A Survey of Privacy Attacks in Machine Learning

MARIA RIGAKI and SEBASTIAN GARCIA, Czech Technical University in Prague, Czech Republic

As machine learning becomes more widely used, the need to study its implications in security and privacy becomes more urgent. Although the body of work in privacy has been steadily growing over the past few years, research on the privacy aspects of machine learning has received less focus than the security aspects. Our contribution in this research is an analysis of more than 45 papers related to privacy attacks against machine learning that have been published during the past seven years. We propose an attack taxonomy, together with a threat model that allows the categorization of different attacks based on the adversarial knowledge, and the assets under attack. An initial exploration of the causes of privacy leaks is presented, as well as a detailed analysis of the different attacks. Finally, we present an overview of the most commonly proposed defenses and a discussion of the open problems and future directions identified during our analysis.

CCS Concepts: • **Computing methodologies** → **Machine learning**; • **Security and privacy**;

Additional Key Words and Phrases: Privacy, machine learning, membership inference, property inference, model extraction, reconstruction, model inversion

ACM Reference format:
Maria Rigaki and Sebastian Garcia. 2023. A Survey of Privacy Attacks in Machine Learning. *ACM Comput. Surv.* 56, 4, Article 101 (November 2023), 34 pages.
https://doi.org/10.1145/3624010

1 INTRODUCTION

Fueled by large amounts of available data and hardware advances, machine learning has experienced tremendous growth in academic research and real-world applications. At the same time, the impact on the security, privacy, and fairness of machine learning is receiving increasing attention. In terms of privacy, our personal data are being harvested by almost every online service and are used to train models that power machine learning applications. However, it is not well known if and how these models reveal information about the data used for their training. If a model is trained using sensitive data such as location, health records, or identity information, then an attack that allows an adversary to extract this information from the model is highly undesirable. At the same time, if private data has been used without its owners' consent, the same type of attack could be used to determine the unauthorized use of data and thus work in favor of the user's privacy.

Apart from the increasing interest on the attacks themselves, there is a growing interest in uncovering what causes privacy leaks and under which conditions a model is susceptible to different

This work was partially supported by Avast Software and the OP RDE funded project Research Center for Informatics No. CZ.02.1.01/0.0./0.0./16_019/0000765.

Authors' address: M. Rigaki and S. Garcia, Czech Technical University in Prague, Karlovo náměstí 13, Prague, Czech Republic, 120 00; e-mails: maria.rigaki@fel.cvut.cz, sebastian.garcia@agents.fel.cvut.cz.

0360-0300/2023/11-ART101
https://doi.org/10.1145/3624010

types of privacy-related attacks. There are multiple reasons why models leak information. Some of them are structural and have to do with the way models are constructed, while others are due to factors such as poor generalization or memorization of sensitive data samples. Training for adversarial robustness can also be a factor that affects the degree of information leakage.

The focus of this survey is the privacy and confidentiality attacks on machine learning algorithms. That is, attacks that try to extract information about the training data or to extract the model itself. Some existing surveys [8, 103] provide partial coverage of privacy attacks and there are a few other peer-reviewed works on the topic [2, 52]. However, these papers are either too high level or too specialized in a narrow subset of attacks.

The security of machine learning and the impact of adversarial attacks on the performance of the models have been widely studied in the community, with several surveys highlighting the major advances in the area [8, 72, 80, 104, 132]. Based on the taxonomy proposed in Reference [8], there are three types of attacks on machine learning systems: (i) attacks against integrity, e.g., evasion and poisoning backdoor attacks that cause misclassification of specific samples, (ii) attacks against a system's availability, such as poisoning attacks that try to maximize the misclassification error, and (iii) attacks against privacy and confidentiality, i.e., attacks that try to infer information about user data and models. While all attacks on machine learning are adversarial in nature, the term "adversarial attacks" is commonly used to refer to security-related attacks and more specifically to adversarial samples. In this survey, we only focus on privacy and confidentiality attacks.

An attack that extracts information about the model's structure and parameters is, strictly speaking, an attack against model confidentiality. The decision to include model extraction attacks was made, because in the existing literature, attacks on model confidentiality are usually grouped together with privacy attacks [8, 104]. Another important reason is that stealing model functionality may be considered a privacy breach as well. Veale et al. [127] made the argument that privacy attacks such as membership inference (Section 4.1) increase the risk of machine learning models being classified as personal data under European Union's **General Data Protection Regulation (GDPR)** law, because they can render a person identifiable. Although models are currently not covered by the GDPR, it may happen that they will be considered as personal data, and then attacks against them may fall under the same scope as attacks against personal data. This may be further complicated by the fact that model extraction attacks can be used as a stepping stone for other attacks.

This article is, as far as we know, the first *comprehensive* survey of privacy-related attacks against machine learning. It reviews and systematically analyzes over 50 research papers. The papers have been published in top tier conferences and journals in the areas of security, privacy, and machine learning during 2014–2022. An initial set of papers was selected in Google Scholar using keyword searches related to "privacy," "machine learning," and the names of the attacks themselves ("membership inference," "model inversion," "property inference," model stealing," "model extraction," etc.). After the initial set of papers was selected, more papers were added by backward search based on their references as well as by forward search based on the papers that cited them.

The main contributions of this article are:

- The first comprehensive study of attacks on privacy and confidentiality of machine learning systems.
- A unifying taxonomy of attacks against machine learning privacy.
- A discussion on the probable causes of privacy leaks in machine learning systems.
- An in-depth presentation of the implementation of the attacks.
- An overview of the different defensive measures tested to protect against the different attacks.

1.1 Organization of the Article

The rest of the article is organized as follows: Section 2 introduces some basic concepts related to machine learning that are relevant to the implementation of the attacks, which are presented in Section 6. The threat model is presented in Section 3 and the taxonomy of the attacks and their definition are the focus of Section 4. In Section 5, we present the causes of machine learning leaks that are known or have been investigated so far. An overview of the proposed defences per attack type is the focus of Section 7. Finally, Section 8 contains a discussion on the current and future research directions and Section 9 offers concluding remarks.

2 MACHINE LEARNING

Machine learning (ML) is a field that studies the problem of learning from data without being explicitly programmed. The purpose of this section is to provide a non-exhaustive overview of machine learning as it pertains to this survey and to facilitate the discussion in the subsequent chapters. We briefly introduce a high level view of different machine learning paradigms and categorizations as well as machine learning architectures. Finally, we present a brief discussion on model training and inference. For the interested reader, there are several textbooks such as References [9, 33, 88, 113] that provide a thorough coverage of the topic.

2.1 Types of Learning

At a very high level, ML is traditionally split into three major areas: *supervised*, *unsupervised*, and *reinforcement* learning. Each of these areas has its own subdivisions. Over the years, new categories have emerged to capture types of learning that cannot easily fit under these three areas such as *semi-supervised* and *self-supervised* learning, or other ways to categorize models such as *generative* and *discriminative* ones.

2.1.1 Supervised Learning. In a supervised learning setting, a model f with parameters θ is a mapping function between inputs $\mathbf{x}$ and outputs $\mathbf{y} = f(\mathbf{x}; \theta)$, where $\mathbf{x}$ is a vector of attributes or features with dimensionality n. The output or label $\mathbf{y}$ can assume different dimensions depending on the learning task. A training set $\mathcal{D}$ used for training the model is a set of data points $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{m}$, where m is the number of input-output pairs. The most common supervised learning tasks are *classification* and *regression*. Examples of supervised learning algorithms include linear regression, logistic regression, decision trees, support vector machines, and many more. The vast majority of the attack papers thus far are focused in supervised learning using deep neural networks.

2.1.2 Unsupervised Learning. In unsupervised learning, there are no labels $\mathbf{y}$. The training set $\mathcal{D}$ consists only of the inputs $\mathbf{x}_i$. Unsupervised algorithms aim to find structure or patterns in the data without having access to labels. Usual tasks in unsupervised learning are *clustering feature learning*, *anomaly detection*, and *dimensionality reduction*. In the context of this survey, attacks on unsupervised learning appear mostly as attacks on language models.

2.1.3 Reinforcement Learning. Reinforcement learning concerns itself with agents that make observations of the environment and use these to take actions with the goal of maximizing a reward signal. In the most general formulation, the set of actions is not predefined and the rewards are not necessarily immediate but can occur after a sequence of actions [123]. To our knowledge, no privacy-related attacks against reinforcement learning have been reported, but it has been used to launch other privacy-related attacks [98].

2.1.4 Semi-supervised Learning. In many real-world settings, the amount of labeled data can be significantly smaller than that of unlabeled ones, and it might be too costly to obtain high-quality labels. Semi-supervised learning algorithms aim to use unlabeled data to learn higher level representations and then use the labeled examples to guide the downstream learning task. An example of semi-supervised learning would be to use an unsupervised learning technique such as clustering on unlabeled data and then use a classifier to separate representative training data from each cluster. Other notable examples are generative models such as **Generative Adversarial Networks (GANs)** [34].

2.1.5 Generative and Discriminative Learning. Another categorization of learning algorithms is that of *discriminative* vs *generative* algorithms. Discriminative classifiers try to model the conditional probability $p(y|\mathbf{x})$, i.e., they try to learn the decision boundaries that separate the different classes directly based on the input data $\mathbf{x}$. Examples of such algorithms are logistic regression and neural networks. Generative classifiers try to capture the joint distribution $p(\mathbf{x}, y)$. An example of such a classifier is Naive Bayes. Usually, generative models that do not require labels, but they try to model $p(\mathbf{x})$, explicitly or implicitly. Notable examples are language models that predict the next word(s) given some input text or GANs and **Variational Autoencoders (VAEs)** [62] that are able to generate data samples that match the properties of the training data.

2.2 Learning Architectures

From a system architecture point of view, we view the learning process as either a centralized or a distributed one. The main criterion behind this categorization is whether the data and the model are collocated or not.

2.2.1 Centralized Learning. In a centralized learning setting, the data and the model are collocated. There can be one or multiple more data producers or owners, but all data are gathered in one central place and used for the training of the model. The location of the data can be in a single or even multiple machines in the same data center. While using parallelism in the form of multiple GPUs and CPUs could be considered a distributed learning mode, it is not for us, since we use the model and data collocation as the main criterion for the distinction between centralized and distributed learning. The centralized learning architecture includes the **Machine Learning as a Service (MLaaS)** setup, where the data owner uploads their data to a cloud-based service that is tasked with creating the best possible model.

2.2.2 Distributed Learning. The requirements that drive the need for distributed learning architectures are the handling and processing of large amounts of data, the need for computing and memory capacity, and even privacy concerns. From the existing variants of distributed learning, we present those that are relevant from a privacy perspective, namely, *collaborative* or ***federated learning*** **(FL)**, *fully decentralized* or ***peer-to-peer*** **(P2P)** learning, and *split learning*.

Collaborative or federated learning is a form of decentralized training where the goal is to learn one global model from data stored in multiple remote devices or locations [67]. The main idea is that the data do not leave the remote devices. Data are processed locally and they used to update the local models. Intermediate model updates are sent to the central server that aggregates them and creates a global model. The central server then sends the global model back to all participant devices.

