Making Machine Learning Forget*

Saurabh Shintre[1], Kevin A. Roundy[1], and Jasjeet Dhaliwal[2]

[1] Symantec Research Labs, Symantec Corporation
350 Ellis Street, Mountain View, CA, USA 94043
[2] Center for Advanced Machine Learning, Symantec Corporation
350 Ellis Street, Mountain View, CA, USA 94043
{Saurabh_Shintre, Kevin_Roundy, Jasjeet_Dhaliwal}@symantec.com

Abstract. Machine learning models often overfit to the training data and do not learn general patterns like humans do. This allows an attacker to learn private membership or attributes about the training data, simply by having access to the machine learning model. We argue that this vulnerability of current machine learning models makes them indirect stores of the personal data used for training and therefore, corresponding data protection regulations must apply to machine learning models as well. In this position paper, we specifically analyze how the "right-to-be-forgotten" provided by the European Union General Data Protection Regulation can be implemented on current machine learning models and which techniques can be used to build future models that can forget. This document also serves as a call-to-action for researchers and policy-makers to identify other technologies that can be used for this purpose.

Keywords: Machine Learning · GDPR · Right-to-be-forgotten · Privacy-by-Design.

1 Introduction

The rise of the data economy has led to the creation of a number of internet services that collect personal data of consumers and offer useful services in return. The data collected by these services is shared with other processors for further analysis or for targeted advertising. Due to this complex network of data controllers and processors, consumers often lack control of the different ways in which their personal data is stored and shared. To make matters worse, the privacy policies of the some of these services are presented to consumers in complex legal parlance that prevents them from making decisions that protect their privacy [15]. Collected data is also stored in data-centers for long periods of time which helps these services build invasive personal profiles of their users, including sensitive information like location, commercial activity, medical and personal history [24]. Large-scale collection and storage of personal information leads to major security and privacy risks for consumers. The data can be hacked or leaked with malicious intent which leads to the consumer losing all control

* Supported by Symantec Corporation

over their personal information [1]. At the same time, such information allows service providers to infer other private information that can cause personal or financial loss to the consumer [6].

To protect consumers from such risks, a number of jurisdictions have implemented regulations that control the collection, storage, and sharing of personal information. The General Data Protection Regulation (GDPR) [11] of the European Union is a comprehensive legislation that covers steps that data controllers and processors must undertake to ensure security and privacy of personal data of subjects within the EU. GDPR extends the notion of personal information from identity information, such as name and addresses, to any information that can be personally identifiable like GPS locations, IP addresses, etc. It also mandates that data controllers and processors can only collect information that is relevant to their services and require explicit user consent to do so. In addition to mandates on transparency, storage, and security, Article 17 of the GDPR also gives a consumer the right to have their personal information removed from a service provider. The "right to erasure", often referred to as the "right-to-be-forgotten", mandates that data controllers must provide a mechanism through which data subjects can request the deletion and stop further processing of all their personal information collected by the data controller [11].

While it is relatively straightforward to keep track of raw stores of private data, the implementation of "right-to-be-forgotten" is made very complex due to the use of personal information in training a variety of machine learning models [16]. Such models are used to provide insights about credit worthiness, bio-metric authentication, medical diagnosis etc [18]. Due to the popularity of machine learning as a service (MLaaS), data controllers often give data to processors that train machine learning models for the controller and delete the raw data once the training is over [21]. This allows data controllers to satisfy legislative mandates because machine learning models are not considered stores of private information under most legislation. However, it has recently been shown that machine learning models often overfit to the training data [25]; i.e., they display higher accuracy on training data than on previously unseen test data. Hence, it is possible for an attacker with access to the model to identify data used to train the model and learn private attributes [13] [19] [21]. Fig. 1 shows the success of model inversion attacks on a facial recognition model by only using the model and the name of the subject.

In this position paper, we opine that the existence of such attacks indirectly makes machine learning models stores of personal information. Therefore, all mandates of the GDPR that apply to regular stores of personal information must be extended to machine learning models trained with such data. In this paper, we specifically look at how the "right-to-be-forgotten" can be implemented on machine learning models and introduce techniques like *influence functions* [14] and *differential privacy* [10] as potential approaches to solve this problem.

