A Critical Overview of Privacy in Machine Learning

Emiliano De Cristofaro, UCL & Alan Turing Institute
e.decristofaro@ucl.ac.uk

Abstract

This article reviews privacy challenges in machine learning, providing an overview of the relevant research literature. We discuss possible adversarial models and settings, cover a wide range of attacks related to private and/or sensitive information leakage, and highlight several open problems in this space.

1 Prologue

Providers like Google, Microsoft, and Amazon provide customers with access to software interfaces to easily embed machine learning (ML) tasks into their applications. Overall, organizations can use Machine Learning as a Service (MLaaS) engines to outsource complex tasks, e.g., training classifiers, performing predictions, etc. They can also let others query models trained on their data. Naturally, this approach can also be used and is often advocated in other contexts, including government collaborations, citizen science projects, and business-to-business partnerships. Alas, if malicious users could recover data used to train these models, the resulting information leakage would create serious issues. Likewise, if the model's parameters are secret, or considered proprietary information, then access to the model should not allow an adversary to learn such parameters. In this article, we set to review privacy challenges in this space, providing a systematic review of the relevant research literature.

We discuss possible adversarial models and settings, cover a wide range of attacks related to private and/or sensitive information leakage, and briefly point to recent results attempting to defend against such attacks. Finally, we conclude with a list of open problems that require more work, including the need for better evaluations, targeted defenses, and the study of the relation to policy and data protection efforts.

NB: This article is not meant to present a comprehensive survey of literature in the field, nor an exhaustive list of all threat models and attacks of privacy in machine learning; interested readers may refer to existing surveys, e.g., [1].

2 ML Background

ML approaches. ML models can also be categorized depending on the probability distributions they learn. In supervised learning, assuming one has some input data x (e.g., pictures of animals) and wants to classify it into labels y (e.g., type of animal), then, roughly speaking, one can use either:

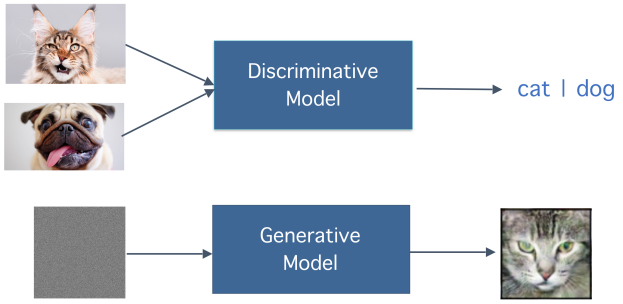

Figure 1: A simple illustration of how one can use discriminative vs generative models. The former learns to distinguish between two classes, i.e., pictures of cats or dogs. The latter estimates the underlying distribution of a dataset (pictures of cats) and randomly generate realistic, yet synthetic, samples according to their estimated distribution.

- *Discriminative models* to learn the *conditional* probability distribution $p(y|x)$, and ultimately learn to distinguish between different classes (e.g., cats vs dogs).
- *Generative models* to learn the *joint* probability distribution $p(x, y)$. Among these, Generative Adversarial Networks (GANs) have become very popular to learn to generate new data with the same (statistical) properties as the training set. The two kinds of models are exemplified in Figure 1.

Another distinction is based on whether the learning task is centralized or (somewhat) distributed:

- *Centralized learning:* in conventional ML methodologies, all training data is pooled and stored at a single entity, and models are trained on this joint pool.
- *Collaborative/federated learning:* multiple participants, each with their own training dataset, construct a joint model by training a local model on their own data, but periodically exchange model parameters, updates to these parameters, or partially constructed models with the other participants. This intuition is illustrated in Figure 2. There are several techniques in this category, including federated learning deployed by Google and Apple on millions of devices, e.g., to train predictive keyboards on character sequences users type on their phones.