In fully decentralized learning or P2P learning, there is no central orchestration server. Instead, the devices communicate in a P2P fashion and exchange their updates directly with other devices. This setup may be interesting from a privacy perspective, since it alleviates the need to trust a central server. However, attacks on P2P systems are relevant in such settings and need to be taken into account. Up to now, there were no privacy-based attacks reported on such systems; although they may become relevant in the future. Moreover, depending on the type of information shared between the peers, several of the attacks on collaborative learning may be applicable.

In split learning, the trained model is split into two or more parts. The edge devices keep the initial layers of the deep learning model and the centralized server keeps the final layers [38, 59]. The reason for the split is mainly to lower the communication costs by sending intermediate model outputs instead of the input data. This setup is also relevant in situations where remote or edge devices have limited resources and are connected to a central cloud server. This latter scenario is common for **Internet of Things (IoT)** devices.

2.3 Training and Inference

Training of supervised ML models usually follows the **Empirical Risk Minimization (ERM)** approach [126], where the objective is to find the parameters θ^* that minimize the *risk* or *objective function*, which is calculated as an average over the training dataset:

$$\mathcal{J}(\mathcal{D};\theta) = \frac{1}{m}\sum_{i=1}^{m} l(f(x_i;\theta), y_i), \tag{1}$$

where $l(\cdot)$ is a loss function, e.g., cross entropy loss, and m is the number of data points in the dataset $\mathcal{D}$.

The idea behind ERM is that the training dataset is a subset drawn from the unknown true data distribution for the learning task. Since we have no knowledge of the true data distribution, we cannot minimize the true objective function, but instead we can minimize the estimated objective over the data samples that we have. In some cases, a regularization term is added to the objective function to reduce overfitting and stabilize the training process.

2.3.1 Training in Centralized Settings. The training process usually involves an iterative optimization algorithm such as gradient descent [13], which aims to minimize the objective function by following the path induced by its gradients. When the dataset is large, as is often the case with deep neural networks, taking one gradient step becomes too costly. In that case, variants of gradient descent that involve steps taken over smaller batches of data, are preferred. One such optimization method is called **Stochastic Gradient Descent (SGD)** [107] defined by

$$\theta_{t+1} = \theta_t - \eta\mathbf{g}, \tag{2}$$

$$\mathbf{g} = \frac{1}{m'}\nabla_\theta \sum_{i=1}^{m'} l(f(\mathbf{x}_i;\theta), \mathbf{y}_i), \tag{3}$$

where η is the learning rate and $\mathbf{g}$ is the gradient of the loss function with respect to parameters θ. In the original formulation of SGD the gradient $\mathbf{g}$ is calculated over a single data point from $\mathcal{D}$, chosen randomly, hence the name stochastic. In practice, it is common to use mini-batches of size m' where $m' < m$, instead of a single data point to calculate the loss gradient at each step (Equation (3)). Mini-batches lower the variance of the stochastic gradient estimate, but the size m' is a tunable parameter that can affect the performance of the algorithm. While SGD is still quite popular, several improvements have been proposed to try to speed up convergence by adding momentum [105], by using adaptive learning rates as, for example, in the RMSprop algorithm [45], or by combining both improvements as in the Adam algorithm [61].

2.3.2 Training in Distributed Settings. The most popular learning algorithm for federated learning is federated averaging [82], where each remote device calculates one step of gradient descent from the locally stored data and then shares the updated model weights with the parameter server.

The parameter server averages the weights of all remote participants and updates the global model, which is subsequently shared again with the remote devices. It can be defined as

$$\theta_{t+1} = \frac{1}{K} \sum_{k=1}^{K} \theta_t^{(k)}, \tag{4}$$

where K is the number of remote participants and the parameters $\theta_t^{(k)}$ of participant k have been calculated locally based on Equations (2) and (3).

Another approach that comes from the area of distributed computing is downpour (or synchronized) SGD [22], which proposes to share the loss gradients of the distributed devices with the parameter server that aggregates them and then performs one step of gradient descent. It can be defined as

$$\theta_{t+1} = \theta_t - \eta \sum_{k=1}^{K} \frac{m^{(k)}}{M} \mathbf{g}_t^{(k)}, \tag{5}$$

where $\mathbf{g}_t^{(k)}$ is the gradient computed by participant k based on Equation (3) using their local data, $m^{(k)}$ is the number of data points in the remote participant and M is the total number of data points in the training data. After the calculation of Equation (5), the parameter server sends the updated model parameters θ_{t+1} to the remote participants.

2.3.3 Inference. Once the models are trained, they can be used to make inferences or predictions over previously unseen data. At this stage, the assumption is that the model parameters are fixed, although the models are usually monitored, evaluated, and retrained if necessary. The majority of the attacks in this survey are attacks during the inference phase of the model lifecycle except for the attacks on collaborative learning, which are usually performed during training.

3 THREAT MODEL

To understand and defend against attacks in machine learning from a privacy perspective, it is useful to have a general model of the environment, the different actors, and the assets to protect.

From a threat model perspective, the assets that are sensitive and are potentially under attack are the training dataset $\mathcal{D}$, the model itself, its parameters θ, its hyper-parameters, and its architecture. The actors identified in this threat model are:

(1) The **data owners**, whose data may be sensitive.
(2) The **model owners**, which may or may not own the data and may or may not want to share information about their models.
(3) The **model consumers**, that use the services that the model owner exposes, usually via some sort of programming or user interface.
(4) The **adversaries**, that may also have access to the model's interfaces as a normal consumer does. If the model owner allows, then they may have access to the model itself.

Figure 1 depicts the assets and the identified actors under the threat model, as well as the information flow and possible actions. This threat model is a logical model and it does not preclude the possibility that some of these assets may be collocated or spread in multiple locations.

Distributed modes of learning, such as federated or collaborative learning, introduce different spatial models of adversaries. In a federated learning setting, the adversary can be collocated with the global model, but it can also be a local attacker. Figure 2 shows the threat model in a collaborative learning setting. The presence of multiple actors allows also the possibility of *colluding* adversaries that join forces.

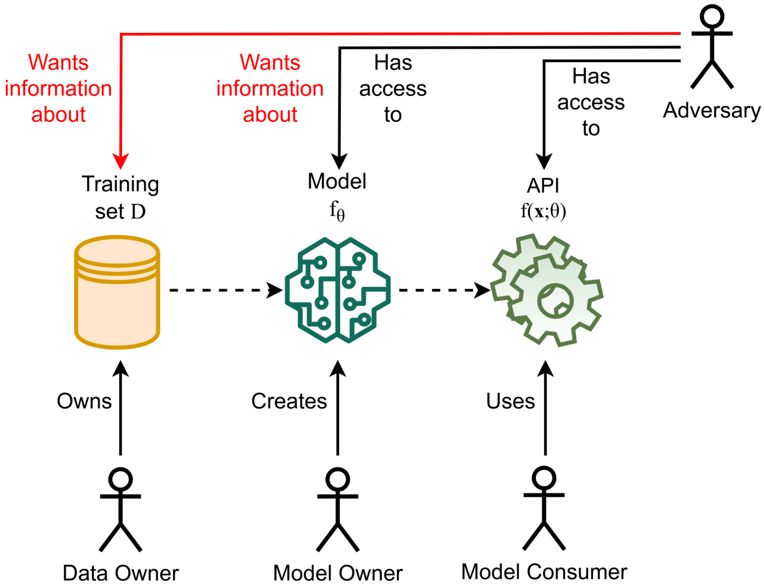


Fig. 1. Threat Model of privacy and confidentiality attacks against machine learning systems. The human figure represents actors and the symbols represent the assets. Dashed lines represent data and information flow, while full lines represent possible actions. In red are the actions of the adversaries, available under the threat model.

The different attack surfaces against machine learning models can be modelled in terms of **adversarial knowledge**. The range of knowledge varies from limited, e.g., having access to a machine learning API, to having knowledge of the full model parameters and training settings. In between these two extremes, there is a range of possibilities such as partial knowledge of the model architecture, its hyper-parameters, or training setup. The knowledge of the adversary can also be considered from a dataset point of view. In the majority of the papers reviewed, the authors assume that the adversaries have no knowledge of the training data samples, but they may have some knowledge of the underlying data distribution.

From a taxonomy point of view, attacks where the adversary has no knowledge of the model parameters, architecture, or training data are called **black-box** attacks. An example of a black-box system is MLaaS, where the users usually provide some input and receive either a prediction vector or a class label from a pre-trained model hosted in the cloud. Most black-box papers assume the existence of a prediction vector. In a similar fashion, **white-box** attacks are those where the adversary has either complete access to the target model parameters or their loss gradients during training. This is the case, for example, in most distributed modes of training. In between the two extremes, there are also attacks that make stronger assumptions than the black-box ones, but do not assume full access to the model parameters. We refer to these attacks as **partial white-box** attacks. It is important to add here that the majority of the works assume full knowledge of the expected input, although some form of preprocessing might be required.

The time of the attack is another parameter to consider from a taxonomy point of view. The majority of the research in the area is dealing with attacks during **inference**; however, most collaborative learning attacks assume access to the model parameters or gradients during **training**. Attacks during the training phase of the model open up the possibility for different types of adversarial behavior. A **passive** or *honest-but-curious* attacker does not interfere with the

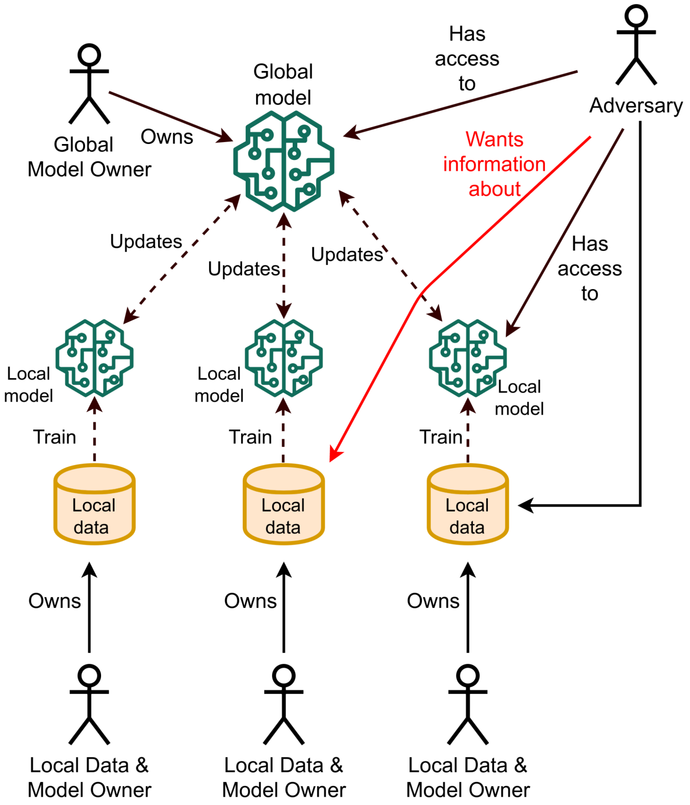


Fig. 2. Threat model in a collaborative learning setting. Dashed lines represent data and information flows, while full lines represent possible actions. In red are the actions of the adversaries, available under the threat model. In this setting the adversary can be placed either at the parameter server or locally. Model consumers are not depicted for reasons of simplicity. In federated learning, local model owners can be also model consumers.

training process and they are only trying to infer knowledge during or after the training. If the adversary interferes with the training in any way, then they are considered an **active** attacker.

