A Critical Overview of Privacy in Machine Learning

Emiliano De Cristofaro | UCL and Alan Turing Institute

This article reviews privacy challenges in machine learning and provides a critical overview of the relevant research literature. The possible adversarial models are discussed, a wide range of attacks related to sensitive information leakage is covered, and several open problems are highlighted.

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 ML-as-a-service (MLaaS) engines to outsource complex tasks, e.g., training classifiers, performing predictions, and so on. 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. Unfortunately, if malicious users were to recover the 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, the privacy challenges in this space are examined, while providing a systematic review of the relevant research literature.

Discussed are the possible adversarial models and settings, which cover a wide range of attacks related to private and/or sensitive information leakage, and the recent results, which attempt to defend against such attacks, are briefly investigated. Finally, a list of open problems that require more work is presented, including the need for better evaluations, targeted defenses, and the study of the relationship to policy and data protection efforts.

Digital Object Identifier 10.1109/MSEC.2021.3076443
Date of current version: 18 May 2021

This article does not offer a comprehensive survey of the literature in the field nor an exhaustive list of all the threat models and attacks to privacy in ML; interested readers may refer to the existing surveys, e.g., in the work of Liu et al.[1]

ML Background

ML Approaches

ML models can be categorized according to the probability distributions that they learn. In supervised learning, assuming one has some input data x (e.g., pictures of animals) and wants to classify them into labels y (e.g., types of animal), then, roughly speaking, one can use either

- discriminative models to learn the conditional probability distribution $p(x|y)$ and ultimately learn to distinguish from among different classes (e.g., cats versus dogs)
- generative models to learn the joint probability distribution $p(x,y)$. Among these, generative adversarial networks (GANs) have become very popular as a way to generate new data with the same (statistical) properties as the training set. The two kinds of models are displayed in Figure 1.

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

- *Centralized learning*: In conventional ML methodologies, all of the training data are pooled and stored

at a single entity, and the models are trained on this joint pool.

- *Collaborative/federated learning*: Multiple participants, each with their own training data set, 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 the character sequences users type on their phones.

MLaaS

Many cloud providers, including Microsoft, Amazon, and IBM, have launched MLaaS offerings, which are aimed at helping clients benefit from ML without the cost, time, and risk of building in-house infrastructure from scratch. MLaaS offers ready-made, generic ML tools, such as predictive analytics, application programming interfaces (APIs), data visualization, and natural language processing, which can be adapted by small- and medium-sized companies according to their needs. The users who purchase MLaaS services can access these tools via prediction APIs on a pay-per-query basis. A typical image-classification service costs approximately US$1–US$10 per 1,000 queries, depending on the customization and sophistication of the ML model.

MLaaS services vary considerably across different providers. In some cases, providers enable clients to download and deploy ML models locally, while others allow clients to access ML models only 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 them.

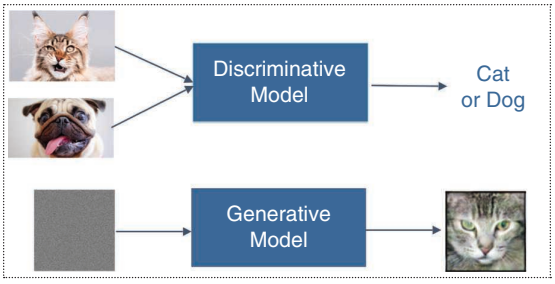

Figure 1. An example of a discriminative and a generative model. The former learns to distinguish from between two classes, i.e., pictures of cats or dogs. The latter estimates the underlying distribution of a data set (pictures of cats) and randomly generates realistic, yet synthetic, samples according to their estimated distribution.

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, different threat models are now discussed. Then, an attempt to provide a definition of privacy in ML is provided, which focuses on the different types of attacks that are reviewed in detail in the "Attacks" section.

Adversarial Models

Overall, we focus on the privacy of the model. (note that adversarial examples and overall robustness issues are outside the scope of this article.) In this section, the adversarial goals related to the extraction of information about the model or the training data are discussed.