Machine Learning as a Service (MLaaS). Many cloud providers, including Microsoft, Amazon, and IBM, have launched Machine Learning as a Service (MLaaS) offerings, aimed to help clients benefit from machine learning without the cost, time, and risk of building in-house infrastructure from scratch. MLaaS offers ready-made,

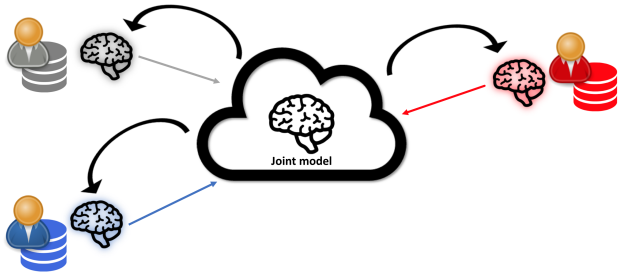

Figure 2: An overview of the federated learning approach.

generic machine learning tools, such as predictive analytics, APIs, data visualization, and natural language processing, that can be adapted by small and medium-sized companies according to their needs. Users who purchase MLaaS services can access these tools via prediction APIs on a pay-per-query basis. Typical image classification service costs around $1–$10 per 1,000 queries, depending on the customization and sophistication of the machine learning model.

MLaaS services vary a lot across different providers. In some cases, providers enable clients to download and deploy machine learning models locally, while others only allow clients to access machine learning models via a prediction query interface, which provides both the predicted label and the confidence score. The latter is much more popular. Some platforms also allow clients to upload their own models and charge others for using their models.

3 Privacy in ML

The security of any system is measured with respect to the adversarial goals and capabilities that it is designed to defend against; to this end, we now discuss different threat models. Then, we attempt to provide a definition of privacy in ML, focusing on the different types of attacks reviewed in detail in Section 4.

3.1 Adversarial Models

Overall, we focus on the privacy of the model. (NB: adversarial examples and overall robustness issues are out of the scope of this article.) In the rest of this section, we discuss adversarial goals related to extracting information about the model or training data.

When the model itself represents intellectual property—e.g., in financial market systems—the model and its parameters should be kept private. In other contexts, it is imperative that the privacy of the training data be preserved, e.g., in medical applications. Regardless of the goal, the attacks and defenses relate to exposing or preventing the exposure of the model and training data.

Access. We first discuss what kind of *access* the attacker might have:

- *White-Box:* she has some information about the model or its original training data, e.g., the ML algorithm, model parameters, network structure, or summary, partial, or full training data.
- *Black-Box:* she has no knowledge about the model. Rather, she might explore a model by providing a series of carefully crafted inputs and observing outputs.

Inference vs training. Another variable is *where* the attack might take place:

- *Training Phase:* the adversary attempts to learn the model, e.g., accessing a summary, partial, or all of the training data. She might create a substitute model (aka auxiliary model) to mount attacks on the victim system.
- *Inference Phase:* the adversary collects evidence about the model characteristics by observing inferences made by it.

Passive vs. Active. Finally, one can distinguish between passive and active attacks, roughly mirroring the traditional distinction in security literature between honest-but-curious and fully malicious adversaries. Consider, for instance, federated learning, where the attacker is one of the participants in the collaborative setting:

- *Passive attack:* the adversary passively observes the updates and performs inference, e.g., without changing anything in the training procedure;
- *Active attack:* the adversary actively changes the way she operates, e.g., in the case of federated learning, by extending their local copy of the collaboratively trained model with an augmented property classifier connected to the last layer.

3.2 Types of attacks

Before delving into the state of the art of actual attacks, we define what privacy means in the context of machine learning or, alternatively, what it means for a machine learning model to breach privacy.

3.2.1 Inference about members of the population

- *Statistical disclosure:* the adversary learns something about the input to the model from the model predictions; in theory, one would like to control statistical disclosure (this is also known as the "Dalenius desideratum"), in that a model should reveal no more about the input to which it is applied than would have been known about this input without applying the model. However, any useful model cannot achieve this.
- *Model inversion:* an adversary can use the model's output to infer the values of sensitive attributes used as input to the model. Note that it may not be possible to prevent this if the model is based on statistical facts about the population. For example, suppose that training the model has uncovered a high correlation between a person's externally observable phenotype features and their genetic predisposition to a certain disease; this correlation is now a publicly known fact that allows anyone to infer information about the person's genome after observing that person.