Finally, since the interest of this survey is in privacy attacks based on unintentional information leakage regarding the data or the machine learning model, there is no coverage of *security-based* attacks, such as model poisoning or evasion attacks, or attacks against the infrastructure that hosts the data, models or provided services.

4 ATTACK TYPES

In privacy-related attacks, the goal of an adversary is to gain knowledge that was not intended to be shared. Such knowledge can be related to the training data $\mathcal{D}$ or the model f, or even to properties of the data such as unintentionally encoded biases. In our taxonomy, the privacy attacks studied are categorized into four types: **membership inference**, **reconstruction**, **property inference**, and **model extraction**.

4.1 Membership Inference Attacks

Membership inference tries to determine whether an input sample $\mathbf{x}$ was used as part of the training set $\mathcal{D}$. This is the most popular category of attacks and was first introduced by

Shokri et al. [116]. The attack only assumes knowledge of the model's output prediction vector (black-box) and was carried out against supervised machine learning models. White-box attacks in this category are also a threat, especially in a collaborative setting, where an adversary can mount both passive and active attacks. If there is access to the model parameters and gradients, then this allows for more effective white-box membership inference attacks in terms of accuracy [90].

Apart from supervised models, generative models such as GANs and VAEs are also susceptible to membership inference attacks [16, 39, 44]. The goal of the attack, in this case, is to retrieve information about the training data using varying degrees of knowledge of the data generating components.

Finally, these types of attacks can be viewed from a different perspective, that of the data owner. In such a scenario, the owner of the data may have the ability to audit black-box models to see if the data have been used without authorization [46, 118].

4.2 Reconstruction Attacks

Reconstruction attacks try to recreate one or more training samples and/or their respective training labels. The reconstruction can be partial or full. Previous works have also used the terms **attribute inference** or **model inversion** to describe attacks that, given output labels and partial knowledge of some features, try to recover sensitive features or the full data sample. For the purpose of this survey, all these attacks are considered as part of the larger set of reconstruction attacks. The term **attribute inference** has been used in other parts of the privacy related literature to describe attacks that infer sensitive "attributes" of a targeted user by leveraging publicly accessible data [31, 53]. These attacks are not part of this review as they are mounted against the individual's data directly and not against ML models.

A major distinction between the works of this category is between those that create an actual reconstruction of the data [41, 134, 138, 145, 148] and the ones that create class representatives or probable values of sensitive features that do not necessarily belong to the training dataset [28, 43, 47, 109, 138]. In classification models, the latter case is limited to scenarios where classes are made up of one type of object, e.g., faces of the same person. While this limits the applicability of the attack, it can still be an interesting scenario in some cases.

4.3 Property Inference Attacks

The ability to extract dataset properties that were not explicitly encoded as features or were not correlated to the learning task is called **property inference**. An example of property inference is the extraction of information about the ratio of women and men in a patient dataset when this information was not an encoded attribute or a label of the dataset. Or having a neural network that performs gender classification and can be used to infer if people in the training dataset wear glasses or not. In some settings, this type of leak can have privacy implications. These types of properties can also be used to get more insight about the training data, which can lead to adversaries using this information to create similar models [3] or even have security implications when the learned property can be used to detect vulnerabilities of a system [29].

Property inference aims to extract information that was learned from the model unintentionally and that is not related to the training task. Even well generalized models may learn properties that are relevant to the whole input data distribution and sometimes this is unavoidable or even necessary for the learning process. What is more interesting from an adversarial perspective, are properties that may be inferred from a specific subset of training data, or eventually about a specific individual.

Property inference attacks so far target either dataset-wide properties [3, 29, 78, 117] or the emergence of properties within a batch of data [84]. The latter attack was performed on the collaborative training of a model.

4.4 Model Extraction Attacks

Model extraction is a class of black-box attacks where the adversary tries to extract information and potentially fully reconstruct a model by creating a substitute model $\hat{f}$ that behaves very similarly to the model under attack f. There are two main goals for the substitute models. First, to create models that match the accuracy of the target model f in a test set that is drawn from the input data distribution and related to the learning task [63, 86, 98, 124]. Second, to create a substitute model $\hat{f}$ that matches f at a set of input points that are not necessarily related to the learning task [21, 50, 56, 124]. Jagielski et al. [50] referred to the former attack as **task accuracy** extraction and to the latter as **fidelity** extraction. In task accuracy extraction, the adversary is interested in creating a substitute that learns the same task as the target model equally well or better. In the latter case, the adversary aims to create a substitute that replicates the decision boundary of f as faithfully as possible. This type of attack can be later used as a stepping stone before launching other types of attacks such as adversarial attacks [56, 103] or membership inference attacks [90]. In both cases, it is assumed that the adversary wants to be as efficient as possible, i.e., to use as few queries as possible. Knowledge of the target model architecture is assumed in some works, but it is not strictly necessary if the adversary selects a substitute model that has the same or higher complexity than the model under attack [56, 63, 98].

Apart from creating substitute models, there are also approaches that focus on recovering information from the target model, such as hyper-parameters in the objective function [131] or information about various neural network architectural properties such as activation types, optimisation algorithm, number of layers, and so on [97].

5 CAUSES OF PRIVACY LEAKS

The conditions under which machine learning models leak is a research topic that has started to emerge in the past few years. Some models leak information due to the way they are constructed. An example of such a case is **Support Vector Machines (SVMs)**, where the support vectors are data points from the training dataset. Other models, such as linear classifiers are relatively easy to “reverse engineer” and to retrieve their parameters just by having enough input/output data pairs [124]. Larger models such as deep neural networks usually have a large number of parameters and simple attacks are not feasible. However, under certain assumptions and conditions, it is possible to retrieve information about either the training data or the models themselves.

5.1 Causes of Membership Inference Attacks

One of the conditions that has been shown to improve the accuracy of membership inference is the poor generalization of the model. The connection between overfitting and black-box membership inference was initially investigated by Shokri et al. [116]. This paper was the first to examine membership inference attacks on neural networks. The authors measured the effect of overfitting on the attack accuracy by training models in different MLaaS platforms using the same dataset. The authors showed experimentally that overfitting can lead to privacy leakage but also noted that it is not the only condition, since some models that had lower generalization error where more prone to membership leaks. The effect of overfitting was later corroborated formally by Yeom et al. [140]. The authors defined membership advantage as a measure of how well an attacker can distinguish whether a data sample belongs to the training set or not, given access to the model.

They proved that the membership advantage is proportional to the generalization error of the model and that overfitting is a sufficient condition for performing membership inference attacks, but not a necessary one. Additionally, Long et al. [74] showed that even in well-generalized models, it is possible to perform membership inference for a subset of the training data, which they named *vulnerable records*. This was also corrobaorated in Reference [11]. In addition, Carlini et al. showed that larger language models are more prone to memorization than smaller models [12].

Other factors, such as the model architecture, model type, and dataset structure, affect the attack accuracy. Similar to Reference [116] but in the white-box setting, Nasr et al. [90] showed that two models with the same generalization error resulted to different degrees of leakage. More specifically, the most complex model in terms of number of parameters exhibited higher attack accuracy, showing that model complexity is also an important factor. In a white-box setting also, Leino and Fredrikson [66] showed that the learned features of the target model capture differences between the training data distribution and the general population. These differences can be used to distinguish members of the training set even when the target model generalizes well.

Truex et al. [125] ran different types of experiments to measure the significance of the model type as well as the the number of classes present in the dataset. They found that certain model types such as Naive Bayes are less susceptible to membership inference attacks than decision trees or neural networks. They also showed that as the number of classes in the dataset increases, so does the potential of membership leaks. This finding agrees with the results in Reference [116].

Securing machine learning models against adversarial attacks can also have an adverse effect on the model's privacy as shown by Song et al. [120]. Current state-of-the-art proposals for robust model training, such as **projective gradient descent (PGD)** adversarial training [77], increase the model's susceptibility to membership inference attacks. This is not unexpected, since robust training methods (both empirical and provable defenses) tend to increase the generalization error. As previously discussed, the generalization error is related to the success of the attack. Furthermore, the authors of Reference [120] argue that robust training may lead to increased model sensitivity to the training data, which can also affect membership inference.

The generalization error is easily measurable in supervised learning under the assumption that the test data can capture the nuances of the real data distribution. In generative models and specifically in GANs this is not the case, hence the notion of overfitting is not directly applicable. All three papers that deal with membership inference attacks against GANs mention overfitting as an important factor behind successful attacks [16, 39, 44]. In this case, overfitting means that the generator has memorized and replays part of the training data. This is further corroborated in the study in Reference [16], where their attacks are shown to be less successful as the training data size increases.

5.2 Causes of Reconstruction Attacks

Regarding reconstruction attacks, Yeom et al. [140] showed that a higher generalization error can lead to a higher probability to infer data attributes but also that the influence of the target feature on the model is an important factor. However, the authors assumed that the adversary has knowledge of the prior distribution of the target features and labels. Using weaker assumptions about the adversary's knowledge, Zhang et al. [145] showed theoretically and experimentally that a model that has high predictive power is more susceptible to reconstruction attacks.

5.3 Causes of Property Inference Attacks

Property inference is possible even with well-generalized models [29, 84] so overfitting does not seem to be a cause of property inference attacks. Unfortunately, regarding property inference attacks, we have less information about what makes them possible and under which circumstances they appear to be effective. This is an interesting avenue for future research, both from a theoretical and an empirical point of view.

5.4 Causes of Model Extraction

While overfitting increases the success of black-box membership inference attacks, the exact opposite holds for model extraction attacks. It is possible to steal model parameters when the models under attack have 98% or higher accuracy in the test set [97]. Also models with a higher generalization error are harder to steal, probably due to the fact that they may have memorized samples that are not part of the attacker's dataset [73]. Another factor that may affect model extraction success is the dataset used for training. Higher number of classes may lead to worse attack performance [73].

6 IMPLEMENTATION OF THE ATTACKS

More than 45 papers were analyzed in relation to privacy attacks against machine learning. This section describes in some detail the most commonly used techniques as well as the essential differences between them. The papers are discussed in two sections: attacks on centralized learning and attacks on distributed learning.

6.1 Attacks Against Centralized Learning

In the centralized learning setting, the main assumption is that models and data are collocated during the training phase. The next subsection introduces a common design approach that is used by multiple papers, namely, the use of *shadow models* or *shadow training*. The rest of the subsections are dedicated to the different attack types and introduce the assumptions, common elements, as well as differences of the reviewed papers.

6.1.1 Shadow Training. A common design pattern for a lot of supervised learning attacks is the use of **shadow models** and **meta-models** or **attack-models** [3, 18, 29, 40, 46, 51, 68, 97, 106, 108, 110, 116, 125]. The general shadow training architecture is depicted in Figure 3. The main intuition behind this design is that models behave differently when they see data that do not belong to the training dataset. This difference is captured in the model outputs as well as in their internal representations. In most designs there is a target model and a target dataset. The adversary is trying to infer either membership or properties of the training data. They train a number of shadow models using shadow datasets $\mathcal{D}_{shadow} = \{\mathbf{x}_{shadow,i}, \mathbf{y}_{shadow,i}\}_{i=1}^{n}$ that usually are assumed to come from the same distribution as the target dataset. After the shadow models' training, the adversary constructs an attack dataset $\mathcal{D}_{attack} = \{f_i(\mathbf{x}_{shadow,i}), \mathbf{y}_{shadow,i}\}_{i=1}^{n}$, where f_i is the respective shadow model. The attack dataset is used to train the meta-model, which essentially performs inference based on the outputs of the shadow models. Once the meta-model is trained, it is used for testing using the outputs of the target model.

