Data Leakage in the Context of Machine Unlearning

Ji Gao **Sanjam Garg** **Mohammad Mahmoody** **Prashant Nalini Vasudevan**

Abstract

Privacy attacks on machine learning models aim to extract, or just identify, the data that is used to train such models. In light of recent legal requirements, many machine learning methods are being upgraded to support *unlearning* as well. In this work, we study the privacy implication of such deletion updates. We consider attacks that leverage having access both to the original model and to the model after unlearning. In this setting, we show simple and intuitive attacks that are extremely effective at violating privacy.

1 Introduction

Machine learning has traditionally focused on deriving predictive models from a collection of data examples/records $\mathcal{S} = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$. Towards this goal, learning algorithms are designed to minimize the risk/error of predicting the correct label y of a new instance $\mathbf{x}$ for a newly sampled record $\mathbf{e} = (\mathbf{x}, y)$. However, a trained model h obtained via such methods could potentially reveal sensitive information about the examples that were used to train them. For example, a model h might reveal the members of its training set, potentially violating the privacy of the individuals who contributed those records. Such exposure is of major concern in certain (e.g., medical/political) contexts. Furthermore, the ever increasing role of machine learning in decision making and the public availability of learning models as a service [38], heightens the importance of such privacy concerns. Recent legal requirements (e.g., the European Union's GDPR [29] or California's CCPA [12]) aim to make such privacy considerations mandatory, but the question of *how* such privacy concerns can be modeled and enforced is the subject of ongoing study [47, 10, 37, 23].

The work of Shokri et al. [43] demonstrated that natural and even commercialized ML models do, in fact, leak a lot about their training sets. In particular, they demonstrated a powerful framework for attacking the privacy of ML models through *membership inference*. In such attacks, the adversary with input example $\mathbf{e}$ and access to ML model h wants to deduce if the example $\mathbf{e}$ was in fact present in the data set $\mathcal{S}$ that was used to train h or not. This work and many follow-up works [34, 40, 35, 49, 9, 33, 50, 39, 32] can be seen as demonstrating ways to infer information about data sets (or even reconstruct them) based on publicly available statistics about them [13, 17, 2, 18, 41, 28], and are also tightly related to works on what an ML model memorizes about its training set [44, 48, 7, 21].

On the defense side, differential privacy [13, 15, 14] provides a framework to provably limit the information that would leak about the training records used in a training process. This is done by guaranteeing that an individual's participation in the data set (versus not doing so) will have little statistical impact on the distribution of the produced ML model. Thus, any form of interaction with the trained model h (or even a full white-box disclosure of it) will essentially not reveal whether a particular record $\mathbf{e}$ was a member of the data set or not. While it is a very powerful privacy guarantee, differential privacy imposes a challenge on the learning process [45, 19, 3, 11, 42, 46] that usually leads to major utility loss when using the same amount of training data [4].

Privacy challenges in the presence of unlearning. The aforementioned attacks deal with settings in which a trained model gets deployed and accessed, and so, the ML model is a *static* object rather than a dynamic one. However, this assumption is not always realistic. Indeed, in light of the recent attention given to the *right to erasure* or the *right to be forgotten* (as also stressed by legal

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

requirements such as GDPR and CCPA) a new line of work has emerged with the goal of *unlearning* or simply *deleting* records from a machine learning model [6, 24, 25, 23, 5, 30, 26, 36]. Namely, upon a deletion request for a record $\mathbf{e} \in \mathcal{S}$, one needs to update h to h_{del} such that h_{del} is (ideally) the same as training a model from scratch using $\mathcal{S} \setminus \{\mathbf{e}\}$. Now, if an ML model gets updated due to a deletion/unlearning request, we are no longer dealing with a static object as the ML model.