When the model itself represents intellectual property, e.g., in financial market systems, the model and its

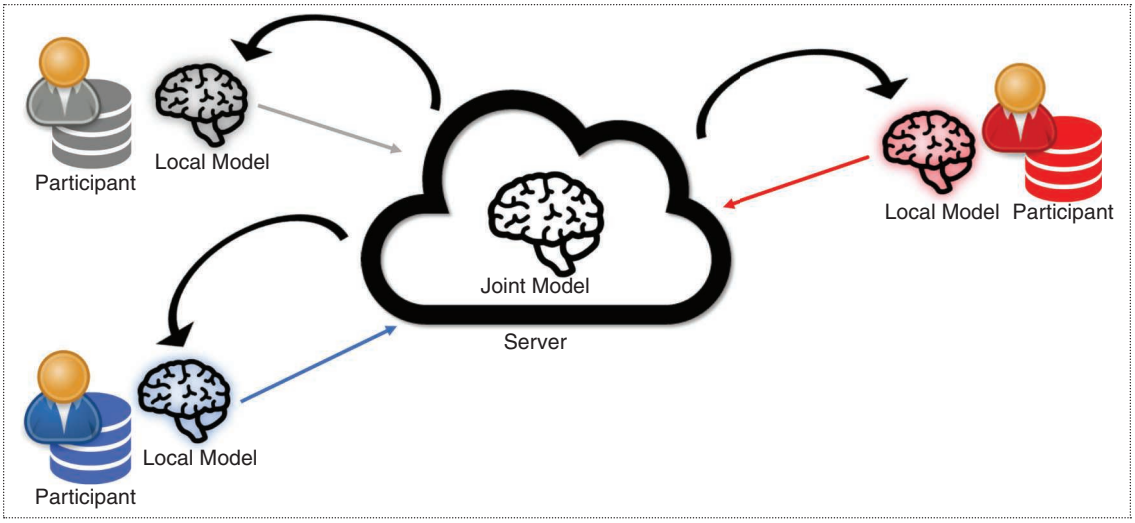

Figure 2. An overview of the federated learning approach.

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 the training data.

The kind of access the attacker might have can be either

- white box, where the adversary has some information about the model or its original training data, e.g., the ML algorithm, model parameters, or network structure; or the summary, partial, or full training data.
- black box, where the adversary has no knowledge about the model. Rather, he/she might explore a model by providing a series of carefully crafted inputs and by observing the outputs.

Another variable to consider is where the attack might take place:

- *Training phase*: In this phase, the adversary attempts to learn the model, e.g., accessing the summary, partial, or full training data. He/she might create a substitute model (also known as an auxiliary model) to mount attacks on the victim's system.
- *Inference phase*: In this phase, the adversary collects evidence about the model's characteristics by observing the inferences made by it.

Finally, one can distinguish passive from active attacks, roughly mirroring the traditional distinction in the security literature between honest-but-curious and fully malicious adversaries.

- *Passive attack*: In this type of attack, the adversary passively observes the updates and performs inference, e.g., without changing anything in the training procedure.
- *Active attack*: In this type of attack, the adversary actively changes the way he/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.

Types of Attacks

Before delving into the state of the art of actual attacks, we define what privacy means in the context of ML or, alternatively, what it means for an ML model to breach privacy. The following inferences about members of the population can be made.

- *Statistical disclosures*: With this inference, the adversary learns something about the input to the model from the model's predictions; in theory, one would like to control statistical disclosures (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.

When inferring about the members of the training data set, the focus is on the privacy of the individuals whose data was used to train the model. Of course, members of the training data set 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, he/she infers that Alice appears in some photos.

Inferring model parameters. As discussed previously, 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 knockoffs of the (black-box) model solely based on the input–output pairs observed from MLaaS queries.

Attacks

Definition and Relevance

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

Sensitivity of the task/model. First of all, a membership inference attack (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 magnetic resonance imaging scans) 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 reoffend, successful membership inference exposes an individual's criminal history.

Signal of leakage. When a record is known to the adversary, learning that it was used to train a particular model indicates information leakage through the model. Overall, an MIA is often considered a signal—a measuring stick of sorts—that access to a model leads to potentially serious privacy breaches. In fact, MIAs are often gateways to further attacks; for example, if the adversary infers that the data of a victim are part of the information he/she has access to, he/she can mount other attacks, like profiling, property inference, and so forth.

Establishing wrongdoing. On the other hand, regulators can also use an 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 United Kingdom's National Health Service for purposes beyond direct patient care, the basis on which the data was collected.