6.1.2 Membership Inference Attacks. In *membership inference* black-box attacks, the most common attack pattern is the use of shadow models. The output of the shadow models is a prediction vector [40, 51, 106, 110, 116, 125] or only a label [68]. The labels used for the attack dataset come from the test and training splits of the shadow data, where the data points that belong to the test set are labeled as non-members of the training set. The meta-model is trained to recognize patterns in the prediction vector output of the target model. These patterns allow the meta-model to infer whether a data point belongs to the training dataset or not. The number of shadow models affects the attack accuracy, but it also incurs cost to the attackers. Salem et al. [110] showed that membership inference attacks are possible with as little as one shadow model.

Shadow training can be further reduced to a threshold-based attack, where instead of training a meta-model, one can calculate a suitable threshold function that indicates whether a sample is

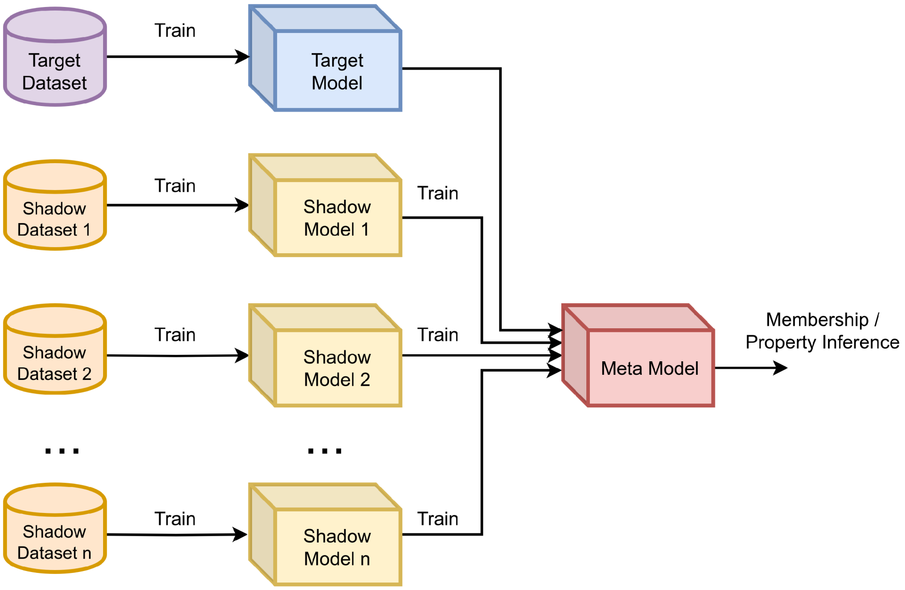


Fig. 3. Shadow training architecture. At first, a number of shadow models are trained with their respective shadow datasets to emulate the behavior of the target model. At the second stage, a meta-model is being trained from the outputs of the shadow models and the known labels of the shadow datasets. The meta-model is used to infer membership or properties of data or the model given the output of the target model.

a member of the training set. The threshold can be learned from multiple shadow models [108] or even without using any shadow models [140]. Sablayrolles et al. [108] showed that a Bayes-optimal membership inference attack depends only on the loss and their attack outperforms previous attacks such as References [116, 140]. In terms of attack accuracy, they reported up to 90.8% on large neural network models such as VGG16 [71] that were performing classification on the Imagenet [23] dataset.

A Bayes-optimal attack was also proposed in the white-box scenario and for linear models [66]. While the optimal attack required strong assumptions regarding the target data distribution and the attacker knowledge, further relaxations make it feasible even for deep neural network targets. The attack on linear models requires the training of an identical proxy model that is used to calculate differences from the white-box model's weights and subsequently use them for membership inference. For the deep neural network, each layer is replaced by a local linear approximation, which can be used for the attack in a similar manner.

In addition to relaxations on the number of shadow models, attacks have been shown to be data driven, i.e., an attack can be successful even if the target model is different than the shadow and meta-models [125]. The authors tested several types of models such as k-NN, logistic regression, decision trees and naive Bayes classifiers in different combinations on the role of the target model, shadow and meta model. The results showed that (i) using different types of models did not affect the attack accuracy and (ii) in most cases, models such as decision trees outperformed neural networks in terms of attack accuracy and precision.

Shadow model training requires a shadow dataset. One of the main assumptions of membership inference attacks on supervised learning models is that the adversary has no or limited knowledge of the training samples used. However, the adversary knows something about the underlying data distribution of the target's training data. If the adversary does not have access to a suitable dataset, then they can try to generate one [116, 125]. Access to statistics about the probability distribution of several features allows an attacker to create the shadow dataset using sampling techniques. If a statistics-based generation is not possible, then a query-based approach using the target models' prediction vectors is another possibility. Generating auxiliary data using GANs was also proposed by Hayes et al. [39]. If the adversary manages to find input data that generate predictions with high confidence, then no prior knowledge of the data distribution is required for a successful attack [116]. Salem et al. [110] went so far as to show that it is not even necessary to train the shadow models using data from the same distribution as the target, making the attack more realistic, since it does not assume any knowledge of the training data.

The previous discussion is mostly relevant to supervised classification or regression tasks. The efficacy of membership inference attacks against sequence-to-sequence model for machine translation, was studied by Reference [46]. The authors used shadow models that try to mimic the target model's behavior and then used a meta-model to infer membership. They found that sequence generation models are much harder to attack compared to those trained for other tasks such as image classification. However, membership of *out-of-domain* and out-of-vocabulary data was easier to infer.

Membership inference attacks are also applicable to deep generative models such as GANs and VAEs [16, 39, 44]. Since these models have more than one component (generator/discriminator, encoder/decoder), adversarial threat modeling needs to take that into account. For these types of models, the taxonomy proposed by Chen et al. [16] is partially followed. We consider black-box access to the generator as the ability to access generated samples and partial black-box access, the ability to provide inputs in the latent space z and generate samples. Having access to the generator model and its parameters is considered a white-box attack. The ability to query the discriminator is also a white-box attack.

The full white-box attacks with access to the GAN discriminator are based on the assumption that if the GAN has "overfitted," then the data points used for its training will receive higher confidence values as output by the discriminator [39]. In addition to the previous attack, Hayes et al. [39] proposed a set of attacks in the partial black-box setting. These attacks are applicable to both GANs and VAEs or any generative model. If the adversary has no auxiliary data, then they can attempt to train an auxiliary GAN whose discriminator distinguishes between the data generated by the target generator and the data generated by the auxiliary GAN. Once the auxiliary GAN is trained, its discriminator can be used for the white-box attack. The authors considered also scenarios where the adversary may have auxiliary information such as knowledge of the target's training and test data. Using the auxiliary data, they can train another GAN whose discriminator would be able to distinguish between members of the original training set and non-members.

A distance-based attack over the nearest neighbors of a data point was proposed by Chen et al. [16] for the full black-box model. In this case, a data point $\mathbf{x}$ is a member of the training set if within its k-nearest neighbors there is at least one point that has a distance lower than a threshold ϵ. The authors proposed more complex attacks as the level of knowledge of the adversary increases, based on the idea that the reconstruction error between the real data point x and a sample generated by the generator given some input z should be smaller if the data point is coming from the training set.

6.1.3 Reconstruction Attacks. The initial reconstruction attacks were based on the assumption that the adversary has access to the model f, the priors of the sensitive and nonsensitive features, and the output of the model for a specific input x. The attack was based on estimating the values of sensitive features, given the values of nonsensitive features and the output label [28]. This method used a **maximum *a posteriori* (MAP)** estimate of the attribute that maximizes the probability of observing the known parameters. Hidano et al. [43] used a similar attack but they made no assumption about the knowledge of the nonsensitive attributes. For their attack to work, they assumed that the adversary can perform a *model poisoning* attack during training.

Both previous attacks worked against linear regression models, but as the number of features and their range increases, the attack feasibility decreases. To overcome the limitations of the MAP attack, Fredrikson et al. [27] proposed another inversion attack, which recovers features using target labels and optional auxiliary information. The attack was formulated as an optimization problem where the objective function is based on the observed model output and uses gradient descent in the input space to recover the input data point. The method was tested on image reconstruction. The result was a class representative image, which in some cases was quite blurry even after denoising. A formalization of the model inversion attacks in References [27, 28] was later proposed by Wu et al. [135].

Since the optimization problem in Reference [27] is quite hard to solve, Zhang et al. [145] proposed to use a GAN to learn some auxiliary information of the training data and produce better results. The auxiliary information in this case is the presence of blurring or masks in the input images. The attack first uses the GAN to learn to generate realistic looking images from masked or blurry images using public data. The second step is a GAN inversion that calculates the latent vector $\hat{z}$, which generates the most likely image:

$$\hat{z} = \arg\min_{z} L_{prior}(z) + \lambda L_{id}(z), \tag{6}$$

where the prior loss L_{prior} is ensuring the generation of realistic images and L_{id} ensures that the images have a high likelihood in the target network. The attack is quite successful, especially on masked images.

Black-box only reconstruction attacks are less common, since the attacker has substantially less information. Nevertheless, Salem et al. [109] proposed reconstruction attacks in an online setting, where they used the prediction vectors of a holdout dataset before and after a training round, in combination with generative models to reconstruct labels and data samples. Finally, Yang et al. [138], proposed a black-box attack that employs an additional classifier that performs an inversion from the output of the target model $f(x)$ to a candidate output $\hat{x}$. The setup is similar to that of an autoencoder, only in this case the target network that plays the role of the encoder is a black box and it is not trainable. The attack was tested on different types of target model outputs: the full prediction vector, a truncated vector, and the target label only. When the full prediction vector is available, the attack performs a good reconstruction, but with less available information, the produced data point looks more like a class representative.

6.1.4 Property Inference Attacks. In most *property inference* the shadow datasets are labeled based on the properties that the adversary wants to infer, so the adversary needs access to data that have the property and data that do not have it. The meta-model is then trained to infer differences in the output vectors of the data that have the property versus the ones that do not. In white-box attacks, the meta-model input can be other feature representations such as the support vectors of an SVM [3] or transformations of neural network layer outputs [29]. When attacking language model embeddings, the embedding vectors themselves can be used to train a classifier to distinguish between properties such as text authorship [117]. Similarly to language model embeddings, graph embeddings and their properties are used to train an attack classifier that can be used to infer properties of the target graph [146]. Finally, the first poisoning attack used for the inference of dataset properties was proposed in Reference [78]. The attacker first poisons the training data to introduce a correlation between the property in question and the target label and then selects query samples that allows them to decise whether the frequency of the property in the training dataset.

6.1.5 Model Extraction Attacks. When the adversary has access to the inputs and prediction outputs of a model, it is possible to view these pairs of inputs and outputs as a system of equations, where the unknowns are the model parameters [124] or hyper-parameters of the objective function [131]. In the case of a linear binary classifier, the system of equations is linear and only $d + 1$ queries are necessary to retrieve the model parameters, where d is the dimension of the parameter vector θ. In more complex cases, such as multi-class linear regression or multi-layer perceptrons, the systems of equations are no longer linear. Optimization techniques such as **Broyden–Fletcher–Goldfarb–Shanno (BFGS)** [96] or stochastic gradient descent are then used to approximate the model parameters [124].

