we remove the duplicate attributes, which leaves us with 53 input attributes.

Communities and Crime [Red11] contains 2215 records of 147 attributes each, describing socio-economic, law enforcement, and crime statistics on communities in the US. We select the number of murders as the output variable of our machine learning models, discarding the other columns predictive of it (i.e., relating explicitly to crime). As input attributes, we select the 101 continuous and discrete ordinal attributes from the dataset that do not have missing values, discarding the rest. We binarize the number of murders by mapping it to $Y = 1$ if is at least 1, and to $Y = 0$ otherwise.

Musk (v2) [Die98] contains 6598 records of 166 attributes each, describing different conformations of 165 molecules, together with a human expert-assigned label of “musk” or “non-musk”. We train the models to classify conformations as “musk” ($Y = 1$) or “non-musk” ($Y = 0$). The dataset being heavily imbalanced with respect to the two classes, we balance it for simplicity, leaving us with 2034 records.

C Parameters and notations used throughout experiments

C.1 Parameters used in experiments of Chapter 3

Notations used in Chapter 3 can be found in Table C 1

Table C 1: **Parameters used throughout experiments of Chapter 3.**

	$ D_{aux} $	$ D_{test} $	n_{shadow}	n_{test}
Adult	10000	5000	4000	200
UK Census	50000	25000	4000	200

C.2 Parameters used in experiments of Chapter 4

Notations used in Chapter 4 can be found in Table C 2

Table C 2: **Parameters used throughout experiments of Chapter 4.** N_{shadow} represents the number of shadow models used to train the MIA, and N_{eval} represents the number of evaluation models used.

Dataset	$\mathcal{M}_{target}$	Library	$ D_{aux} $	$ D_{eval} $	$ D_{target} $	N_{shadow}	N_{eval}
Adult [RB96]	Synthpop [Haz19]	Reprosyn [Ala22]	30000	15222	1000	1000	1000
	BayNet [ZCP+17]						
	NN	Sk-learn [PVG+11]			2000	500	500
UK Census [Off11]	Synthpop [Haz19]	Reprosyn [Ala22]	52390	27193	1000	1000	1000
	BayNet [ZCP+17]						
CIFAR10 [KH+09]	ResNet [HZRS16]	PyTorch [PGM+19] FFCV [LIE+23]	30000	30000	10000	256	256

C.3 Parameters used in experiments of Chapter 5

Notations used in Chapter 5 can be found in Table C 3

Table C 3: **Parameters used throughout experiments of Chapter 5.**

Dataset	Scenario	$ D_{aux} $	$ D_{test} $	m	n_{shadow}	n_{test}
Adult	S0	10000	5000	1000	2000	100
	S1	0	5000	20000		
	S2	0	5000	1000		
	S3	0	5000	20000		
UK Census	S0	50000	25000	1000	2000	100
	S1	0	25000	20000		
	S2	0	25000	1000		
	S3	0	25000	20000		

C.4 Notations used in Chapter 6

Notations used in Chapter 6 can be found in Table C 4

Table C 4: **Glossary of notations used in Chapter 6.**

Notation	Definition
$\mathcal{S}$	Set of regions of interests (ROIs)
$\mathcal{T}$	Set of epochs in observation period
Ω	Set of all users in the dataset
L^u	Location trace of user $u \in \Omega$ over $\mathcal{S} \times \mathcal{T}$
$\mathcal{U}$	Aggregation group of users sampled from Ω
$A^{\mathcal{U}}$	Raw aggregate count matrix in $\mathcal{S} \times T$ over users in $\mathcal{U}$
$A_{DP}^{\mathcal{U}}(\varepsilon)$	An ε -DP aggregate
$A_{SSC}^{\mathcal{U}}(k)$	An aggregate with counts $\leq k$ suppressed
$A_{DP,SSC}^{\mathcal{U}}(\varepsilon, k)$	An ε -DP aggregate with counts $\leq k$ suppressed
$\overline{A}^{\mathcal{U}}$	The released aggregate count matrix
m	Number of users in the aggregation group $\mathcal{U}$
u^*	Target drawn from full population, $u^* \in \Omega$
Adv	Adversary performing MIA on u^*