MIAs beyond ML. As a side note, we remark that MIAs have been studied not only in the context of ML 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 a 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, and so on.

State of the Art

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

More specifically, the customers in possession of a data set and a data-classification task can upload the data set 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 ML services also let data owners expose their models to external users for querying or even sell them.

Inference via overfitting. Shokri et al.'s approach[2] exploits the 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 using the same ML technique; the intuition is that the model ends up "overfitting" on the 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 the inputs belonging to the same classes and learn to infer membership.

Generative models. Although the aforementioned research focuses on discriminative models, other work targets generative models. As discussed previously, the models are used to generate new samples from the same underlying distribution of a given training data set, e.g., to artificially generate plausible images and videos. Here the attacker targets an MLaaS engine that provides synthetic samples on demand; for example, 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 ones: In the former, the attacker cannot exploit confidence values on the inputs belonging to the same classes. It is therefore more difficult to detect overfitting and mount the attack.

Hayes et al.[4] consider both black- and white-box attacks. In the former, the adversary can make queries to only the model under attack, i.e., the target model, and has no access to the internal parameters. In the latter, he/she also has access to the parameters. To mount the attacks, he/she trains a GAN on the samples generated from the target model, i.e., using generative models to learn information about the target generative model, thus creating a local copy of the target model from which they can launch the attack. The thinking is that, if a generative model overfits, then a GAN, which combines a discriminative and a generative model, should detect this overfitting because the discriminator is trained to learn the statistical differences in distributions. Moreover, for white-box attacks, the attacker-trained discriminator itself can be used to measure the 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 particular 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 of the deep learning models that operate on nonnumeric 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 backpropagating the loss through the entire network from the last to the first layer). An illustration of Melis et al.'s attack[3] is presented 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's prior knowledge.

Model Inversion

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

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 the 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 "hill climbing" on the output probabilities of a computer-vision classifier to reveal individual faces in 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.

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 on only the 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 from another image in 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 an ML model that memorizes the training data, which can then be extracted using 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

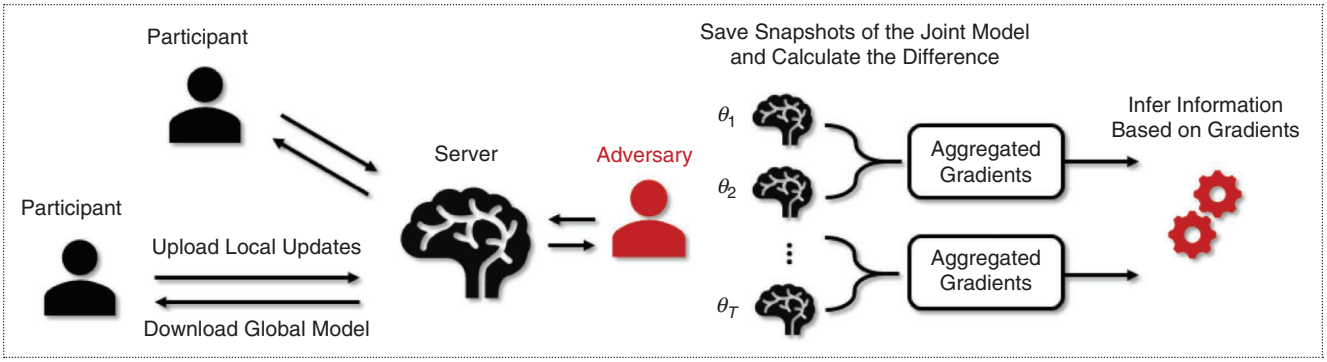

Figure 3. Inference attacks against federated learning (passive adversary).[3]

training inputs, which can then be extracted using black-box access. Even though the models are trained on text, extraction is demonstrated for only sequences of digits (artificially introduced into the text), which are not affected by the relative word frequencies in the language model.

Property Inference

As mentioned in previous sections, there has been work[6,7,11] that aims to infer the properties that characterize an entire class: For example, given a facial-recognition model where one of the classes is Bob, infer what Bob looks like (e.g., Bob wears glasses). However, although Ateniese et al.[11] were actually the first (to the author's best 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.