Lack of prediction vectors or a high number of model parameters renders equation solving attacks inefficient. A strategy is required to select the inputs that will provide the most useful information for model extraction. From this perspective, model extraction is quite similar to *active learning* [15]. Active learning makes use of an external oracle that provides labels to input queries. The oracle can be a human expert or a system. The labels are then used to train or update the model. In the case of model extraction, the target model plays the role of the oracle.

Following the active learning approach, several papers propose an adaptive training strategy. They start with some initial data points or *seeds*, which they use to query the target model and retrieve labels or prediction vectors, which they use to train the substitute model $\hat{f}$. For a number of subsequent rounds, they extend their dataset with new synthetic data points based on some adaptive strategy that allows them to find points close to the decision boundary of the target model [15, 56, 100, 103, 124, 142]. Chandrasekaran et al. [15] provided a more query efficient method of extracting nonlinear models such as kernel SVMs, with slightly lower accuracy than the method proposed by Tramer et al. [124], while the opposite was true for Decision Tree models. ActiveThief [100] and CloudLeak [142] are attacks that are based on the combination of active learning and adversarial examples for the extraction of deep neural network models. Both attacks were also combined with other techniques such as transfer learning or k-center [112] to optimize their performance. One of the main differences between the two approaches is that the CloudLeak attack uses adversarial samples to query the target, while ActiveThief uses adversarial samples as a way to find samples from the training dataset that are closed to the decision boundary of the substitute model, hence data with higher uncertainty.

Several other strategies for selecting the most suitable data for querying the target model use: (i) data that are not synthetic but belong to different domains such as images from different datasets [6, 21, 98], (ii) semi-supervised learning techniques such as rotation loss [143] or MixMatch [7] to augment the dataset [50], (iii) data generated through model inversion techniques [32], or iv) randomly generated input data [56, 63, 124]. In terms of efficiency, semi-supervised methods such as MixMatch require much fewer queries than fully supervised extraction methods to perform similarly or better in terms of task accuracy and fidelity, against models trained for classification using CIFAR-10 and SVHN datasets [50]. For larger models, trained for Imagenet classification, even querying a 10% of the Imagenet data, gives a comparable performance to the target model [50]. Against a deployed MLaaS service that provides facial characteristics, Orekondy et al. [98] managed to create a substitute model that performs at 80% of the target in task accuracy, spending as little as $30.

Some, mostly theoretical, work has demonstrated the ability to perform direct model extraction beyond linear models [50, 86]. Full model extraction was shown to be theoretically possible against two-layer fully connected neural networks with **rectified linear unit (ReLU)** activations by Milli et al. [86]. However, their assumption was that the attacker has access to the loss gradients with respect to the inputs. Jagielski et al. [50] managed to do a full extraction of a similar network without the need of gradients. Both approaches take into account that ReLUs transforms the neural network into a piecewise linear function of the inputs. By probing the model with different inputs, it is possible to identify where the linearity breaks and use this knowledge to calculate the network parameters. In a hybrid approach that uses both a learning strategy and direct extraction, Jagielski et al. [50], showed that they can extract a model trained on MNIST with almost 100% fidelity by using an average of $2^{19.2}$ to $2^{22.2}$ queries against models that contain up to 400,000 parameters. However, this attack assumes access to the loss gradients similar to Reference [86].

Finally, apart from learning substitute models directly, there is also the possibility of extracting model information such as architecture, optimization methods and hyper-parameters using shadow models [97]. The majority of attacks were performed against neural networks trained on MNIST. Using the shadow models' prediction vectors as input, the meta-models managed to learn to distinguish whether a model has certain architectural properties. An additional attack by the same authors, proposed to generate adversarial samples, which were created by models that have the property in question. The generated samples were created in a way that makes a classifier output a certain prediction if they have the attribute in question. The target model's prediction on this adversarial sample is then used to establish if the target model has a specific property. The combination of the two attacks proved to be the most effective approach. Some properties such as activation function, presence of dropout, and max-pooling were the most successfully predicted.

6.2 Attacks Against Distributed Learning

In the federated learning setting, multiple devices acquire access to the global model that is trained from data that belong to different end users. Furthermore, the parameter server has access to the model updates of each participant either in the form of model parameters or that of loss gradients. In split learning settings, the central server also gains access to the outputs of each participant's intermediate neural network layers. This type of information can be used to mount different types of attacks by actors that are either residing in a central position or even by individual participants. The following subsection presents the types of attacks in distributed settings, as well as their common elements, differences, and assumptions.

6.2.1 Membership Inference Attacks. Nasr et al. [89] showed that a white-box membership inference attack is more effective than the black-box one, under the assumption that the adversary has some auxiliary knowledge about the training data, i.e., has access to some data from the training dataset, either explicitly or because they are part of a larger set of data that the adversary possesses. The adversary can use the model parameters and the loss gradients as inputs to another model that is trained to distinguish between members and non-members. The white-box attack accuracy with various neural network architectures was up to 75.1%; however, all target models had a high generalization error.

In the active attack scenario, the attacker, which is also a local participant, alters the gradient updates to perform a gradient ascent instead of descent for the data whose membership is under question. If some other participant uses the data for training, then their local SGD will significantly reduce the gradient of the loss and the change will be reflected in the updated model, allowing the adversary to extract membership information. Attacks from a local active participant reached an attack accuracy of 76.3% and in general, the active attack accuracy was higher than that of the passive attack in all tested scenarios. However, increasing the number of participants has adverse effects on the attack accuracy, which drops significantly after five or more participants. A global active attacker that is in a more favourable position can isolate the model parameter updates they receive from each participant. Such an active attacker reached an attack accuracy of 92.1%.

6.2.2 Property Inference Attacks. Passive property inference requires access to some data that possess the property and some that do not. The attack applies to both federated average and synchronized SGD settings, where each remote participant receives parameter updates from the parameter server after each training round [84]. The initial dataset is of the form $\mathcal{D}' = \{(\mathbf{x}, \mathbf{y}, \mathbf{y}')\}$, where $\mathbf{x}$ and $\mathbf{y}$ are the data used for training the distributed model and $\mathbf{y}'$ are the property labels. Every time the local model is updated, the adversary calculates the loss gradients for two batches of data. One batch that has the property in question and one that does not. This allows the construction of a new dataset that consists of gradients and property labels $(\nabla L, \mathbf{y}')$. Once enough labeled data have been gathered, a second model, f', is trained to distinguish between loss gradients of data that have the property versus those that do not. This model is then used to infer whether subsequent model updates were made using data that have the property. The model updates are assumed to be done in batches of data. The attack reaches an attack **area under the curve (AUC)** score of 98% and becomes increasingly more successful as the number of epochs increases. Attack accuracy also increases as the fraction of data with the property in question also increases. However, as the number of participants in the distributed training increases, the attack performance decreases significantly.

6.2.3 Reconstruction Attacks. Some data reconstruction attacks in federated learning use generative models and specifically GANs [47, 134]. When the adversary is one of the participants, they can force the victims to release more information about the class they are interested in reconstructing [47]. This attack works as follows: The potential victim has data for a class "A" that the adversary wants to reconstruct. The adversary trains an additional GAN model. After each training round, the adversary uses the target model parameters for the GAN discriminator, whose purpose is to decide whether the input data come from the class "A" or are generated by the generator. The aim of the GAN is to create a generator that is able to generate faithful class "A" samples. In the next training step of the target model, the adversary generates some data using the GAN and labels them as class "B." This forces the target model to learn to discriminate between classes "A" and "B," which in turn improves the GAN training and its ability to generate class "A" representatives.

If the adversary has access to the central parameter server, then they have direct access to the model updates of each remote participant. This makes it possible to perform more successful reconstruction attacks [134]. In this case, the GAN discriminator is again using the shared model parameters and learns to distinguish between real and generated data, as well as the identity of the participant. Once the generator is trained, the reconstructed samples are created using an optimization method that minimizes the distance between the real model updates and the updates due to the generated data. Both GAN-based methods assume access to some auxiliary data that belong to the victims. However, the former method generates only class representatives.

In a synchronized SGD setting, an adversary with access to the parameter server has access to the loss gradients of each participant during training. Using the loss gradients is enough to produce a high quality reconstruction of the training data samples, especially when the batch size is small [148]. The attack uses a second "dummy" model. Starting with random dummy inputs x' and labels y', the adversary tries to match the dummy model's loss gradients $\nabla_\theta \mathcal{J}'$ to the participant's loss gradients $\nabla_\theta \mathcal{J}$. This gradient matching is formulated as an optimization task that seeks to find the optimal x' and y' that minimize the gradients' distance:

$$x^*, y^* = \arg\min_{x', y'} \|\nabla_\theta \mathcal{J}'(\mathcal{D}'; \theta) - \nabla_\theta \mathcal{J}(\mathcal{D}; \theta)\|^2. \tag{7}$$

The minimization problem in Equation (7) is solved using **limited memory BFGS (L-BFGS)** [69]. The size of the training batch is an important factor in the speed of convergence in this attack.

Data reconstruction attacks are also possible during the inference phase in the split learning scenario [41]. When the local nodes process new data, they perform inference on these initial layers and then send their outputs to the centralized server. In this attack, the adversary is placed in the centralized server and their goal is to try to reconstruct the data used for inference. He et al. [41] cover a range of scenarios: (i) white-box, where the adversary has access to the initial layers and uses them to reconstruct the images, (ii) black-box where the adversary has no knowledge of the initial layers but can query them and thus recreate the missing layers, and (iii) query-free where the adversary cannot query the remote participant and tries to create a substitute model that allows data reconstruction. The latter attack produces the worst results, as expected, since the adversary is the weakest. The split of the layers between the edge device and the centralized server is also affecting the quality of reconstruction. Fewer layers in the edge neural network allow for better reconstruction in the centralized server.

6.3 Summary of Attacks

To summarize the attacks proposed against machine learning privacy, Table 1 presents the 47 papers analyzed in terms of adversarial knowledge, model under attack, attack type, and timing of the attack.

In terms of model types, 86.8% of the papers dealt with attacks against neural networks, with decision trees and linear models being the second most popular models to attack at 10.6% (some papers covered attacks against multiple model types). The concept of neural networks groups together both shallow and deep models, as well as multiple architectures, such as convolutional neural networks, recurrent neural networks, while under SVMs, we group together both linear and nonlinear versions.

The most popular attack type is membership inference (41.5% of the papers) with reconstruction attacks the second most popular (30.2% of the papers) and model extraction in the third place (28.3%). The majority of the proposed attacks are performed during the inference phase (86.8%). Attacks during training are mainly on distributed forms of learning or related to poisoning. Black-box and white-box attacks were studied in 71.7% and 43.4% of the papers, respectively (some papers covered both settings). In the white-box category, we also include partial white-box attacks.