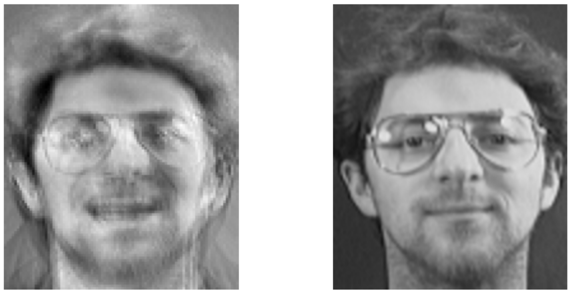

Fig. 1: Recovered training image using attribute inference attacks v/s original training image [13]

2 The “right-to-be-forgotten”

The GDPR framework created by the European Union provides EU residents with protection against predatory practices of data-based internet services. GDPR enforces rules about the kind of user data that can be collected, shared, stored in a persistent manner, and how it should be safe-guarded. While other regulatory frameworks, such as HIPAA and COPPA regulations in the United States, also control the collection and storage of personal information, the “**right-to-be-forgotten**” is certainly unique to the GDPR [11]. Specified in the Article 17 of the GDPR framework, the right-to-be-forgotten, also known as the right-to-erasure, states that “*the data subject shall have the right to obtain from the controller the erasure of personal data concerning him or her without undue delay and the controller shall have the obligation to erase personal data without undue delay where one of the following grounds applies:*

- *the personal data are no longer necessary in relation to the purposes for which they were collected or otherwise processed;*
- *the data subject withdraws consent on which the processing is based according to point (a) of Article 6(1), or point (a) of Article 9(2), and where there is no other legal ground for the processing;*
- *the data subject objects to the processing pursuant to Article 21(1) and there are no overriding legitimate grounds for the processing, or the data subject objects to the processing pursuant to Article 21(2);*
- *the personal data have been unlawfully processed;*
- *the personal data have to be erased for compliance with a legal obligation in Union or Member State law to which the controller is subject;*
- *the personal data have been collected in relation to the offer of information society services referred to in Article 8(1).”*

The right-to-be-forgotten also requires the data controller to take any technical steps necessary to prevent the processing of information by data processors with whom the data controller has shared this data.

3 Privacy leakage in machine learning systems

The use of collected information to train machine learning models, specifically in a Machine Learning as a Service model, makes implementation of the right-to-be-forgotten extremely complicated. Users' data is used by controllers and processors to build machine learning systems for a variety for services, ranging from facial recognition [18] to medical diagnostics as in IBM Watson. However, most of the popularly-implemented machine learning algorithms often memorize the data used to train them [25]. Therefore, even if raw copies of the training data are deleted, data can be recreated from the machine learning model [13].

Leakage of private information in machine learning models can be done via two types of attacks. In **attribute inference** attacks, an attacker can recreate sensitive features about a user by having access to the machine learning model and partial publicly-available features, such as names, gender, etc. In order to do so, the attacker simply needs access to the confidence values outputted by the model. For example, given a facial recognition model, the attacker can recreate the face of a person of his/her choice by simply identifying images that are classified as that person with high confidence [13], as can be seen in Fig. 1. Substantial evidence points to the phenomenon of *over-fitting* as a lead cause of such attacks [25]. Over-fitting occurs when a machine learning model memorizes the training data rather than learning general features about it [22]. Such a model performs extremely well on data points close to training data points while performing poorly at other data points. Thus, by identifying regions of the input space where the model predictions are confident, the attacker can recreate the training points in that region [13].

In **membership inference** attacks, the attacker wishes to learn if a certain data point was used to train a model [21]. This attack can be successful even if the attacker only has a black-box API access to the model [19]. Membership of the user's data in a specific dataset can reveal sensitive information about that user. For example, the presence of user's data in control vs experimental groups of a medical trial can reveal the user's medical condition. To implement such attacks, the attacker builds multiple shadow models for which he knows the training data. The shadow models are trained to mimic the performance of the target model and have an additional binary output deciding whether a data point is "in" or " out" of the training set. At test time, all the different shadow models are engaged and if the majority of them classify the test point as "in", then the data point is part of the original training data. The reasons behind the success of these attacks aren't fully understood due to the lack of explainability in machine learning algorithms like deep learning. Yeom *et al.* identified high influence of specific training data points on the model parameters as one of the root cause of this weakness [25] .