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 of the photos? In particular, Melis et al.[3] focus on the properties that are independent of the class's characteristic features. In contrast to the facial-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 of Melis et al.[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 that other batches do not. He/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 the properties that characterize a participant's entire data set (but not the entire class), e.g., authorship of the texts used to train a sentiment-analysis model.

Model and Functionality Stealing

Model extraction. Finally, we look into adversarial efforts toward inferring model parameters. The concept of model stealing, or extraction, is first presented by Tramèr 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 inspiration for the attack is to exploit the information-rich outputs returned by the ML prediction APIs, e.g., high-precision confidence values in addition to the 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 multiclass logistic regressions and neural networks.)

Overall, Tramèr et al.'s work[12] is focused on inferring model parameters, but follow-up work also focuses on stealing hyperparameters, architectures, and so forth. 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 nonlinear activation) of neural networks in MLaaS as well as their optimization processes (e.g., stochastic gradient descent or Adam).

Functionality extraction. As mentioned in the "Types of Attacks" section, the goal of functionality extraction is, rather than to steal the model, to create knockoffs. Orekondy et al.[13] do so based solely on the input–output pairs observed from MLaaS queries. The adversary interacts with a black-box "victim" convolutional neural network by providing it input images and obtaining the 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.

Defenses

Overall, the aforementioned defenses against attacks include advanced privacy-enhancing technologies like cryptography and differential privacy (DP). They also comprise the 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 the literature that 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 multiparty computations and fully homomorphic encryption could be used to train ML models securely.

Overall, cryptography in ML is aimed at protecting confidentiality, rather than privacy, which constitutes the main focus of this article. Confidentiality is an explicit design property whereby one party wants to keep information (e.g., training and testing data, model parameters, and so on) 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.

DP. The state-of-the-art method for providing access to information in a private way is to satisfy 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 his/her data are 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) data sets. However, there is no one-size-fits-all solution and, as discussed later, the privacy-utility tradeoffs 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. Unfortunately, the settings that provide limited accuracy loss often provide little privacy, and vice versa the 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., the 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 where 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 and is often used to mitigate overfitting in neural networks; as such, this might reduce the effectiveness of MIAs based on overfitting. The additional techniques in this space include weight normalization (a reparameterization of the weight vectors that decouple the length of those weights from their direction), dimensionality reduction (e.g., using only the 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), and so forth. However, in many settings, these approaches provide very few/not particularly robust privacy defenses.[3]

Discussion

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

What Do the Attacks Mean?

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

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

Overall, 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 privacywise whenever training data are sensitive. However, further work is needed to provide clear

guidelines and usable tools for the practitioners willing to provide access to trained models to fully understand the privacy risks, on their specific data/specific learning tasks, and for the users whose data are 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 their impact on actual users, relationship to an adversary's prior knowledge, and so on. Much work is left to be done here, especially considering the ways in which guidelines and evaluation frameworks are provided for practitioners.

Limitations of model inversion. Although the research that roughly falls into the model-inversion category is important, there are some limitations to what they mean for privacy. The class members produced by model inversion and GANs are similar to the training inputs only if all the class members are similar, as is the case for the Modified National Institute of Standards and Technology (the data set of handwritten digits[7]) and facial recognition. This does not violate the privacy of the training data, it simply shows that ML 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" data set taken from the work of Melis et al.[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.

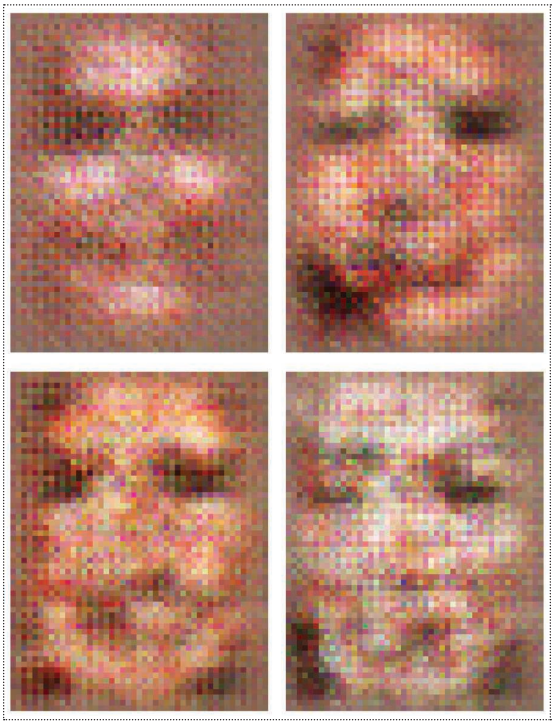

Figure 4. The samples from a GAN attack on a gender-classification model where the class is "female."[3]

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 the study of what are defined as *property inference attacks*.