The focus on neural networks in the existing literature as well as the focus on supervised learning is also apparent in Figure 4. The figure depicts types of machine learning algorithms versus the types of attacks that have been studied so far based on the existing literature. The list of algorithms is indicative and not exhaustive, but it contains the most popular ones in terms of research and deployment in real-world systems. Algorithms such as random forests [10] or gradient boosting trees [17, 60] have received little to no focus and the same holds for whole areas of machine learning such as reinforcement learning.

Another dimension that is interesting to analyze is the types of learning tasks that have been the target of attacks so far. Figure 5 presents information about the number of papers in relation to the learning task and the attack type. By learning task, we refer to the task in which the target model is initially trained. As the figure clearly shows, the majority of the attacks are on models that were trained for classification tasks, both binary and multiclass. This is the case across all four attack types.

While there is a diverse set of reviewed papers, it is possible to discern some high-level patterns in the proposed attacking techniques. Figure 6 shows the number of papers in relation to the attacking technique and attack type. Most notably, 10 papers used shadow training mainly for membership and property inference attacks. Active learning was quite popular in model extraction attacks and was proposed by six papers. Generative models (mostly GANs) were used in five papers across all attack types and another three papers used gradient matching techniques. It should be noted here that the "Learning" technique includes a number of different approaches, spanning from using model parameters and gradients as inputs to classifiers [84, 89] to using input-output

Table 1. Summary of Papers on Privacy Attacks on Machine Learning Systems, Including Assumptions About Adversarial Knowledge (Black / White-box), the Type of Model(s) under Attack, the Attack Type, and the Timing of the Attack (During Training or During Inference)

Reference	Year	Knowledge		ML Algorithms							Attack Type				Timing	
		Black-box	White-box	Linear regression	Logistic regression	Decision Trees	SVM	HMM	Neural network	GAN / VAE	Membership Inference	Reconstruction	Property Inference	Model Extraction	Training	Inference
Fredrikson et al. [28]	2014		●	●							●				●	
Fredrikson et al. [27]	2015	●	●			●			●			●				●
Ateniese et al. [3]	2015		●				●	●					●			●
Tramer et al. [124]	2016	●	●		●	●	●		●					●		●
Wu et al. [135]	2016	●	●		●				●			●				●
Hidano et al. [43]	2017		●	●								●				●
Hitaj et al. [47]	2017		●						●			●			●	
Papernot et al. [103]	2017	●							●					●		●
Shokri et al. [116]	2017	●							●		●					●
Correia-Silva et al. [21]	2018	●							●					●		●
Ganju et al. [29]	2018		●						●				●			●
Oh et al. [97]	2018	●							●					●		●
Long et al. [74]	2018	●							●		●					●
Rahman et al. [106]	2018		●						●		●					●
Wang & Gong [131]	2018		●	●	●		●		●					●		●
Yeom et al. [140]	2018	●	○	●		●			●			●				●
Carlini et al. [11]	2019	●						●				●				●
Hayes et al. [39]	2019	●	●							●	●					●
He et al. [41]	2019	●	●						●			●				●
Hilprecht et al. [44]	2019	●								●	●					●
Jayaraman & Evans [51]	2019	●	●						●		●	●				●
Juuti et al. [56]	2019	●							●					●		●
Milli et al. [86]	2019	●							●					●		●
Nasr et al. [90]	2019		●						●		●				●	
Melis et al. [84]	2019		●						●		●		●		●	
Orekondy et al. [98]	2019	●							●					●		●
Sablayrolles et al. [108]	2019		○						●		●					●
Salem et al. [110]	2019	●							●		●					●
Song L. et al. [120]	2019	●							●		●					●
Truex, et al. [125]	2019	●			●	●			●		●					●
Wang et al. [134]	2019		●						●			●			●	
Yang et al. [138]	2019	●							●			●				●
Zhu et al. [148]	2019		●						●			●			●	
Barbalau et al. [6]	2020	●							●					●		●
Chandrasekaran et al. [15]	2020	●				●	●		●					●		●
Chen et al. [16]	2020	●	●							●	●					●
He et al. [40]	2020	●							●		●					●
Hishamoto et al. [46]	2020	●							●		●					●
Jagielski et al. [50]	2020	●							●					●		●
Krishna et al. [63]	2020	●							●					●		●
Leino and Fredrikson	2020			●	●				●		●					●
Pal et al. [100]	2020	●							●					●		●
Pan et al. [101]	2020		●						●			●				●
Salem et al. [109]	2020	●							●			●				●
Song & Raghunathan [117]	2020	●	●						●		●	●	●			●
Yu et al. [142]	2020	●							●					●		●
Zhang et al. [145]	2020		●						●			●				●
Carlini et al. [12]	2021	●							●		●					●
Choquette-Choo et al. [18]	2021	●							●		●					●
Gong et al. [32]	2021	●							●					●		●
Li & Zhang [68]	2021	●							●		●					●
Mahloujifar et al. [78]	2022	●			●				●				●		●	
Zhang et al. [146]	2022	●							●		●	●	●			●

The transparent circle in the Knowledge column indicates partial white-box attacks.

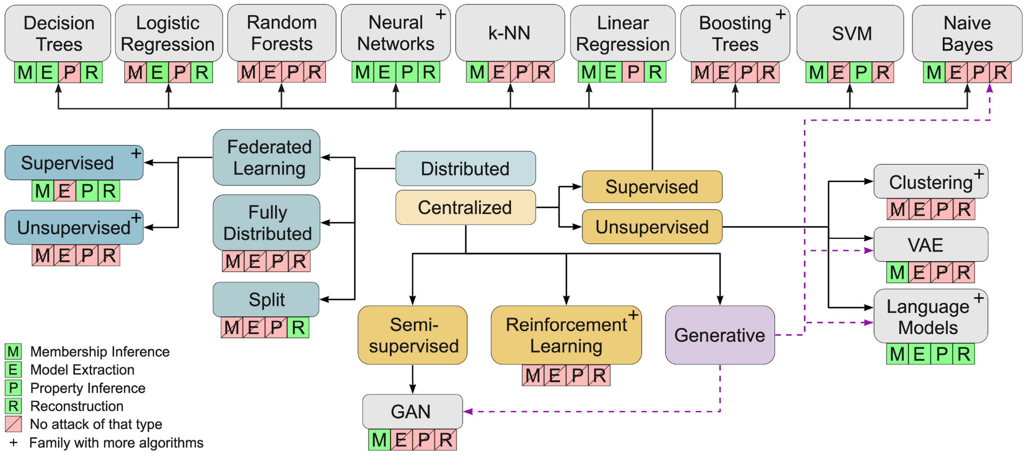


Fig. 4. Map of attack types per algorithm. The list of algorithm presented is not exhaustive but indicative. Underneath each algorithm or area of machine learning there is an indication of the attacks that have been studied so far. A red box indicates no attack.

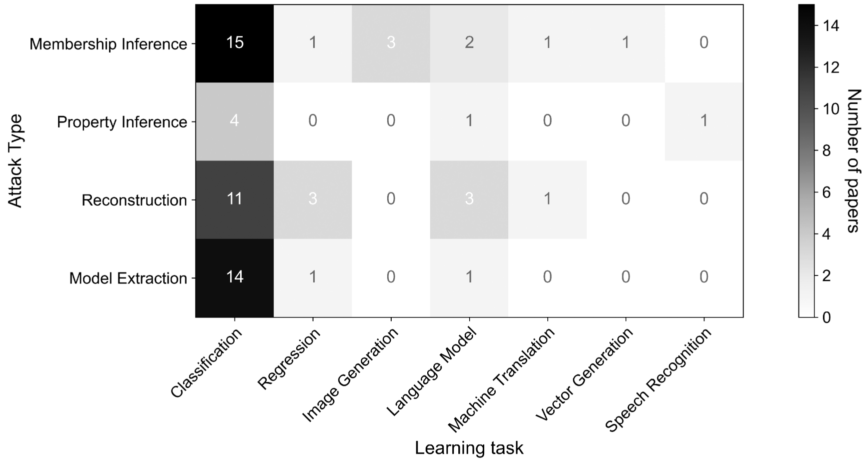
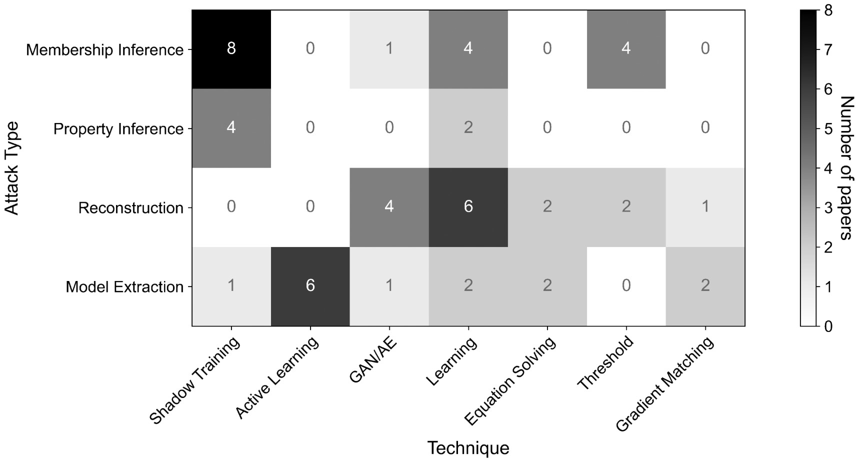


Fig. 5. Number of papers used against each learning task and attack type. Classification includes both binary and multi-class classification. Darker gray means higher number of papers.

queries for substitute model creation [21, 50, 98] and learning classifiers from language models for reconstruction [101] and property inference [117]. In "Threshold"-based attacks, we categorized the attacks proposed in References [140] and [108] and subsequent papers that used them for membership and property inference.

Some attacks may be applicable to multiple learning tasks and datasets, however, this is not the case universally. Dataset size, number of classes, and features might also be factors for the success of certain attacks, especially since most of them are empirical. Table 2 is a summary of the datasets used in all attack papers along with the data types of their features, the learning task they were used for, and the dataset size. The datasets were used during the training of the target models and in some cases as auxiliary information during the attacks. The table contains 56 unique datasets used across 53 papers, an indication of the variation of different approaches.



	Shadow Training	Active Learning	GAN/AE	Learning	Equation Solving	Threshold	Gradient Matching
Membership Inference	8	0	1	4	0	4	0
Property Inference	4	0	0	2	0	0	0
Reconstruction	0	0	4	6	2	2	1
Model Extraction	1	6	1	2	2	0	2

Fig. 6. Number of papers that used an attacking technique for each attack type. Darker gray means higher number of papers.

This high variation is both a blessing and a curse. On the one hand, it is highly desirable to use multiple types of datasets to test different hypotheses and the majority of the reviewed research follows that approach. On the other hand, these many options make it harder to compare methods. As it is evident from Table 2, some of the datasets are quite popular. MNIST, CIFAR-10, CIFAR-100, and UCI Adult have been used by more than nine papers, while 29 datasets have been used by only one paper.

The number of model parameters varies based on the model, task and datasets used in the experiments. As it can be seen in Table 2, most datasets are not extremely large, hence the models under attack are not extremely large. Given that most papers deal with neural networks, this might indicate that most attacks focused on smaller datasets and models, which might not be representative of realistic scenarios. However, privacy attacks do not necessarily have to target large models with extreme amounts of data; and neural networks, however popular, are not necessarily the most used models in the "real world."