4 Implementing "right-to-be-forgotten" in machine learning models

Membership and attribute inference attacks described in the previous section demonstrate that machine learning models act like indirect stores of the personal information used to train them. Therefore, the right-to-be-forgotten is incomplete if it does not apply to the machine learning models trained with personal information. Apart from the reasons of privacy, the ability of machine learning models to forget certain training data points also help improve their security and usability, because the model can unlearn the effects of poisoned or erroneously created training data [7].

A straightforward way to implement the right-to-be-forgotten in machine learning models is to delete the requesting user's personal information from the training set and retrain the model entirely. This method is impractical because commercial machine learning models may have millions of parameters and are trained over large corpora of data. Retraining them requires significant cost and effort which a data processor may not be able to afford without charging a fee for entertaining such requests. Additionally, the possibility of retraining the model to comply with right-to-be-forgotten may compel data processors to persistently store personal information in its raw format, when they would not do so otherwise, which can make it susceptible to theft or leakage. Therefore, we must design solutions that allow to models to forget training data without requiring retraining. We identify three existing techniques that can potentially be used for this purpose.

4.1 Influence Functions

Influence functions are tools from robust statistics that measure the effect of a training point on the machine learning model's parameters and predictions. Specifically, they measure the change in model's accuracy at a test input when a training point is removed from the training set. Koh and Liang [14] formalized this concept for deep neural networks and provided a closed-form expression to measure the influence of a training point on the model's parameters and performance at a test input. The measurement of influence of a training point on a test input is done in two parts. First, we measure the change in the model parameters caused by the removal of a training point and then we measure the change in model loss at the test point given the change in the model parameters.

Consider a model $\mathcal{F}$, trained on the training data $\boldsymbol{X}_{tr}$, $\boldsymbol{Y}_{tr}$ where X represents the features and Y represents the labels. Let $\mathcal{L}$ represent the loss function used to train the model. That is, the function $\mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, \mathbf{y})$ measures how far the prediction made by the model under parameters $\boldsymbol{\theta}$ at an input $\mathbf{x}$ is from the corresponding true label, $\mathbf{y}$. For algorithms like deep learning, mean squared error or categorical cross entropy are routinely chosen as the loss function.

Given initial model parameters, model risk is measured as the average model loss over the training data,

$$\mathcal{R}(\boldsymbol{\theta}) = \frac{1}{|\boldsymbol{X}_{tr}|} \sum_{\substack{\mathbf{x} \in \boldsymbol{X}_{tr} \\ \mathbf{y} \in \boldsymbol{Y}_{tr}}} \mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, \mathbf{y})$$

The goal of model training is to find parameters θ^* that minimize the model risk. Therefore,

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\ \mathcal{R}(\boldsymbol{\theta})$$

Assuming that $\mathcal{R}(\boldsymbol{\theta})$ is convex and differentiable, we have,

$$\nabla_{\boldsymbol{\theta}} \mathcal{R}(\theta^*) = 0$$

Increasing the weight of a specific training point $\mathbf{x}^*$ by a small amount $\epsilon \in \mathcal{R}$ leads to a new risk function

$$\mathcal{R}_{\mathbf{x}^*,\epsilon}(\theta) = \mathcal{R}(\theta) + \epsilon \mathcal{L}(\boldsymbol{\theta}, \mathbf{x}^*, \mathbf{y}^*)$$

Note: setting $\epsilon = -\frac{1}{|\boldsymbol{X}_{tr}|}$ is equivalent to leaving the training point $\mathbf{x}^*$ out of the training data completely. Minimizing the new model risk leads to a different set of optimal parameters

$$\boldsymbol{\theta}^*_{\mathbf{x}^*,\epsilon} = \arg \min_{\boldsymbol{\theta}}\ \mathcal{R}_{\mathbf{x}^*,\epsilon}(\theta)$$