- *Inferring class representatives:* overall, model inversion can be generalized to potential breaches where the adversary, given some access to the model, infers features that characterize each class, making it possible to construct representatives of these classes.

3.2.2 Inference about members of the training dataset

Here the focus is on the privacy of the individuals whose data was used to train the model. Of course, members of the training dataset are members of the population, too. Therefore, one should focus on what the model reveals about them beyond what it reveals about an arbitrary member of the population:

- *Membership inference:* given a model and an exact data point, the adversary infers whether this point was used to train the model or not.
- *Property inference:* training data may not be identically distributed across different users whose records are in the training set; unlike model inversion, the adversary tries to infer properties that are true of a *subset* of the training inputs but not of the class as a whole. For instance, when Bob's photos are used to train a gender classifier, she infers that Alice appears in some photos.

3.2.3 Inferring Model Parameters

As discussed earlier, MLaaS allows model owners to charge others for queries to their commercially valuable models. This pay-per-query deployment option exemplifies an increasingly common tension: on the one hand, the query interface of an ML model may be widely accessible, yet the model itself and the data on which it was trained may be proprietary and confidential. Moreover, for security applications such as spam or fraud detection, an ML model's secrecy is critical to its utility; an adversary that can learn the model can also often evade detection.

In this space, we can distinguish between:

- *Model Extraction:* a black-box adversary that can query an ML model to obtain predictions on input feature vectors and may or may not know the model type (e.g., logistic regression) or the distribution over the data used to train the model. The adversary's goal is to extract an equivalent or near-equivalent ML model.
- *Functionality Stealing:* Rather than stealing the model, here the ultimate goal is to create "knock-offs" of the (black-box) model solely based on input-output pairs observed from MLaaS queries.

4 Attacks

4.1 Membership Inference Attack (MIA)

4.1.1 Definition and Relevance

Membership inference relates to the problem of deciding, given a data point, whether or not it was included in the training dataset. This can constitute a serious privacy breach in several settings, which we discuss next.

Sensitivity of task/model. First of all, MIA can directly violate privacy if inclusion in a training set is itself sensitive based on the nature of the task at hand. For example, if health-related records (or images like MRIs) are used to train a classifier, discovering that a specific record was used for training inherently leaks information about the individual's health. Similarly, if images from a database of criminals are used to train a model predicting the probability that one will re-offend, successful membership inference exposes an individual's criminal history.

Signal of leakage. When a record is fully known to the adversary, learning that it was used to train a particular model indicates information leakage through the model. Overall, MIA is often considered to be a signal—a measuring stick of sort—that access to a model leads to potentially serious privacy breaches. In fact, MIAs are often a gateway to further attacks: e.g., if the adversary infers that data of a victim is part of the information she has access to, she can mount other attacks, like profiling, property inference, etc.

Establishing wrongdoing. On the other hand, regulators can also use MIA to support the suspicion that a model was trained on personal data without an adequate legal basis or for a purpose not compatible with the data collection. For instance, DeepMind was recently found to have used personal medical records provided by the UK's National Health Service for purposes beyond direct patient care; the basis on which the data was collected.

MIA beyond machine learning. As a side note, we remark that MIAs have been studied not only in the context of machine learning but also in other fields. Overall, given a data point and a *"function"*, one can define membership inference as the problem of determining whether the point is part of the input to the function. Often, this function is some form of aggregation, and in fact, researchers have demonstrated the existence of successful MIAs against aggregate statistics in the context of genomic studies, location data, etc.

4.1.2 State of the Art

Attacking Machine Learning as a Service. MIA against black-box machine learning models was first studied by Shokri et al. [2], in the context of supervised learning. They focus on classification models trained by commercial Machine Learning as a Service (MLaaS) providers, such as Google and Amazon, whereby a user has API access to a trained model.

More specifically, customers in possession of a dataset and a data classification task can upload the dataset to the MLaaS service and pay it to construct a model. The service then makes the model available to the customer—typically as a black-box API. For example, a mobile app maker can use such a service to analyze users' activities and query the resulting model inside the app to promote in-app purchases to users when they are most likely to respond. Moreover, some machine-learning services also let data owners expose their models to external users for querying or even sell them.