7 DEFENDING MACHINE LEARNING PRIVACY

Leaking personal information such as medical records or credit card numbers is usually an undesirable situation. The purpose of studying attacks against machine learning models is to be able to explore the limitations and assumptions of machine learning and to anticipate the adversaries' actions. Most of the analyzed papers propose and test mitigations to counter their attacks. In the next subsections, we present the various defences proposed in several papers organized by the type of attack they attempt to defend against.

7.1 Defenses Against Membership Inference Attacks

The most prominent defense against membership inference attacks is **Differential Privacy (DP)**, which provides a guarantee on the impact that single data records have on the output of an algorithm or a model. However, other defenses have been tested empirically and are also presented in the following subsections.

7.1.1 Differential Privacy. Differential privacy started as a privacy definition for data analysis and it is based on the idea of "learning nothing about an individual while learning useful information about a population" [25]. Its definition is based on the notion that if two databases differ only

Table 2. Summary of Datasets Used in the Papers About Privacy Attacks on Machine Learning Systems

Name	Data Type	Learning Task	Reference(s)	Size (Samples)
538 Steak Survey [42]	mixed features	multi-class classification	[15, 27, 43, 124]	332
AT&T Faces [4]	images	multi-class classification	[27, 47, 134]	400
Bank Marketing [24]	mixed features	multi-class classification	[131]	45,210
Bitcoin prices	time series	regression	[124]	1,076
Book Corpus [149]	text	word-level language model	[117]	14,000 sent.
Breast Cancer [24]	numerical feat.	binary classification	[15, 66, 74, 124]	699
Caltech 256 [35]	images	multi-class classification	[98]	30,607
Caltech birds [130]	images	multi-class classification	[98]	6,033
CelebA [70]	images	binary classification	[6, 16, 29, 138, 145]	20-202,599
CIFAR-10 [64]	images	image generation, multi-class classification	[6, 18, 32, 39, 41, 44, 50, 66, 68, 86, 100, 106, 108–110, 116, 120, 125, 138, 140]	60,000
CIFAR-100 [64]	images	multi-class classification	[6, 18, 51, 66, 68, 90, 110, 116, 140, 148]	60,000
Cityscapes [20]	images	image segmantation	[40]	3,475
CLiPS stylometry [128]	text	binary classification	[84]	1,412 reviews
Chest X-ray [133]	images	multi-class classification	[145]	10,000
Diabetes [24]	time series	binary class., regression	[15, 124, 131]	768
Diabetic ret. [58]	images	image generation	[39, 98]	88,702
Enron emails	text	char-level language model	[11, 78]	—
Eyedata [111]	numerical feat.	regression	[140]	120
FaceScrub [93]	images	binary classification	[84, 138]	18,809–48,579
Fashion-MNIST [136]	images	multi-class classification	[6, 44, 50, 120]	60,000
Foursquare [137]	mixed features	binary classification	[84, 110, 116]	528,878
Geog. Orig. Music [24]	numerical feat.	regression	[131]	1,059
German Credit [24]	mixed features	binary classification	[66, 124]	1,000
GSS marital survey [36]	mixed features	multi-class classification	[15, 27, 124]	16,127
GTSRB [122]	images	multi-class classification	[32, 56, 68, 100, 103, 142]	51,839
HW Perf. Counters (private)	numerical feat.	binary classification	[29]	36,000
Imagenet [23]	images	multi-class classification	[6, 50, 97, 108]	14,000,000
Instagram [5]	location data	vector generation	[16]	—
Iris [26]	numerical feat.	multi-class classification	[15, 124]	150
IWPC [19]	mixed features	regression	[28, 140]	3,497
IWSLT Eng-Vietnamese [76]	text	neural machine translation	[11]	—
KDEF [75]	images	multi-class classification	[142]	4,900
LFW [48]	images	image generation	[39, 66, 68, 84, 148]	13,233
Madelon [24]	mixed features	multi-class classification	[131]	4,400
MIMIC-III [55]	binary features	record generation	[16]	41,307
Movielens 1M [37]	numerical feat.	regression	[43]	1,000,000
MNIST [65]	images	multi-class classification	[15, 29, 32, 41, 44, 47, 50, 56, 66, 74, 86, 97, 100, 103, 106, 109, 110, 116, 124, 125, 134, 138, 140, 145, 148]	70,000

(Continued)

Table 2. Continued from previous page

Name	Data Type	Learning Task	Reference(s)	Size (Samples)
MR [102]	text	multi-class classification	[100]	–
Mushrooms [24]	categorical feat.	binary classification	[15, 124]	8,124
Netflix [91]	binary features	binary classification	[140]	2,416
Netflows (private)	network data	binary classification	[3]	–
PTB [81]	text	char-level language model	[11]	5 MB
PiPA [144]	images	binary classification	[84]	18,000
Purchase-100 [57]	binary features	multi-class classification	[18, 51, 90, 116, 125]	197,324
SVHN [92]	images	multi-class classification	[50, 148]	60,000
TED talks [49]	text	machine translation	[11]	100,000 pairs
Texas-100 [14]	mixed features	multi-class classification	[18, 90, 116]	67,330
TU Dataset [87]	graph data	graph classification & regression	[146]	–
UJIndoor [24]	mixed features	regression	[131]	19,937
UCI / Adult [24]	various	binary classification	[15, 18, 29, 74, 78, 110, 116, 124, 125]	48,842
VGG Flowers [95]	images	multi-class classification	[142]	6,146
Voxforge [129]	audio	speech recognition	[3]	11,137 rec.
Wikipedia [79]	text	language model	[117]	150,000 articles
Wikitext-103 [85]	text	word-level language model	[11, 63]	500 MB
Yale-Face [30]	images	multi-class classification	[120]	2,414
Yelp reviews [139]	text	binary classification	[84]	16–40,000

The size of each dataset is measured by the number of samples unless otherwise indicated. A range in the size column indicates that different papers used different subsets of the dataset.

by one record and are used by the same algorithm (or mechanism), the output of that algorithm should be similar. More formally,

Definition 7.1 ((ϵ, δ)-Differential Privacy). A randomized mechanism $\mathcal{M}$ with domain $\mathcal{R}$ and output $\mathcal{S}$ is (ϵ, δ)-differentially private if for any adjacent inputs $D, D' \in \mathcal{R}$ and for any subsets of outputs $\mathcal{S}$ it holds that

$$Pr[\mathcal{M}(D) \in \mathcal{S}] \leq e^{\epsilon} Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta, \tag{8}$$

where ϵ is the privacy budget and δ is the failure probability.

The original definition of DP did not include δ, which was introduced as a relaxation that allows some outputs not to be bounded by e^{ϵ}.

The usual application of DP is to add Laplacian or Gaussian noise to the output of a query or function over the database. The amount of noise is relevant to the *sensitivity*, which gives an upper bound on how much we must perturb the output of the mechanism to preserve privacy [25]:

Definition 7.2. l_1 (or l_2)-Sensitivity of a function f is defined as

$$\Delta f = \max_{D, D', \|D - D'\| = 1} \|f(D) - f(D')\|, \tag{9}$$

where $\|.\|$ is the l_1 or the l_2-norm and the max is calculated over all possible inputs D, D'.

From a machine learning perspective, D and D' are two datasets that differ by one training sample and the randomized mechanism $\mathcal{M}$ is the machine learning training algorithm. In deep learning, the noise is added at the gradient calculation step. Because it is necessary to bound the gradient norm, gradient clipping is also applied [1].

Differential privacy offers a trade-off between privacy protection and utility or model accuracy. Evaluation of differentially private machine learning models against membership inference attacks concluded that the models could offer privacy protection only when they considerably sacrifice their utility [51, 106]. Jayaraman et al. [51] evaluated several relaxations of DP in both logistic regression and neural network models against membership inference attacks. They showed that these relaxations have an impact on the utility-privacy trade-off. While they reduce the required added noise, they also increase the privacy leakage.

Distributed learning scenarios require additional considerations regarding differential privacy. In a centralized model, the focus is on sample level DP, i.e., on protecting privacy at the individual data point level. In a federated learning setting where there are multiple participants, we not only care about the individual training data points they use but also about ensuring privacy at the participant level. A proposal that applies DP at the participant level was introduced by McMahan et al. [83]; however, it requires a large number of participants. When it was tested with a number as low as 30, the method was deemed unsuccessful [84].

7.1.2 Regularization. Regularization techniques in machine learning aim to reduce overfitting and increase model generalization performance. Dropout [121] is a form of regularization that randomly drops a predefined percentage of neural network units during training. Given that black-box membership inference attacks are connected to overfitting, it is a sensible approach to this type of attack and multiple papers have proposed it as a defense with varying levels of success [39, 84, 110, 116, 120]. Another form of regularization uses techniques that combine multiple models that are trained separately. One of those methods, model stacking, was tested in Reference [110] and produced positive results against membership inference. An advantage of model stacking or similar techniques is that they are model agnostic and do not require that the target model is a neural network.

7.1.3 Prediction Vector Tampering. As many models assume access to the prediction vector during inference, one of the countermeasures proposed was the restriction of the output to the top k classes or predictions of a model [116]. However, this restriction, even in the strictest form (outputting only the class label) did not seem to fully mitigate membership inference attacks, since information leaks can still happen due to model misclassifications. Another option is to lower the precision of the prediction vector, which leads to less information leakage [116]. Adding noise to the output vector also affected membership inference attacks [54].

7.2 Defenses Against Reconstruction Attacks

Reconstruction attacks often require access to the loss gradients during training. Most of the defences against reconstruction attacks propose techniques that affect the information retrieved from these gradients. Setting all loss gradients that are below a certain threshold to zero was proposed as a defence against reconstruction attacks in deep learning. This technique proved quite effective with as little as 20% of the gradients set to zero and with negligible effects on model performance [148]. However, performing quantization or using half-precision floating points for neural network weights did not seem to deter the attacks in References [11] and [148], respectively.

7.3 Defenses Against Property Inference Attacks

Differential privacy is designed to provide privacy guarantees in membership inference attack scenarios and it does not seem to offer protection against property inference attacks [3]. In addition to DP, Melis et al. [84] explored other defenses against property inference attacks. Regularization (dropout) had an adverse effect and actually made the attacks stronger. Since the attacks in Reference [84] were performed in a collaborative setting, the authors tested the proposal in Reference

[115], which is to share fewer gradients between training participants. Although sharing less information made the attacks less effective, it did not alleviate them completely.

7.4 Defenses Against Model Extraction Attacks

Model extraction attacks usually require that the attacker performs a number of queries on the target model. The goal of the proposed defenses so far has been the detection of these queries. This contrasts with the previously presented defences that mainly try to prevent attacks.

7.4.1 Protecting Against DNN Model Stealing Attacks (PRADA). Detecting model stealing attacks based on the model queries that are used by the adversary was proposed by Juuti et al. [56]. The detection is based on the assumption that model queries that try to explore decision boundaries will have a different distribution than the normal ones. While the detection was successful, the authors noted that it is possible to be evaded if the adversary adapts their strategy.

7.4.2 Membership Inference. The idea of using membership inference to defend against model extraction was studied by Krishna et al. [63]. It is based on the premise that using membership inference, the model owner can distinguish between legitimate user queries and nonsensical ones whose only purpose is to extract the model. The authors note that this type of defence has limitations such as potentially flagging legitimate but out-of-distribution queries made by legitimate users, but more importantly that they can be evaded by adversaries that make adaptive queries.