Koh and Liang were able to measure the change in optimal model parameters due to up-weighting $\mathbf{x}^*$ by an infinitesimally small ϵ as

$$\frac{\partial}{\partial \epsilon} \boldsymbol{\theta}^*_{x,\epsilon} = -H^{-1}_{\boldsymbol{\theta}^*} \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^*, \mathbf{x}^*, \mathbf{y}^*)$$

where $H_{\boldsymbol{\theta}*} = \nabla^2_{\boldsymbol{\theta}} \mathcal{R}(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}$ represents the Hessian matrix of the model risk with respect to the model parameters [14]. Koh and Liang defined the influence of a training point, $\mathbf{x}^*$, on the loss at a test input, $\mathbf{x}'$ as

$$\begin{aligned} \mathcal{I}(\mathbf{x}^*, \mathbf{x}') &\overset{\text{def}}{=} \frac{\partial}{\partial \epsilon} \mathcal{L}(\boldsymbol{\theta}^*, \mathbf{x}', \mathbf{y}') \Big|_{\epsilon=0} \\ &= -\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^*, \mathbf{x}', \mathbf{y}')^T \cdot H^{-1}_{\boldsymbol{\theta}^*} \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^*, \mathbf{x}^*, \mathbf{y}^*) \end{aligned}$$

Thus, the quantity $Q_1 = \frac{1}{|\mathbf{X}_{tr}|} H^{-1}_{\boldsymbol{\theta}^*} \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^*, \mathbf{x}^*, \mathbf{y}^*)$ measures the change in the optimal model parameters and the quantity $Q_2 = -\frac{1}{|\boldsymbol{X}_{tr}|} \mathcal{I}(\mathbf{x}^*, \mathbf{x}')$ measures the change in model loss at a test point $\mathbf{x}'$, when the training point $\mathbf{x}^*$ has been left out from training. Koh and Liang experimentally verified that their approach is equivalent to leaving one data point out and retraining the model [14].

Our proposal: With this formulation, we propose to use influence functions to implement right-to-be-forgotten in existing models. When a user requests for his/her data to be removed, the data processor must identify all the machine learning models where the user's personal data was used for training. Having complete access to the model parameters, the processor can compute the new parameters when the user's data is removed from the training set. These new parameters can be easily computed by measuring the influence of the user's data and adding the amounts specified by Q_1 to the parameters. Influence functions also allow a neutral auditor to audit and confirm that the request to erase data was completed. To do so, the auditor must maintain the store of the current parameters used in the model. When a right-to-be-forgotten request is made, the requesting user can provide his or her data securely to the auditor and the auditor can measure the change in model parameters before and after the erasure request. If the change measures out to be the same as that specified by influence functions (Q_1), then the auditor can verify that the request was correctly processed. Even if the data processor cannot give the model parameters to the auditor, say to protect intellectual property, the auditor can maintain a standard set of test inputs and measure the change in the model's loss on these inputs. If the change is equivalent to the amount specified by influence function (Q_2), the auditor can verify that the request was properly met.

One of the advantages of using influence functions is that their use to implement right-to-be-forgotten does not require major changes to existing models or to training methods. Therefore, the use of influence functions does not adversely affect the model's performance or add substantial operating cost for the data processor. However, this approach is not a complete solution. If the model is stolen or leaked, the attacker might be able to re-create all the sensitive data. To protect user's personal information against this possibility, it is important to train models that are resilient to membership and attribute inference attacks. Differential privacy can be used for training such models [10].

4.2 Differential privacy

Differential privacy is a framework proposed by Dwork *et al.* [10], that captures precisely how much additional information of an individual is leaked by participating in a dataset, that would not have been leaked otherwise. Responsible dataset curators can use differential privacy practices to measure the leakage of information pertaining to individuals when disclosing aggregate statistics about the data and when replying to dataset queries in general. In the context of machine learning, the differential privacy framework allows one to measure how much additional information a machine learning model leaks about an individual.