Inference via overfitting. Shokri et al. [2]'s approach exploits differences in the model's response to inputs that were or were not seen during training. For each class of the targeted black-box model, they train a *shadow model*, with the same machine learning technique; the intuition is that the model ends up "overfitting" on data used for training. Overfitting is a modeling error that occurs when a function is too closely fit to a limited set of data points and performs better on the training inputs than on the inputs drawn from the same population but not used during the training. Therefore, the attacker can exploit the confidence values on inputs belonging to the same classes and learn to infer membership.

Generative models. While the research discussed above focuses on discriminative models, other work targets generative models. As discussed earlier, they are used to generate new samples from the same underlying distribution of a given training dataset, e.g., to artificially generate plausible images and videos. Here the attacker targets an MLaaS engine that provides synthetic samples on demand – e.g., the user's query is *"provide an image sample of a cat"* – based on a trained generative model. Once again, inferring whether specific data points are part of the training set for that generative model may constitute a serious privacy breach. Note that membership inference on generative models is much more challenging than on discriminative models: in the former, the attacker cannot exploit confidence values on inputs belonging to the same classes, and therefore it is more difficult to detect overfitting and mount the attack.

Hayes et al. [4] consider both black-box and white-box attacks: in the former, the adversary can only make queries to the model under attack, i.e., the target model, and has no access to the internal parameters. In the latter, she also has access to the parameters. To mount the attacks, they train a Generative Adversarial Network (GAN) on samples generated from the target model, i.e., using generative models to learn information about the target generative model, and thus create a local copy of the target model from which they can launch the attack. The intuition is that, if a generative model overfits, then a GAN–which combines a discriminative model and a generative model–should detect this overfitting since the discriminator is trained to learn statistical differences in distributions. Moreover, for white-box attacks, the attacker-trained discriminator itself can be used to measure information leakage of the target model.

Federated Learning. In this setting, the attack can be mounted by an adversary, a participant in the federated learning, attempting to infer whether a specific record is part of the training set of either a specific or any participant. The first MIA against federated learning is presented by Melis et al. [3], whose main intuition is to exploit unintended leakage from either the *embedding layer* (all deep learning models operating on non-numeric data where the input space is discrete and sparse first use an embedding layer to transform inputs into a lower-dimensional vector representation) or the *gradients* (in deep learning models, gradients are computed by back-propagating the loss through the entire network from the last to the first layer). An illustration of Melis et al. [3]'s attack is in Figure 3. Then, Nasr et al. [5] design MIAs during the training phase in a white-box setting, including passive and active attackers based on the different adversary prior knowledge.

4.2 Model Inversion

As mentioned earlier, model inversion techniques aim to infer class features and/or construct class representatives, given that the adversary has *some* access (either black-box or white-box) to a model.

4.2.1 Definition and Early Work

The concept of model inversion is introduced by Fredrikson et al. [6]. First, they show how an attacker can rely on outputs from a classifier to infer sensitive features used as inputs to the model itself: given the model and some demographic information about a patient whose records are used for training, an attacker might predict sensitive attributes of the patient. Then, they use so-called "hill-climbing" on the output probabilities of a computer-vision classifier to reveal individual faces from the training data.

These techniques are sometimes described as violating the privacy of the training data, even though the inferred features characterize an entire class and not specifically the training data, except in the cases of pathological overfitting where the training sample constitutes the entire membership of the class.

4.2.2 Further Attacks

Collaborative learning. Hitaj et al. [7] show that a participant in collaborative learning can use GANs to construct class representatives. However, this technique has been evaluated only on models where all members of the same class are visually similar (handwritten digits and faces). Thus, there is no evidence that it produces actual training images or can distinguish a training image and another image from the same class.

Aono et al. [8] show that, in collaborative deep learning, an adversarial server can partially recover participants' data points from the shared gradient updates, although in a greatly simplified setting where the batch consists of a single data point.