There are, however, cases where model inversion is also due to model overfitting on the training data as correlations between multiple attacks occur.[14] To some extent, this calls for further work to study the 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 previously, inferring sensitive attributes is actually 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 actually, the only "attack" in this sense that we are aware of is that of Melis et al.,[3] which has been studied only 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 number 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 noncollaborative learning settings and at scale and 2) can be thwarted in collaborative settings involving a small number of participants.

Policy Implications and Further Study Needed

The implication of the attacks covered in this article vis-à-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. Although the GDPR focuses heavily on the concept of identification, what it means

for a person to be "identified, directly or indirectly" is not clear. As pointed out by Cohen and Nissim,[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 the data that do not allow singling out may be excepted from the regulation. Clearly, more work that links up privacy attacks (and defenses) with regulation and data protection efforts needs to ramp up.

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 tradeoff they incur regarding privacy and utility. This prompts the need for a more thorough evaluation of how defenses fare in practice, vis-à-vis realistic use cases and data sets, 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 steps in the right direction; for instance, Jayaraman and Evans[17] study the impact of variable choices of the ε parameter, different variants of DP, and several learning tasks on both utility and privacy (including in the context of MIAs) for privacy-preserving ML. Unfortunately, however, their main finding is that there is no way to obtain privacy for free; relaxed definitions of DP that reduce the amount of noise needed to improve utility also increase privacy leakage. In other words, the current mechanisms for differentially private ML rarely offer acceptable utility-privacy tradeoffs for complex learning tasks: The settings that offer limited accuracy loss provide little effective privacy, and the settings that afford strong privacy result in useless models. Once again, this points to the need to better understand where tradeoffs are possible, in what context, and at what expense, 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 Comput. Surv.*, vol. 54, no. 2, Mar. 2021.
2. R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Security Privacy*, 2017.
3. L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symp. Security Privacy*, 2019.
4. J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "Logan: Membership inference attacks against generative models," *Proc. Privacy Enhan. Technol.*, 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 *Proc. IEEE Symp. Security 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 *Proc. USENIX Security*, 2014.
7. B. Hitaj, G. Ateniese, and F. Pérez-Cruz, "Deep models under the GAN: Information leakage from collaborative deep learning," in *Proc. ACM Conf. Computer and Communications Security*, 2017.
8. L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning: Revisited and enhanced," in *Proc. Applications and Techniques in Information Security*, 2017.
9. C. Song, T. Ristenpart, and V. Shmatikov, "Machine learning models that remember too much," in *Proc. ACM Conf. Computer Communications Security*, 2017.
10. N. Carlini, C. Liu, J. Kos, Ú. Erlingsson, and D. Song, "The secret sharer: Measuring unintended neural network memorization & extracting secrets," 2018, arXiv:1802.08232.
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," *Int. J. Security Netw.*, vol. 10, no. 3, Sept. 2015. doi: 10.1504/IJSN.2015.071829.
12. F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction APIs," in *Proc. USENIX Security*, 2016.
13. T. Orekondy, B. Schiele, and M. Fritz, "Knockoff nets: Stealing functionality of black-box models," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2019.
14. Y. Liu et al., "ML-Doctor: Holistic risk assessment of inference attacks against machine learning models," 2021, arXiv:2102.02551.
15. M. Abadi et al., "Deep learning with differential privacy," in *Proc. ACM Conf. Computer and Communications Security*, 2016.
16. A. Cohen and K. Nissim, "Towards Formalizing the GDPR's notion of singling out," 2019, arXiv:1904.06009.
17. B. Jayaraman and D. Evans, "Evaluating differentially private machine learning in practice," in *Proc. USENIX Security*, 2019.

Emiliano De Cristofaro is an associate professor with the University College London, London, WC1E 6BT, U.K. He earned his Ph.D. from the University of California, Irvine, in 2011. His research interests include privacy and security. Contact him at e.decristofaro@ucl.ac.uk.