Formally, a randomized learning algorithm $\mathcal{A}$ is said to be (ϵ, δ) *differentially-private* if, for two datasets $\mathbf{X}$, $\mathbf{X}'$ differing in only data point, and a machine learning model $\mathcal{M}$,

$$Pr[\mathcal{A}(\mathbf{X}') = \mathcal{M}] \leq e^{\epsilon} Pr[\mathcal{A}(\mathbf{X}) = \mathcal{M}] + \delta$$

That is, the probability that the learning algorithm $\mathcal{A}$ returns a model $\mathcal{M}$ is approximately the same, whether it is trained on $\mathbf{X}$ or $\mathbf{X}'$. The lower the values of the parameters ϵ and δ are, the higher the privacy provided by the randomized learning algorithm.

Hence, differential privacy provides guarantees about how much the addition or removal of a data point from the training dataset will affect the trained machine learning model. Consequently, a learning algorithm that provides differential privacy guarantees with ϵ and δ equal to zero leaks no information about whether a single individual was part of the training dataset or not. Further, learning algorithms that provide such guarantees are immune to inference attacks by definition. Achieving this property in practice however is not trivial and the goal then becomes that of finding the lowest possible (ϵ, δ) while still maintaining utility. Despite this compromise in utility, algorithms that achieve good differential privacy guarantees are increasingly used in practice because the differential privacy metric provides one of the strongest theoretical guarantees of privacy [3].

One of the earliest works combining differential privacy and machine learning was done by Agrawal and Ramakrishnan [4] in which the authors developed a novel algorithm to learn a decision tree classifier on differentially-private data. That is, they considered the problem of building a decision tree classifier on a dataset that was differentially-private. In order to do so, they first developed a reconstruction algorithm that estimated the distribution of the original dataset and then used this estimated distribution in conjunction with the perturbed data in order to build a decision tree classifier. Chaudhari *et al.* [8] extended research in this direction by generalizing the approach for training differentially-private machine learning models. They did so by developing a differentially-private framework for empirical risk minimization in which they perturbed the objective function to provide privacy guarantees. Since then, other works have focused on releasing differentially-private models including logistic regression, 2^{nd} moment matrix approximation, rule mining and more [12], [20],[26].

In a recent example, Abadi *et al.* [2] developed a method for providing differential privacy guarantees for deep learning models by adding Gaussian noise to the gradient values during model training. The amount of noise they add is carefully crafted to achieve differential privacy guarantees while still maintaining model efficacy. There has also been progress in situations when part of the dataset is sensitive and the other part is public. Papernot *et al.* [17] developed a framework in which first a fixed number of teacher models are trained on disjoint subsets of the sensitive data. An ensemble of these teacher models is then used to label the public data in a differentially-private manner while keeping number of labelling queries fixed in order to limit privacy cost. The public data along with differentially-private labels is then used to train a student model which provides differential privacy guarantees with respect to the sensitive data.

We note that the application of differential privacy that we have described thus far still requires individuals to place significant trust in the dataset curator. Practical implementations of solid differentially-private algorithms have been found to contain mistakes that result in significantly weaker privacy guarantees in practice than in theory [9] [23]. In addition, users still have no protection against the dataset itself being breached or leaked by a malicious insider. Many of these concerns can be alleviated by the application of differential privacy mechanisms directly on individual data at the point of data collection. This practice is known as Local Differential Privacy, which systematically adds noise to the data as it is being collected. The amount of added noise depends on the desired privacy guarantees. As the collected data itself is noisy, even a breach at the data collector does not expose users' raw data. Due to such strict privacy control, local differential privacy tends to severely limit the utility provided by the dataset, and truly massive collections of data may be required to perform even simple analysis, such as frequent itemset mining [5]. In practice, local differential privacy algorithms also destroy the usefulness of the dataset for inferences other than the pre-specified ones which makes it a very attractive technique from a consumer privacy standpoint. For these reasons, it seems important for regulators to encourage the use of local differential privacy techniques when appropriate.