Unintended Memorization. Song et al. [9] engineer a machine learning model that memorizes the training data, which can then be extracted with black-box access to the model, without affecting the accuracy of the model on its primary task. Then, Carlini et al. [10] show that deep learning-based generative sequence models trained on text data can unintentionally memorize specific training inputs, which can then be extracted with black-box access. Even though the models are trained on text, extraction is demonstrated only for sequences of digits (artificially introduced into the text), which are not affected by the relative word frequencies in the language model.

4.3 Property Inference

As mentioned above, work presented in [6, 7, 11] aimed to infer properties that characterize an entire class: for ex-

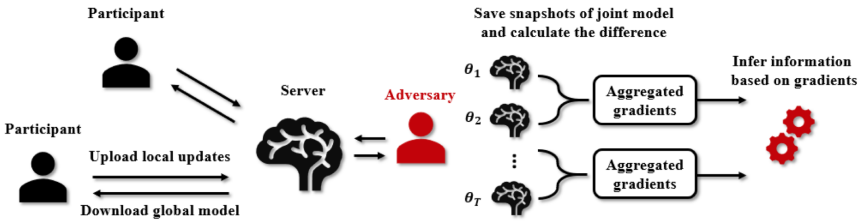

Figure 3: Inference attacks against federated learning (passive adversary) by Melis et al. [3].

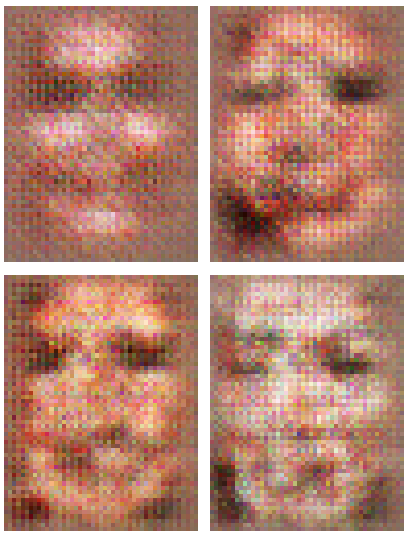

Figure 4: Samples from a GAN attack on a gender classification model where the class is "female."

ample, given a face recognition model where one of the classes is Bob, infer what Bob looks like (e.g., Bob wears glasses). However, while Ateniese et al. [11] are actually the first, to the best of our knowledge, to reason about extracting "something meaningful relating to properties of the training set," it is not clear that hiding this kind of information in a good classifier is possible or desirable.

4.3.1 Attacks

By contrast, here, we focus on the adversarial goal of inferring properties that are *true of a subset of the training inputs but not of the class as a whole*. For instance, when Bob's photos are used to train a gender classifier, can the attacker infer that Alice appears in some photos? In particular, Melis et al. [3] focus on the properties that are *independent* of the class's characteristic features. In contrast to the face recognition example, where "Bob wears glasses" is a characteristic feature of an entire class, in their gender classifier study, they infer whether people in Bob's photos wear glasses—even though wearing glasses does not correlate with gender. There is no "legitimate" reason for a model to leak this information; it is purely an artifact of the learning process.

The work in [3] studies this kind of property inference in the context of collaborative/federated learning. More specifically, their intuition is that a participant's contribution to each iteration of collaborative learning is based on a batch of their training data, and the adversary can infer *single-batch properties*, i.e., detect that the data in a given batch has the property, but other batches do not. She can also infer *when a property appears in the training data*, which has dire privacy implications. For instance, the adversary can infer when a certain person starts appearing in a participant's photos or when the participant starts visiting a certain type of doctor. Finally, they infer properties that characterize a participant's entire dataset (but not the entire class), e.g., authorship of the texts used to train a sentiment-analysis model.

4.4 Model and Functionality Stealing

Finally, we look into adversarial efforts toward inferring model parameters.

4.4.1 Model Extraction

The concept of model stealing, or extraction, is first presented by Tramer et al. [12]. In this kind of attack, an adversary with black-box access, but no prior knowledge of an ML model's parameters or training data, aims to steal the model parameters. The intuition behind their attack is to exploit the information-rich outputs returned by the ML prediction APIs, e.g., high-precision confidence values in addition to class labels.