Consider the process of perfectly deleting record $\mathbf{e}$ from the data set $\mathcal{S}$ that was used to construct a model h as described above: obtain h_{del} by re-training using the smaller data set $\mathcal{S} \setminus \{\mathbf{e}\}$. Intuitively, it seems like this should resolve privacy concerns regarding the record $\mathbf{e}$, at least if the job of the adversary is to extract some information about $\mathbf{e}$ from the ML model. After all, we are eliminating $\mathbf{e}$ from the learning process. However, there is a catch! The adversary now can access *both* models h and h_{del}, and so it can potentially decode additional information about the deleted record $\mathbf{e}$. As a simplified demonstrating example, suppose the records $\mathbf{e}_1, \ldots, \mathbf{e}_n = \mathcal{S}$ are real-valued vectors, and suppose and the ML model, upon a query, returns their summation. Then, if the set $\mathcal{S}$ is large with sufficient entropy, it might hide, to some extend, its elements. But, upon deleting one of the records $\mathbf{e}_i$, and updating the model that returns the new sum $\sum_{j \neq i} \mathbf{e}_j$, one can find out $\mathbf{e_i}$ exactly. In other words, the very task of deletion might *harm* the privacy concerns around the deleted record $\mathbf{e}$.

Our contribution. To understand the privacy implications of machine unlearning, we revisit privacy attacks and study their power and limitations in the new setting where access to both h and h_{del} is provided to the adversary.[1] In particular, we study three types of attacks as follows. In each case, we propose new attacks that leverage access to the ML models before and after deletion and show through experiments that our attacks achieve very high success rates. In each case, we also explore and explain the theoretical intuition enabling our attacks.

- **Deletion inference.** Can the adversary distinguish between a data record $\mathbf{e}$ that was deleted from an ML model and one that was not?[2]
  We show that extremely simple attacks can be designed to distinguish deleted records from other records by relying on the intuition that the model is more fit to the training data than to other data. This attack builds on the implicit intuition of many previous membership inference attacks. In fact, one can even reduce the task of deletion inference to *four* sub-tasks of membership inference of the same records $\mathbf{e}$ and $\mathbf{e}'$ (the two records to be distinguished) with respect to the models before and after the deletion. However, our attacks show that one can achieve *very high* precision beyond what we can achieve by two queries to previous membership inference attacks. We present simple attacks both for regression and classification against a diverse range of ML models.
- **Deleted data approximation.** Can the adversary *reconstruct* the deleted record $\mathbf{e}$ at least approximately under a meaningful approximation metric?
  We show that having black-box access to models h and h_{del} can sometimes allow the adversary to get a very good approximation of the record $\mathbf{e}$. The idea is to find local differences in the loss space of the two ML models and then track such differences to find the (approximate) point that is the cause. We show how to implement this idea for the case of nearest-neighbor models.
- **Deleted label approximation.** For a deleted record $(\mathbf{x}, y) = \mathbf{e}$, can an adversary given $\mathbf{x}$ learn *more information* about the label y, than each of the models h, h_{del} alone provide?
  We show that doing so is possible for linear regression. In particular, we show an attack using which one can extrapolate a deleted point's label to a precision that is *more* than what is provided through the original model h or the model after deletion h_{del}.

Conclusion. Our attacks demonstrate that the unlearning operation could come at an extra cost in privacy loss. One remedy to prevent such leakage is to use very strong forms of differential privacy [16, 31, 8] that handle any form of continual observation. However as mentioned above, even basic forms of differential privacy come with a computational cost in training and the amount of data, and hence it remains an important direction to directly study the implications of deletion operations on data privacy for efficient algorithms as well. Many intriguing questions remain. In particular, it would be interesting to study attacks that leverage *multiple rounds* of deletions, as well as finding *efficient* learning methods that allow deletion with provable privacy guarantees.

[1] Yet, one constraint is that h can only be queried before h_{del} becomes available.

[2] One can show that distinguishing attacks are equivalent to inference attacks (that is, inferring whether $\mathbf{e}$ was deleted), however we find our attacks to be simpler to explain and analyze in the distinguishing form.

2 Our Attacks and Experiments

In this section we describe our three types of attacks on machine unlearning. In each case, we will first explain our experiment's setting, then explain the theoretical intuition behind the attack's design, and finally will report our experimental results. Due to space limitations, we describe the details of the data sets that we use and how we synthesize data in the supplemental material.