4.3 Machine unlearning

Cao and Yang developed an approach of making machine learning unlearn a given data point [7]. In their approach, the machine learning model is not directly trained on the training data but on a small number of aggregates (summations) computed on the training data. Each summation is the sum of efficiently computable transformations on the training data. Once these transformations are computed, the training data is erased and only the transforms are used to train the model. To erase the effect of a specific training point, its contribution is subtracted from summed transformations. For certain machine learning algorithms like naive Bayes classifiers or support vector machines, the entire influence of training point can be removed in $\mathcal{O}(1)$ complexity. However, this approach is limited to such algorithms only and not to more advanced methods like deep learning.

To show how machine unlearning can be implemented in practice, we will use the example of the naive Bayes classifier [7]. Given a data point with features F_1, F_2, $\cdots$, F_k, the label L selected by the classifier is the one which has the maximum probability of being observed given the feature $F_1, F_2, \cdots F_k$. The posterior probability of being observed is computed using the Bayes rules as

$$P(L|F_1, F_2, \cdots F_K) = \frac{P(L)\prod_j P(F_j|L)}{\prod_j P(F_j)}$$

Each component, such as $P(F_j|L)$ is computed from the training data by computing the number of training points that have feature F_j and the label L, i.e. $\#(F_J \text{ AND } L)$ and dividing it with the number of training points that have the label L, i.e. $\#(L)$. That is, $P(F_j|L) = \frac{\#(F_J \text{ AND } L)}{\#(L)}$. From the point of view of the classifier only these aggregates are important and once they are computed, individual data points can be discarded. To unlearn a data point, we only need the feature F and the label L of the data point and update these counts. Say we need to remove a data point that has both the feature F_j and the label L, we need to update $P(F_j|L) = \frac{\#(F_J \text{ AND } L)-1}{\#(L)-1}$. Other sophisticated algorithms like Support Vector Machines and k-mean clustering can also be represented in this form [7].

Despite its efficacy and efficiency, machine unlearning suffers from two main drawbacks. One, it still requires that the data point to be removed must be submitted in its raw format to an auditor to fulfill the removal request. This is so because the model creator may have removed all the raw data and might only be storing the summations and the features/label of the data point have to be re-submitted by the user to update the summations. Machine unlearning shares this drawback with the use of influence functions proposed in Section 4.1. Two, machine unlearning provides no way to tell whether a specific data point is currently being used to train the current machine learning model or not. This is in contrast to the use of influence functions where low influence of a data point may imply that either it is not part of the training set or it is not an important piece of data for training. This makes the job of an auditor difficult as model designer can claim that the user's data is not being used from training.

5 Discussion and Conclusions

In this position paper, we identify machine learning models as indirect stores of personal information. We described membership and attribute inference attacks that can be used to recover the personal information hidden in these models. Due to this fact, we suggest that the right to erasure enshrined in GDPR Article 17 must extend not only to raw storage of personal information but also to machine learning models trained with such information. We describe three methods i) influence functions ii) differential privacy, and iii) machine unlearning that either allow erasure of specific data points from trained models or train models from which original data cannot be recovered. Such methods can allow services that build models on personal information to maintain users' privacy with minimal cost and service disruptions.

Each method has its own benefits and limitations. Influence functions have an advantage that they can work on existing models without requiring any fundamental change to model training and therefore, do not impact the utility of the model. However, removal of data using influence functions requires that the raw data be submitted to the auditor. Differential privacy provides the strongest guarantee of privacy among all the listed methods but it requires the development of new training methods altogether and may suffers from loss in the model's utility (some works [17] claim that differential privacy acts as a regularization technique and may actually improve model performance). Machine unlearning requires some logistical changes in training. Also, it does not work with all machine learning models, requires raw data sample to be submitted for removal, and provide no way to inferring if a data point is already being used to train the algorithm. Thus, each approach may be suitable in some context while not in others. The goal of achieving privacy in machine learning models also appears to be at odds with other desirable properties, such as explainability and transparency. Therefore, it is important to invest in lines of research that develop models that maintain privacy while providing transparency and explainability.