Consider the case of ML algorithms like logistic regression: the confidence value is a simple log-linear function $1/(1+e^{-(w\cdot x+\beta)})$ of the d-dimensional input vector x. By querying $d+1$ random d-dimensional inputs, an attacker can with high probability solve for the unknown $d+1$ parameters w and β defining the model. (Such equation-solving attacks extend to multi-class logistic regressions and neural networks).

Overall, Tramer et al. [12]'s work is focused on inferring model parameters, but follow-up work also focuses on stealing hyperparameters, architectures, etc. In the former, the focus is on hyperparameters rather than parameters, which are configurations external to the model and whose values cannot be estimated from data. In the latter, a black-box adversary succeeds to infer (hidden) model architectures (e.g., the type of non-linear activation) of neural networks in MLaaS as well as their optimization processes (e.g., stochastic gradient descent or ADAM).

4.4.2 Functionality Extraction

As mentioned in Section 3.2, the goal of functionality extraction is, rather than to steal the model, to create "knock-offs." In [13], Orekondy et al. do so solely based on input-output pairs observed from MLaaS queries. The adversary interacts with a black-box "victim" Convolutional Neural

Network (CNN) by providing it input images and obtaining respective predictions. The resulting image-prediction pairs are used to train a knock-off model, e.g., to compete with the victim model at the victim's task.

4.5 Defenses

Overall, defenses against attacks discussed above include advanced *privacy-enhancing technologies* like cryptography and differential privacy as well approaches used as part of the learning process (mainly, training) to reduce the information available to the adversary.

Cryptography Cryptography in ML can support confidential computing scenarios where, for instance, a server has a model trained on its private data and wishes to provide inferences (e.g., classification) on clients' private data. In this context, there are many research proposals and prototypes in literature, which allow the client to obtain the inference result without revealing their input to the server while preserving the confidentiality of the server's model. For instance, privacy-enhancing tools based on secure multi-party computation (SMC) and fully homomorphic encryption (FHE) could be used to train ML models securely.

Overall, cryptography in ML is really aimed at protecting *confidentiality*, rather than *privacy*, which constitutes the main focus of our report. Confidentiality is an explicit design property whereby one party wants to keep information (e.g., training data, testing data, model parameters, etc.) hidden from both the public and other parties (e.g., clients with respect to servers or vice-versa). Whereas privacy is about protecting against *unintended* information leakage, whereby an adversary aims to infer sensitive information through some (intended) interaction with the victim. In other words, cryptographically-enforced confidential computing does not provide any guarantees about what the output of the computation reveals.

Differential Privacy (DP). The state-of-the-art method for providing access to information in a private way is to satisfy differential privacy (DP). DP addresses the paradox of learning nothing about an individual while learning useful information about a population; generally speaking, it provides rigorous, statistical guarantees against what an adversary can infer from learning the result of some randomized algorithm. Typically, differentially private techniques protect the privacy of individual data subjects by adding random noise when producing statistics. DP guarantees that an individual will be exposed to the same privacy risk whether or not her data is included in a differentially private analysis.

This applies to ML as well, and more precisely to providing access to models that have been trained on (sensitive) datasets. However, there is no one-size-fits-all solution, and, as discussed later, the privacy-utility trade-offs are not particularly promising across the board. In other words, as DP in ML relies on adding noise, it does affect the utility of the learning tasks; alas, settings that provide limited accuracy loss often provide little privacy, and vice versa settings that provide strong privacy result in useless models.

Trusted Execution Environments. A different line of work focuses on privacy (as well as integrity) guarantees for ML computations in untrusted environments (i.e., tasks outsourced by a client to a remote server, including MLaaS) by leveraging so-called Trusted Execution Environments (TEEs), such as Intel SGX or ARM TrustZone. TEEs use hardware and software protections to isolate sensitive code from other applications while attesting to its correct execution. The main idea is that TEEs outperform purely cryptographic approaches by multiple orders of magnitude. However, these approaches are increasingly targeted by side-channel attacks, whereby information can still leak out of the TEEs, ultimately compromising the systems' security.