2.1 Deletion Inference Attack on Regression

Attack's setting and the success criteria. In this attack, the adversary is given two labeled examples $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2)$ (with real valued labels y_1, y_2) where one of them is the deleted sample $\mathbf{e} = (\mathbf{x_e}, y_\mathbf{e})$, the adversary can pick the deleted sample out with high success rate. We used synthesized data sets (details in the appendix) and multiple regression models including linear regression, Lasso regression, SVM Regressor and Decision Tree Regressor[3] in the experiment. We then randomly draw one sample $(\mathbf{x_e}, y_\mathbf{e})$ from the training dataset to delete, and draw an additional sample that is either inside $\mathcal{S}$ (for both models) or outside $\mathcal{S}$ (for both models). We repeat this experiment for 1000 runs. The success criteria of the experiment is the success rate of the attack.

Our attack and the intuition behind it. We propose two attacks, $\mathsf{DelInfLbl}$ which uses both data $\mathbf{x}$ and label y, and DelInf which only uses $\mathbf{x}$. $\mathsf{DelInfLbl}$ compares the change of loss function used in training (MSE for example), namely, $\ell(h'(\mathbf{x}_1), y_1) - \ell(h(\mathbf{x}_1), y_1)$ and $\ell(h'(\mathbf{x}_2), y_2) - \ell(h(\mathbf{x}_2), y_2)$. The attack marks the record with *larger* positive change on the loss to be the deleted sample. Intuitively, the deleted sample's loss will increase after deletion, while another training sample's loss will decrease by average (assume Learn follows the ERM principle). DelInf directly compares the distances of outputs between two models, namely, $|h(\mathbf{x}_1) - h'(\mathbf{x}_1)|$ and $|h(\mathbf{x}_2) - h'(\mathbf{x}_2)|$. The adversary marks the record with *larger* distance as the deleted sample. Intuitively, the deleted sample's distance will be larger in comparison to a sample which either remains in the data set $\mathcal{S}$ or remains out of the data set $\mathcal{S}$.

	DelInf		DelInfLbl	
Learner method	Inside $\mathcal{S}$	Outside $\mathcal{S}$	Inside $\mathcal{S}$	Outside $\mathcal{S}$
Linear Regression	99.30%	98.70%	99.60%	99.40%
Lasso Regression	93.90%	92.80%	99.80%	99.90%
Decision Tree	100.00%	82.40%	100.00%	92.20%
Support Vector Machine	89.70%	89.40%	91.20%	91.30%

Table 1: Summary of success rate of the attacks DelInf and $\mathsf{DelInfLbl}$ on Regression Learners

2.2 Deletion Inference Attack on Classification Models

Attack's setting and the success criteria. Similarly to the regression setting, in this attack the adversary is given two examples and wants to infer which one is the deleted one, but the difference is that we are dealing with discrete labels (e.g., in $\{0, 1\}$). We use synthesized datasets (details in the appendix) and multiple classification models including logistic regression, SVM Classifier and Decision Tree Classifier. We then randomly draw one sample $(\mathbf{x_e}, y_\mathbf{e})$ from the training dataset to delete, and draw an additional sample. Similarly, we consider two scenarios, the additional sample being inside $\mathcal{S}$ and outside $\mathcal{S}$. The success criteria of the experiment is the success rate of both attacks DelInf and $\mathsf{DelInfLbl}$.

	DelInf		DelInfLbl	
Learning method	Inside $\mathcal{S}$	Outside $\mathcal{S}$	Inside $\mathcal{S}$	Outside $\mathcal{S}$
Logistic Regression	99.60%	99.50%	99.90%	99.60%
Random Forest	100.00%	99.50%	100.00%	99.90%
Support Vector Machine	89.50%	89.60%	91.00%	92.90%

Table 2: Summary of success rate of the attacks DelInf and $\mathsf{DelInfLbl}$ on Classification Learners

Our attack and the intuition behind it. We apply the same attacks DelInf and $\mathsf{DelInfLbl}$ in a similar style to regression models. Comparing to regression where the labels are numbers, in classification we use the predicted posterior probability over the labels as the output. Similar to regression, intuitively

[3]Implementation of the methods are from the python library Scikit-learn.

the deleted sample $\mathbf{e}$'s posterior will likely to change more after the deletion. The loss will also be larger for the deleted sample $\mathbf{e}$. For the choice of loss function in $\mathsf{DelInfLbl}$, we use Hinge Loss.