C.5 Parameters used in experiments of Chapter 7

Given $\mathcal{M}_T$, we train a meta-classifier on outputs extracted from $k = |D_{\text{meta}}|$ models. We generate $K = 5000$ synthetic datasets for target models trained on $n = 3, 4$ and 5 variables and $K = 10000$ synthetic datasets for $n = 6$ to 10 variables. In the experiments on synthetic data, the models are all trained on 1000 samples and the target model test accuracy is computed on 500 unseen samples. In the experiments on real-world datasets, the target models are trained on all the samples available ($|D_T| = 2215$ for Communities and crime and $|D_T| = 2034$ for Musk), except for Fifa19, where we sample $|D_T| = 2000$ records uniformly without replacement to keep the dataset sizes similar. The synthetic datasets are always generated to have a number of samples equal to the target dataset size $|D_T|$. We run the MODIFYCORRELATIONCONSTRAINTS heuristic (Algorithm 11 of Chapter 7) using parameter values $S = 100$ and $N_D = |D_T|$, for a maximum of $M = 10$ iterations and with an error tolerance of $e = 0.01$. We set the size of the query dataset D_{query} equal to $Q = 100$, and discuss in Section 7.2.3 of Chapter 7 the impact of this parameter. We use a Logistic Regression as the meta-classifier for LR models, as it leads to similar accuracy as the MLP but takes less time to train. We use an MLP as the meta-clasifier

for MLP models, trained the same as stated the MLPs described above for the target models, except that we use a batch size of 128, a learning rate of $\eta = 0.001$, $\mathbb{L}_2$ weight decay of 0.01, and early stopping after 10 epochs of non-improving accuracy on 10% of held-out samples.

D Additional results

D.1 Additional results for Chapter 7

Impact of constraint values on the results. For completeness, we here include results on the impact of the average constraint $(|\rho(X_1, Y)|, \rho(X_2, Y)|)/2$ (Figure D 1) and of the smallest constraint $\min(|\rho(X_1, Y)|, \rho(X_2, Y)|)$ (Figure D 2).

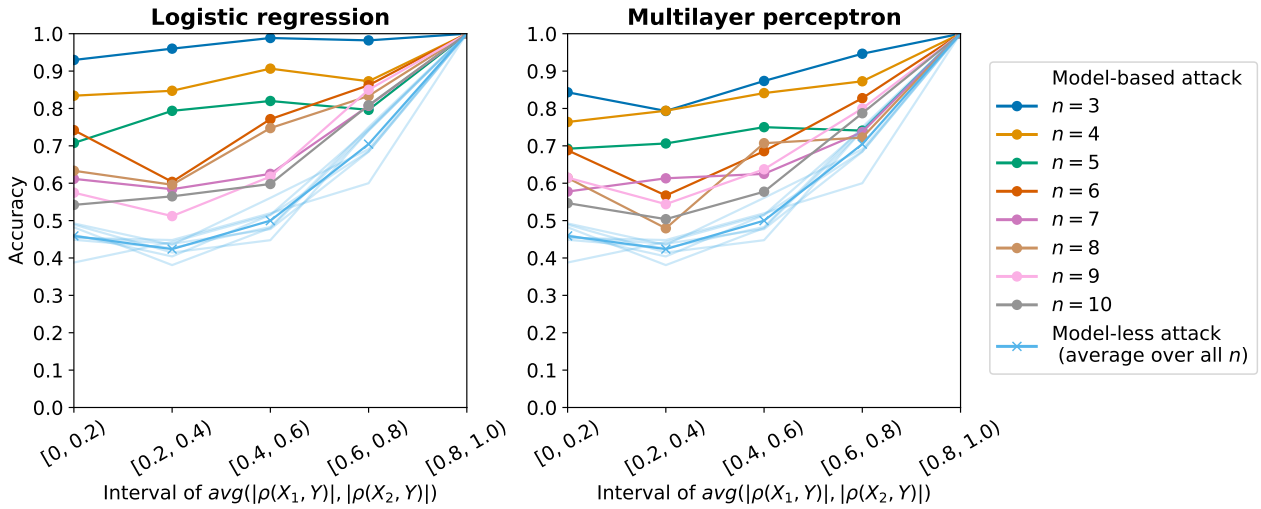


Figure D 1: **Impact of the average constraint $(|\rho(X_1, Y)| + |\rho(X_2, Y)|)/2$ on the attack accuracy for different number of variables n .** We compare the results of our model-based and model-less attacks on logistic regression (left) and MLP models (right).

Lowering the precision of confidences scores for the MLP. Additional mitigation experiments computed using an MLP as target model.

Impact of the granularity of the marginals. Figure D 4 shows the impact of the number of sub-intervals G over which the one-way marginals released to the adversary are computed on the accuracy of our attack.