ML-Specific Approaches. Finally, several ML techniques are used to reduce the information available to the adversary to mount their attacks. For instance, *dropout* is a regularization method for neural networks, often used to mitigate overfitting in neural networks; as such, this might reduce the effectiveness of MIAs based on overfitting. Additional techniques in this space include weight normalization (re-parameterization of the weights vectors that decouples the length of those weights from their direction), dimensionality reduction (e.g., only using inputs that occur many times in the training data), selective gradient sharing (in collaborative learning, participants could share only a fraction of their gradients during each update), etc. However, in many settings, these approaches provide very little/not particularly robust privacy defenses [3].

5 Discussion

We provided a review on privacy and machine learning, presenting a wide range of attacks that relate to private and/or sensitive information leakage. Next, we provide a discussion of the main takeaways and list areas where further work is needed.

5.1 What Do Attacks Mean?

Membership inference attacks are real. As evident from the above discussion, there has been a very significant amount of research work on membership inference attacks against ML. Arguably, this is motivated by 1) the seriousness of the privacy risks stemming from such attacks, 2) the fact that MIA is often just a signal of leakage and can serve as a canary for broad privacy issues, and 3) the interesting challenges in making attacks more effective, less reliant on strong assumptions, etc.

Several attacks have been proposed in the context of a wide variety of datasets (images, text, etc.), models (discriminative, generative, federated), as well as threat models (API access, white-box, black-box, active, passive, etc.). Such attacks are realistic, but obviously their effectiveness depends on the actual settings, e.g., adversary's knowledge of records, model parameters, etc., and are likely to affect certain users more than others.

Overall, we are confident in arguing that MIAs are a real problem that, at the very least, should make practitioners and researchers question whether deploying ML models

in the wild is a good idea, privacy-wise, whenever training data is sensitive. However, further work is needed to provide clear guidelines and usable tools for practitioners willing to provide access to trained models to fully understand the privacy risks, on their specific data/specific learning task, for the users whose data is used for training. In other words, MIAs are very much possible, but it is hard to grasp the real-world effect on actually deployed models due to the lack of case studies vis-à-vis impact on actual users, relation to adversary's prior knowledge, etc. Much work is left to be done here – especially considering ways to provide guidelines and evaluation framework for practitioners.

Limitations of model inversion. Although research roughly falling in the "model inversion" category is important, we believe there are some limitations in what they mean for privacy. Class members produced by model inversion and GANs are similar to the training inputs only if all class members are similar, as is the case for MNIST (the dataset of handwritten digit used in [7]) and facial recognition. This does not violate the privacy of the training data; it simply shows that machine learning works as it should. A trained classifier reveals the input features characteristic of each class, thus enabling the adversary to sample from the class population. For instance, Figure 4 shows GAN-constructed images for the gender classification task on the Labeled Faces in the Wild (LFW) dataset, taken from [3]. These images show a generic female face, but there is no way to tell from them whether an image of a specific female was used in training or not.

Therefore, the informal property violated by such attacks is, roughly speaking: "a classifier should prevent users from generating an input that belongs to a particular class or even learning what such an input looks like." However, it is not clear why this property is desirable or whether it is even achievable. In fact, this motivated us to study what we defined as property inference attacks.

However, there are also cases where model inversion is also due to model overfitting on training data, as correlations between multiple attacks occur [14]. To some extent, this calls for further work to study scenarios where the attacker might indeed benefit from having access to the target model.

Property inference needs further work. Overall, property inference attacks are not to be ignored, even though their effectiveness depends on the context. As mentioned earlier, inferring sensitive attributes is really a privacy breach when the attacker can confidently assess that those attributes are related to records in the training set. Even more so if they do not leak simply because the class the model is learning to classify is strictly correlated.

So really, the only "attack" in this sense we are aware of is that of Melis et al. [3], which has only been studied in the context of collaborative learning. Even in that case, the authors essentially show that the accuracy of the attack quickly degrades with an increasing number of participants. In fact, if this is large enough, then differentially private defenses based on the moments accountant method [15] could be used to thwart such attacks.

It remains, however, an open research question to investigate whether property inference attacks: 1) are possible, as per our definition, in non-collaborative learning settings and at scale, and 2) can be thwarted in collaborative settings involving a small number of participants.