2.3 Deleted Label Approximation Attack on Linear Regression

Attack's setting and the success criteria. In this experiment, the adversary is given a features vector of the deleted record $\mathbf{x_e}$ and wishes to approximate the true label of the deleted sample $y_\mathbf{e}$ by querying the models before and after the deletion. The goal is to beat the correctness of both models. We perform the attack on the linear regression model. We test the attack on two traditional regression datasets, the Boston Housing Price Dataset [27] and the diabetes dataset [20]. The details of the datasets can be found in the appendix. For each dataset, we train the model h with the whole dataset. We then randomly pick a sample $\mathbf{e}$ and perform the re-training on the data set without $\mathbf{e}$. The adversary returns an approximation $\tilde{y}_\mathbf{e}$ and the success criteria is the distance between $\tilde{y}_\mathbf{e}$ and $y_\mathbf{e}$.

Our attack and the intuition behind it. We propose an attack that we call $\mathsf{LabelApp}$: it utilizes $\hat{y}_\mathbf{e} = h(\mathbf{x_e})$ and $\hat{y}'_\mathbf{e} = h'(\mathbf{x_e})$. The attacker then returns $\tilde{y}_\mathbf{e} = \hat{y}_\mathbf{e} + \lambda \cdot (\hat{y}_\mathbf{e} - \hat{y}'_\mathbf{e})$ as a close approximation to $y_\mathbf{e}$ where λ is a carefully chosen constant parameter of the attack. Intuitively, we have $\hat{y}'_\mathbf{e} \geq \hat{y}_\mathbf{e}$. Therefore, moving further from $\hat{y}'_\mathbf{e}$ towards $\hat{y}_\mathbf{e}$ for a positive λ is going to have less loss, which is closer to the actual $y_\mathbf{e}$. The best value of λ in each different scenario could be empirically estimated by a similar size data set that is individually sampled by the attacker.

Experiments' results. We calculate the average distance of $\tilde{y}_i$ and y_i with different λ. The results are shown in Table 3. Our results show that there exists a λ value for each data set that can greatly increase the approximation by reducing the the estimated loss by around 70%, which leads to a much smaller error than both $\hat{y}$ and $\hat{y}'$. In case the two models were supposed to hide the label (perhaps if it was a sensitive information to know very precisely) the data removal process, in this case, clearly goes against the goal of hiding y in its exact form.

	Best λ	$\mathbb{E}[(y_i - \hat{y}_i)^2]$	$\mathbb{E}[(y_i - \hat{y}'_i)^2]$	$\mathbb{E}[(y_i - \tilde{y}_i)^2]$	Gain(%)
Boston Housing	17.5	21.897	23.728	7.149	14.75(70%)
Diabetes	30	2859.7	3001.7	829.8	2029.8(72%)

Table 3: Result of the Data Label Extraction Attack on LR

2.4 Deleted Data Approximation Attack on K-NN model

Attack's setting and the success criteria. In this experiment, the goal of the adversary is to approximate the whole vector of the deleted sample $\mathbf{x_e}$ as a point in high dimension. We perform our experimental attack on the K-Nearest-Neighbors (K-NN), also one of the most basic machine learning approaches. K-NN model predicts the label of a sample by taking average of the labels of K nearest neighbors of that sample. We test the attack on two traditional classification datasets, the Iris Dataset [22] and the Wine Recognition dataset [1]. For each dataset, we train the model h following the whole dataset with $K = 3$. We then randomly pick a sample e and perform the re-training on the data set without $\mathbf{e}$. The adversary returns an $\tilde{\mathbf{x}}_\mathbf{e}$ with queries to both models and the success criteria is the distance between $\tilde{\mathbf{x}}_\mathbf{e}$ and $\mathbf{x_e}$.

Our attack and the intuition behind it. We define an attack $\mathsf{DataApp}$ in this scenario that first randomly draws samples from the data distribution, and query the two models in the corresponding order. The adversary then returns the average of all samples whose output label is different. Intuitively, for a well generalized model, the impact of one sample's deletion to the model is mostly local rather than global. In this case, the average of these samples that have different outputs gives a much closer estimation of $\mathbf{x_e}$ comparing to a random approximation.