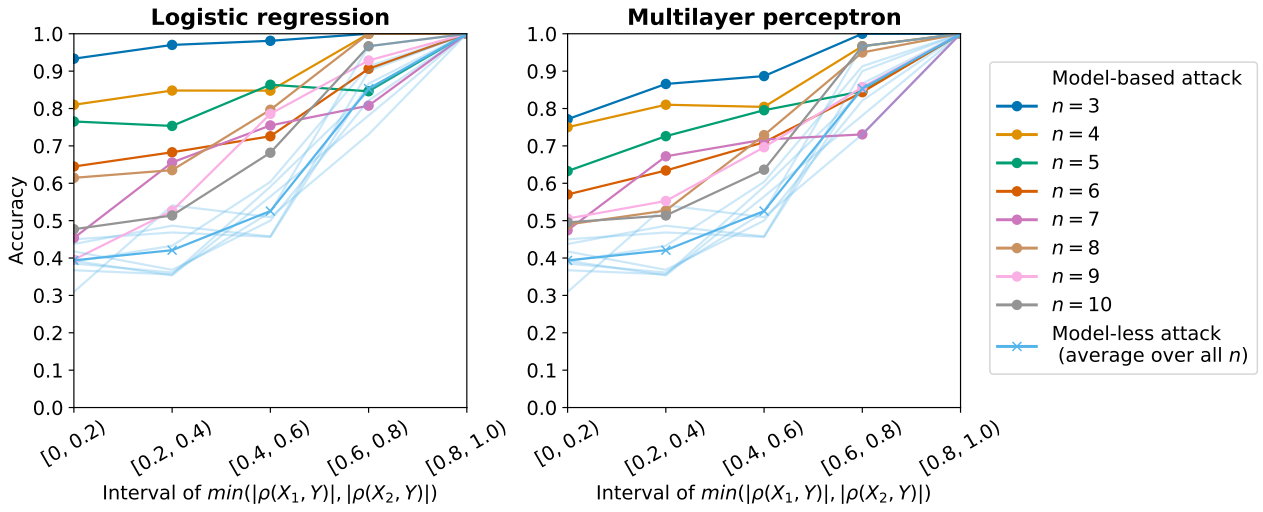


Figure D 2: **Impact of the smallest constraint $\min(|\rho(X_1, Y)|, |\rho(X_2, Y)|)$ on the attack accuracy for different number of variables n .** We compare the results of our model-based and model-less attacks on logistic regression (left) and MLP models (right).

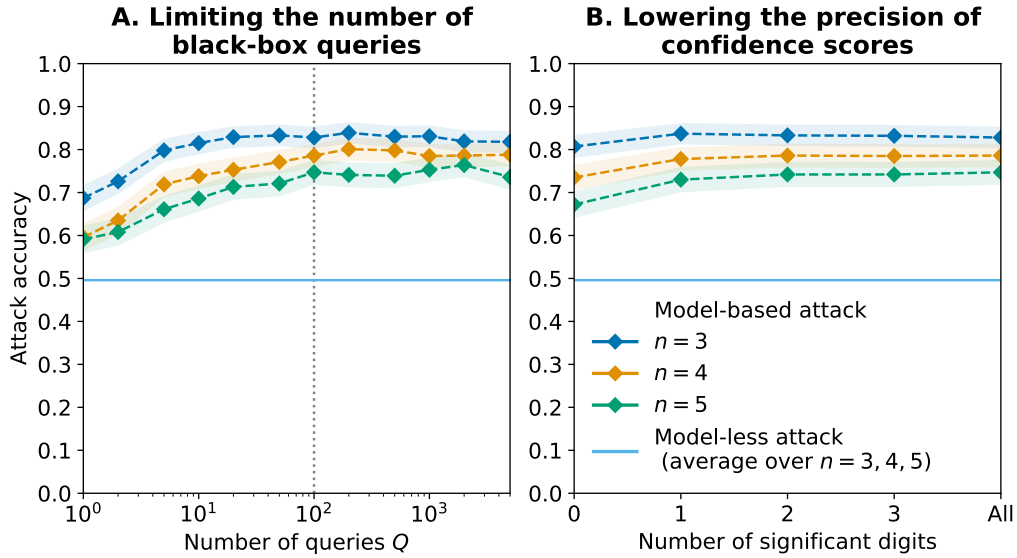


Figure D 3: **Impact of mitigations on the accuracy of our attack against MLP models.** We report results for two mitigations: limiting the number of black-box queries (A) and lowering the precision of confidence scores (B). We report the accuracy (with 95% confidence interval) of our model-less and model-based attacks over 1000 targets models for $n \in \{3, 4, 5\}$ variables.

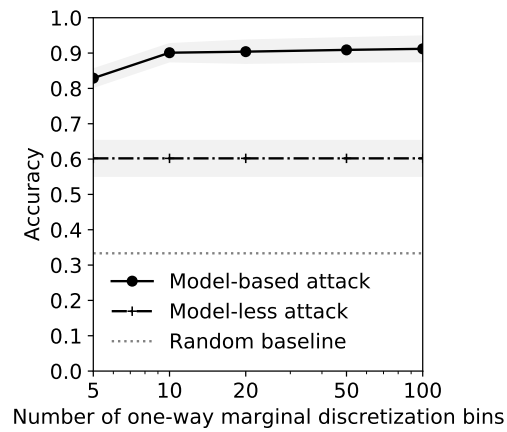


Figure D 4: **Fifa19: Impact of the granularity of one-way marginals available to the adversary.** We report the attack accuracy (mean and standard deviation) on logistic regression models as we vary the number of discretization bins applied to the one-way marginals.