5.2 Policy Implications and Further Study Needed

The implication of the attacks covered in this manuscript vis-a-vis policy and data protection is also largely unexplored. The only exception in this context is the work by Cohen and Nissim [16], which rephrases privacy attacks in the General Data Protection Regulation (GDPR) framework and, more specifically, within its *"singling out"* concept. While the GDPR heavily focuses on the concept of identification, what it means for a person to be "identified, directly or indirectly" is not clear. As pointed in [16], Recital 26 sheds a little more light: "To determine whether a natural person is identifiable, 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."

Therefore, singling out is one way to identify a person in data, and only data that does not allow singling out may be excepted from the regulation. Clearly, more work linking up privacy attacks (and defenses) with regulation and data protection efforts needs to ramp up.

5.3 Need for Better Evaluations

Overall, several defense techniques against privacy attacks have been proposed over the past few years. However, it is very hard to assess how generalizable they are and the trade-off they incur regarding privacy and utility. This prompts the need for a more thorough evaluation of how defenses fare in practice, vis-a-vis realistic use cases and datasets, rather than the standard public ones that, more often than not, say little or nothing about real-world performance.

In this context, some recent work has taken some good steps in the right direction; for instance, Jayaraman and Evans [17] study the impact of variable choices of the ϵ parameter, different variants of differential privacy, and several learning tasks on both utility and privacy (including in the context of MIAs) for privacy-preserving machine learning. Alas, however, their main finding is that there is no way to obtain privacy for free–relaxed definitions of differential privacy that reduce the amount of noise needed to improve utility also increase the privacy leakage. In other words, current mechanisms for differentially private machine learning rarely offer acceptable utility-privacy trade-offs for complex learning tasks: settings that provide limited accuracy loss provide little effective privacy, and settings that provide strong privacy result in useless models.

Once again, this points to the need to understand better where trade-offs are possible, in what context, and at what expenses, rather than hoping to deploy generic, one-size-fits-all defenses across the board.

References

[1] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, and Z. Lin, "When Machine Learning Meets Privacy: A Survey and Outlook," *ACM Computing Surveys*, vol. 54, no. 2, Mar. 2021.

[2] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *IEEE S&P*, 2017.

[3] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *IEEE Symposium on Security and Privacy*, 2019.

[4] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "Logan: Membership inference attacks against generative models," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 1.

[5] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *IEEE Symposium on Security and Privacy*, 2019.

[6] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *USENIX Security*, 2014.

[7] B. Hitaj, G. Ateniese, and F. Pérez-Cruz, "Deep models under the GAN: information leakage from collaborative deep learning," in *ACM CCS*, 2017.

[8] Y. Aono, T. Hayashi, L. Wang, S. Moriai *et al.*, "Privacy-preserving deep learning: Revisited and Enhanced," in *ATIS*, 2017.

[9] C. Song, T. Ristenpart, and V. Shmatikov, "Machine learning models that remember too much," in *ACM CCS*, 2017.

[10] N. Carlini, C. Liu, J. Kos, Ú. Erlingsson, and D. Song, "The Secret Sharer: Measuring Unintended Neural Network Memorization & Extracting Secrets," *arXiv preprint 1802.08232*, 2018.

[11] G. Ateniese, L. V. Mancini, A. Spognardi, A. Villani, D. Vitali, and G. Felici, "Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers," *IJSN*, 2015.

[12] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction APIs," in *USENIX Security*, 2016.

[13] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff nets: Stealing functionality of black-box models," in *IEEE CVPR*, 2019.

[14] Y. Liu, R. Wen, X. He, A. Salem, Z. Zhang, M. Backes, E. De Cristofaro, M. Fritz, and Y. Zhang, "ML-Doctor: Holistic Risk Assessment of Inference Attacks Against Machine Learning Models," *arXiv:2102.02551*, 2021.

[15] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *ACM CCS*, 2016.

[16] A. Cohen and K. Nissim, "Towards Formalizing the GDPR's Notion of Singling Out," *arXiv preprint arXiv:1904.06009*, 2019.

[17] B. Jayaraman and D. Evans, "Evaluating Differentially Private Machine Learning in Practice," in *USENIX Security*, 2019.