In the experiment, we run $\mathsf{DataApp}$ with 10000 random samples draw uniformly from the data range. We denote the attack to be failed when no sample has its label changed in this phase, otherwise we compare the distance of predicted $\tilde{\mathbf{x}}_\mathbf{e}$ to the average of samples whose output label changed.

	Failed rate	Estimated point to e	Avg Sample Distance
Iris	34%	0.32	0.64
Wine	6.7%	0.75	0.99

Table 4: Result of Data Feature Extraction Attack in K-NN model

References

[1] Stefan Aeberhard, Danny Coomans, and Olivier De Vel. Comparative analysis of statistical pattern recognition methods in high dimensional settings. *Pattern Recognition*, 27(8):1065–1077, 1994.

[2] Michael Backes, Pascal Berrang, Mathias Humbert, and Praveen Manoharan. Membership privacy in microrna-based studies. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 319–330, 2016.

[3] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014.

[4] Amos Beimel, Hai Brenner, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. *Machine learning*, 94(3):401–437, 2014.

[5] Lucas Bourtoule, Varun Chandrasekaran, Christopher Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. *arXiv preprint arXiv:1912.03817*, 2019.

[6] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pages 463–480. IEEE, 2015.

[7] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 267–284, 2019.

[8] TH Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. In *International Colloquium on Automata, Languages, and Programming*, pages 405–417. Springer, 2010.

[9] Christopher A Choquette Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. *arXiv preprint arXiv:2007.14321*, 2020.

[10] Aloni Cohen and Kobbi Nissim. Towards formalizing the gdpr's notion of singling out. *CoRR*, abs/1904.06009, 2019.

[11] Ashish Dandekar, Debabrota Basu, and Stéphane Bressan. Differential privacy for regularised linear regression. In *International Conference on Database and Expert Systems Applications*, pages 483–491. Springer, 2018.

[12] Lydia de la Torre. A guide to the California consumer privacy act of 2018. *Available at SSRN 3275571*, 2018.

[13] 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.

[14] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

[15] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284, New York, NY, USA, March 4–7, 2006. Springer, Heidelberg, Germany.

[16] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 715–724, 2010.

[17] Cynthia Dwork, Adam Smith, Thomas Steinke, and Jonathan Ullman. Exposed! a survey of attacks on private data. *Annual Review of Statistics and Its Application*, 4:61–84, 2017.

[18] Cynthia Dwork, Adam Smith, Thomas Steinke, Jonathan Ullman, and Salil Vadhan. Robust traceability from trace amounts. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 650–669. IEEE, 2015.

[19] Cynthia Dwork, Kunal Talwar, Abhradeep Thakurta, and Li Zhang. Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 11–20, 2014.

[20] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[21] 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.

[22] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[23] Sanjam Garg, Shafi Goldwasser, and Prashant Nalini Vasudevan. Formalizing data deletion in the context of the right to be forgotten. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part II*, volume 12106 of *Lecture Notes in Computer Science*, pages 373–402. Springer, 2020.

[24] Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. In *Advances in Neural Information Processing Systems*, pages 3513–3526, 2019.

[25] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep neural networks. *arXiv preprint arXiv:1911.04933*, 2019.

[26] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030*, 2019.

[27] DAVID HARRISON. Hedonic housing prices and the demand for clean air. *JOURNAL OF ENVIRONMENTAL ECONOMICS AND MANAGEMENT*, 5:81–102, 1978.

[28] 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 Genet*, 4(8):e1000167, 2008.

[29] Chris Jay Hoofnagle, Bart van der Sloot, and Frederik Zuiderveen Borgesius. The european union general data protection regulation: what it is and what it means. *Information & Communications Technology Law*, 28(1):65–98, 2019.

[30] Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models: Algorithms and evaluations. *arXiv preprint arXiv:2002.10077*, 2020.

[31] Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. In *Conference on Learning Theory*, pages 24–1, 2012.

[32] Bargav Jayaraman, Lingxiao Wang, David Evans, and Quanquan Gu. Revisiting membership inference under realistic assumptions. *arXiv preprint arXiv:2005.10881*, 2020.

[33] Zheng Li and Yang Zhang. Label-leaks: Membership inference attack with label. *arXiv preprint arXiv:2007.15528*, 2020.