References

1. Equifax identifies additional 2.4 million customers hit by data breach (2018), https://www.nbcnews.com/business/business-news/equifax-identifies-additional-2-4-million-customers-hit-data-breach-n852226
2. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 308–318. CCS '16, ACM, New York, NY, USA (2016)
3. Abowd, J.M.: The u.s. census bureau adopts differential privacy. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &#38; Data Mining. pp. 2867–2867. KDD '18 (2018)
4. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: ACM Sigmod Record. vol. 29, pp. 439–450. ACM (2000)
5. Bittau, A., Erlingsson, Ú., Maniatis, P., Mironov, I., Raghunathan, A., Lie, D., Rudominer, M., Kode, U., Tinnés, J., Seefeld, B.: Prochlo: Strong privacy for analytics in the crowd. In: Proceedings of the 26th Symposium on Operating Systems Principles. pp. 441–459. (SOSP), ACM (2017)
6. Cadwalladr, C., Graham-Harrison, E.: Revealed: 50 million facebook profiles harvested for cambridge analytica in major data breach (2018), https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election
7. Cao, Y., Yang, J.: Towards making systems forget with machine unlearning. In: 2015 IEEE Symposium on Security and Privacy. pp. 463–480 (May 2015). https://doi.org/10.1109/SP.2015.35
8. Chaudhuri, K., Monteleoni, C., Sarwate, A.D.: Differentially private empirical risk minimization. Journal of Machine Learning Research **12**(Mar), 1069–1109 (2011)
9. Ding, Z., Wang, Y., Wang, G., Zhang, D., Kifer, D.: Detecting violations of differential privacy. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. pp. 475–489. CCS '18, ACM, New York, NY, USA (2018)
10. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) Theory of Cryptography. pp. 265–284. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
11. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). Official Journal of the European Union **L119**, 1–88 (May 2016), http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:L:2016:119:TOC
12. Evfimievski, A., Srikant, R., Agrawal, R., Gehrke, J.: Privacy preserving mining of association rules. Information Systems **29**(4), 343–364 (2004)
13. Fredrikson, M., Jha, S., Ristenpart, T.: Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security. pp. 1322–1333. CCS '15, ACM, New

York, NY, USA (2015). https://doi.org/10.1145/2810103.2813677, http://doi.acm.org/10.1145/2810103.2813677

14. Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. In: Internation Conference on Machine Learning (2017)
15. McDonald, A.M., Cranor, L.F.: The cost of reading privacy policies **4** (2008)
16. Michael Veale, R.B., Edwards, L.: Algorithms that remember: model inversion attacks and data protection law (2018)
17. Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., Erlingsson, Ú.: Scalable private learning with pate. CoRR **abs/1802.08908** (2018)
18. Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: British Machine Vision Conference (2015)
19. Salem, A., Zhang, Y., Humbert, M., Berrang, P., Fritz, M., Backes, M.: Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In: 26th Annual Network and Distributed System Security Symposium (NDSS 2019) (February 2019), https://publications.cispa.saarland/2754/
20. Sheffet, O.: Private approximations of the 2nd-moment matrix using existing techniques in linear regression. arXiv preprint arXiv:1507.00056 (2015)
21. Shokri, R., Stronati, M., Shmatikov, V.: Membership inference attacks against machine learning models. 2017 IEEE Symposium on Security and Privacy (SP) pp. 3–18 (2017)
22. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**, 1929–1958 (2014)
23. Tang, J., Korolova, A., Bai, X., Wang, X., Wang, X.: Privacy loss in apple's implementation of differential privacy on macos 10.12. CoRR (2017), http://arxiv.org/abs/1709.02753
24. Valentino-DeVries, J., Singer, N., Keller, M.H., Krolik, A.: Your apps know where you were last night, and theyre not keeping it secret (2018), https://www.nytimes.com/interactive/2018/12/10/business/location-data-privacy-apps.html?module=inline
25. Yeom, S., Giacomelli, I., Fredrikson, M., Jha, S.: Privacy risk in machine learning: Analyzing the connection to overfitting. 2018 IEEE 31st Computer Security Foundations Symposium (CSF) pp. 268–282 (2018)
26. Zhu, T., Li, G., Zhou, W., Philip, S.Y.: Differentially private deep learning. In: Differential Privacy and Applications, pp. 67–82. Springer (2017)