[34] Yunhui Long, Vincent Bindschaedler, and Carl A Gunter. Towards measuring membership privacy. *arXiv preprint arXiv:1712.09136*, 2017.

[35] Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. Understanding membership inferences on well-generalized learning models. *arXiv preprint arXiv:1802.04889*, 2018.

[36] Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. *arXiv preprint arXiv:2007.02923*, 2020.

[37] Kobbi Nissim, Aaron Bembenek, Alexandra Wood, Mark Bun, Marco Gaboardi, Urs Gasser, David R O'Brien, Thomas Steinke, and Salil Vadhan. Bridging the gap between computer science and legal approaches to privacy. *Harv. JL & Tech.*, 31:687, 2017.

[38] Mauro Ribeiro, Katarina Grolinger, and Miriam AM Capretz. Mlaas: Machine learning as a service. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 896–902. IEEE, 2015.

[39] Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. Updates-leak: Data set inference and reconstruction attacks in online learning. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 1291–1308, 2020.

[40] Ahmed Salem, Yang Zhang, Mathias Humbert, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Network and Distributed Systems Security Symposium 2019*. Internet Society, 2019.

[41] 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.

[42] Or Sheffet. Old techniques in differentially private linear regression. In *Algorithmic Learning Theory*, pages 789–827, 2019.

[43] 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.

[44] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 587–601, 2017.

[45] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248. IEEE, 2013.

[46] Kunal Talwar, Abhradeep Guha Thakurta, and Li Zhang. Nearly optimal private lasso. In *Advances in Neural Information Processing Systems*, pages 3025–3033, 2015.

[47] Michael Veale, Reuben Binns, and Lilian Edwards. Algorithms that remember: Model inversion attacks and data protection law. *CoRR*, abs/1807.04644, 2018.

[48] 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.

[49] Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, and Zhenkai Liang. Neural network inversion in adversarial setting via background knowledge alignment. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 225–240, 2019.

[50] Santiago Zanella-BÃ©guelin, Lukas Wutschitz, Shruti Tople, Victor Ruehle, Andrew Paverd, Olga Ohrimenko, Boris KÃ¶pf, and Marc Brockschmidt. Analyzing information leakage of updates to natural language models. In *ACM Conference on Computer and Communication Security (CCS)*. ACM, ACM, November 2020.

A Supplemental Material: Details of Data Used

A.1 Synthesized Datasets

For the deletion inference attack on regression ,we assume the input $\mathbf{x}_i$ is drawn from a 10 dimensional Gaussian distribution $N(\mathbf{0}, \mathbf{I})$ where $\mathbf{0} = (0, \ldots, 0), \mathbf{1} = (1, \ldots, 1)$, and $\mathbf{I}$ is the identity matrix, and output $y_i = \langle \mathbf{w}, \mathbf{x}_i \rangle + \varepsilon_i$ follows a linear function with fixed $\mathbf{w}$ and an independent additive Gaussian noise from $N(\mathbf{0}, 0.1 \cdot \mathbf{I})$ represented by ε. We draw 1000 random samples from the data distribution to create a training dataset.

For the deletion inference attack on classification, we assume the input $\mathbf{x}_i$ is drawn from a mixture Gaussian distribution that includes two independent 10 dimensional Gaussian distribution $N(\mathbf{0}, \mathbf{I})$ and $N(0.1 \cdot \mathbf{1}, \mathbf{I})$ where $\mathbf{0} = (0, \ldots, 0), \mathbf{1} = (1, \ldots, 1)$, and $\mathbf{I}$ is the identity matrix. Example's label is determined by its distribution, that is, $y = 0$ for the 1st Gaussian distribution and $y = 1$ for the 2nd Gaussian distribution. In this experiment we draw 500 random samples for each Gaussian distribution to create a training dataset.

A.2 Real Datasets

Table 5 and 6 are the details of the real datasets we used in the experiments.

	No. of Samples	No. of Features	Predict
Boston Housing	506	14	The median house price
Diabetes	442	10	Predict Disease progression

Table 5: Regression Dataset Descriptions

	No. of Samples	No. of Features	No. of Labels	Predict
Iris [22]	150	4	3	The type of Iris plants
Wine [1]	178	13	3	Wine cultivator

Table 6: Classification Dataset Descriptions