7.4.3 Obfuscating Prediction. Zheng et al. [147] proposed to obfuscate the predictions of the target model for data points that are near the decision boundary using the idea of boundary DP, which guarantees to defend the model regardless of the number of queries and it was tested in a binary classification setting. Orekondy et al. [99] proposed to perturb the prediction vectors of the target model in such a way that they poison the substitute model created by the attacker.

8 FUTURE RESEARCH DIRECTIONS

8.1 Causes of Privacy Attacks

Attacks on machine learning privacy have been increasingly brought to light. However, we are still at an exploratory stage. For some attacks such as membership inference, we know that there is a connection to overfitting and memorization but there are also indications that some data points are easier to infer than others. Currently, we lack a deeper understanding about the rest of the inference attacks, especially property inference and model extraction attacks. Potential future work may be done, using a data-centric approach, on why some data points may be easier to infer, and which are their properties.

8.2 Real-world Impact

As much as we need answers about why leaks happen at a theoretical level, we also need to know how well privacy attacks work on real deployed systems. Adversarial attacks on realistic systems bring to light the issue of additional constraints that need to be in place for the attacks to work. For example, when creating glasses to fool a face recognition system, Sharif et al. [114] had to pose constraints that had to do with physical realizations, e.g., that the color of the glasses should be printable. In privacy-related attacks, the most realistic cases come from the model extraction area, where attacks against MLaaS systems have been demonstrated in multiple papers. For the majority of other attacks, it is certainly an open question of how well they would perform on deployed models and what kind of additional requirements would need to be in place for them to succeed. Potential future work may be done by exploring attacks in real-world implementations and deployments, where the assumptions about the access to the training dataset are a challenge.

8.3 Datasets and Attack Evaluation

Beyond expanding the focus on different learning tasks, there is the question of datasets. The impact of datasets on the attack success has been demonstrated by several papers. Yet, we currently lack a common approach as to which datasets are best suited to evaluate privacy attacks, or what constitutes the minimum requirements for a successful attack. Several questions are worth considering: Do we need standardized datasets? If yes, then how do we go about and create them? Are all data worth protecting? If some are more interesting than others, then is it not a good idea to test attacks beyond popular image datasets? A potential future work would be the standardization of datasets and testing environments. This may be an avenue that would benefit researchers to advance the state of the art, as well as engineers that deploy models in production environments. Part of this future direction should also include guidelines for evaluation and deployment with respect to potential privacy leaks.

8.4 Unexplored Privacy Attacks on Machine Learning

The main research focus up to now has been supervised learning. Even within supervised learning, there are areas and learning tasks that have been largely unexplored. Information leakage in newer model architectures such as attention networks and graph neural networks just recently started to attract some focus. In addition there are very few attacks reported on popular algorithms such as random forests or gradient boosting trees despite their wide application and this is research area that deserves more attention and future work. In unsupervised and semi-supervised learning, the focus is mainly on generative models, and only just recently, papers started exploring areas such as representation learning and language models. Some attacks on image classifiers do not transfer that well to natural language processing tasks [46], while others do but may require different sets of assumptions and design considerations [101]. Figure 4 shows which type of attacks were still not tried in which models or algorithms. All these are potential future work directions.

8.5 Interdisciplinary Attacks

Finally, as we strive to understand the privacy implications of machine learning, we also realize that several research areas are connected and affect each other. We know, for instance, that adversarial training (using adversarial samples to make the model more robust) affects membership inference [119] and that model censoring can still leak private attributes [119]. Property inference attacks can deduce properties of the training dataset that were not specifically encoded or were not necessarily correlated to the learning task. This can be understood as a form of bias detection, which means that relevant literature in the area of model fairness should be reviewed as potentially complementary. Furthermore, while deep learning models are considered black-boxes in terms of explainability, future work that sheds light on what type of data make neurons activate [94, 141] can be relevant to discovering information about the target's training dataset and can therefore lead to privacy leaks. All these are examples of potential inter-dependencies between different areas of machine learning research. Therefore, a better understanding of privacy attacks calls for an interdisciplinary approach.

9 CONCLUSION

As machine learning becomes ubiquitous, the scientific community becomes increasingly interested in its impact and side-effects in terms of security, privacy, fairness, and explainability. This survey conducted a comprehensive study of the state-of-the-art privacy-related attacks and proposed a threat model and a unifying taxonomy of the different types of attacks based on their characteristics. An in-depth examination of the current state-of-the-art research allowed us to perform a detailed analysis, which revealed common design patterns and differences between them.

Several open problems that merit further research were identified. First, our analysis revealed a somewhat narrow focus of the research conducted so far, which is dominated by attacks on deep learning models. We believe that there are several popular algorithms and models in terms of real-world deployment and applicability that merit a closer examination. Second, a thorough theoretical understanding of the reasons behind privacy leaks is still underdeveloped and this affects both the proposed defensive measures and our understanding of the limitations of privacy attacks. Experimental studies on factors that affect privacy leaks have provided useful insights so far. However, in total, there are very few works that test attacks in realistic conditions in terms of dataset size and deployment. Finally, examining the impact of other adjacent study areas such as security, explainability, and fairness is also a topic that calls for further exploration. Even though it may not be possible to construct and deploy models that are fully private against all types of adversaries, understanding the inter-dependencies that affect privacy will help make more informed decisions.

While the community is still in an exploratory mode regarding privacy leaks of machine learning systems, we hope that this survey will provide the necessary background to both the interested readers as well as the researchers that wish to work on this topic.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS'16)*. ACM, New York, NY, 308–318. https://doi.org/10.1145/2976749.2978318

[2] Mohammad Al-Rubaie and Morris J. Chang. 2019. Privacy-preserving machine learning: Threats and solutions. *IEEE Secur. Privacy* 17, 2 (2019), 49–58. https://doi.org/10.1109/MSEC.2018.2888775

[3] Giuseppe Ateniese, Luigi V. Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. 2015. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *Int. J. Secur. Netw.* 10, 3 (Sept. 2015), 137–150. https://doi.org/10.1504/IJSN.2015.071829

[4] AT&T 1994. Database of Faces. Retrieved April 17, 2020 from http://cam-orl.co.uk/facedatabase.html

[5] Michael Backes, Mathias Humbert, Jun Pang, and Yang Zhang. 2017. Walk2friends: Inferring social links from mobility profiles. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS'17)*. ACM, New York, NY, 1943–1957. https://doi.org/10.1145/3133956.3133972

[6] Antonio Barbalau, Adrian Cosma, Radu Tudor Ionescu, and Marius Popescu. 2020. Black-box ripper: copying black-box models using generative evolutionary algorithms. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS'20)*, Vol. 33. 20120–20129.

[7] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A. Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*. NeurIPS, 5050–5060.

[8] Battista Biggio and Fabio Roli. 2018. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recogn.* 84 (2018), 317–331.

[9] Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin.

[10] Leo Breiman. 2001. Random forests. *Mach. Learn.* 45, 1 (2001), 5–32.

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

[12] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *Proceedings of the 30th USENIX Security Symposium (USENIX Security'21)*. USENIX Association, 2633–2650.

[13] Augustin Cauchy et al. 1847. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris* 25, 1847 (1847), 536–538.

[14] Texas Health Care Information Collection Center. 2006-2009. Texas Inpatient Public Use Data File (PUDF). Retrieved April 17, 2020 from https://www.dshs.texas.gov/thcic/hospitals/Inpatientpudf.shtm

[15] Varun Chandrasekaran, Kamalika Chaudhuri, Irene Giacomelli, Somesh Jha, and Songbai Yan. 2020. Exploring connections between active learning and model extraction. In *Proceedings of the 29th USENIX Security Symposium (USENIX Security'20)*. USENIX Association.

[16] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. 2020. GAN-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security* (Virtual Event) *(CCS'20)*. ACM, New York, NY, 343–362. https://doi.org/10.1145/3372297.3417238

[17] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*. ACM, New York, NY, 785–794. https://doi.org/10.1145/2939672.2939785

[18] Christopher A. Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. 2021. Label-only membership inference attacks. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 1964–1974.

[19] International Warfarin Pharmacogenetics Consortium. 2009. Estimation of the warfarin dose with clinical and pharmacogenetic data. *New Engl. J. Med.* 360, 8 (2009), 753–764.

[20] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 3213–3223.

[21] Jackson Rodrigues Correia-Silva, Rodrigo F. Berriel, Claudine Badue, Alferto F. de Souza, and Thiago Oliveira-Santos. 2018. Copycat CNN: Stealing knowledge by persuading confession with random non-labeled data. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8. https://doi.org/10.1109/IJCNN.2018.8489592

[22] Jeffrey Dean, Greg S. Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, and Andrew Y. Ng. 2012. Large scale distributed deep networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems—Volume 1 (NIPS'12)*. Curran Associates, 1223–1231.

[23] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 248–255.

[24] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. Retrieved April 17, 2020 from http://archive.ics.uci.edu/ml

[25] Cynthia Dwork and Aaron Roth. 2013. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* 9, 3-4 (2013), 211–487. https://doi.org/10.1561/0400000042

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

[27] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS'15)*. ACM, New York, NY, 1322–1333. https://doi.org/10.1145/2810103.2813677

[28] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. 2014. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *Proceedings of the 23rd USENIX Security Symposium (USENIX Security'14)*. USENIX Association, 17–32.

[29] Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. 2018. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS'18)*. ACM, New York, NY, 619–633. https://doi.org/10.1145/3243734.3243834

[30] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intell.* 23, 6 (2001), 643–660.

[31] Neil Zhenqiang Gong and Bin Liu. 2016. You are who you know and how you behave: Attribute inference attacks via users' social friends and behaviors. In *Proceedings of the 25th USENIX Security Symposium (USENIX Security'16)*. USENIX Association, 979–995.

[32] Xueluan Gong, Yanjiao Chen, Wenbin Yang, Guanghao Mei, and Qian Wang. 2021. InverseNet: Augmenting model extraction attacks with training data inversion. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI'21)*, Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 2439–2447. https://doi.org/10.24963/ijcai.2021/336 Main Track.

[33] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press, Cambridge, MA.

[34] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems—Volume 2 (NIPS'14)*. MIT Press, Cambridge, MA, 2672–2680.

[35] Gregory Griffin, Alex Holub, and Pietro Perona. 2007. *The Caltech 256*. Technical Report. Pasadena, CA.

[36] GSS. 2006. *Social Science Research on Pornography*. Retrieved April 17, 2020 from https://byuresearch.org/ssrp/downloads/GSS.xls

[37] F. Maxwell Harper and Joseph A. Konstan. 2015. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (Dec. 2015), 19 pages. https://doi.org/10.1145/2827872

[38] Johann Hauswald, Thomas Manville, Qi Zheng, Ronald Dreslinski, Chaitali Chakrabarti, and Trevor Mudge. 2014. A hybrid approach to offloading mobile image classification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'14)*. IEEE, 8375–8379.

[39] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. 2019. LOGAN: Membership inference attacks against generative models. In *Proceedings on Privacy Enhancing Technologies*, Vol. 2019. 133–152. https://doi.org/10.2478/popets-2019-0008

[40] Yang He, Shadi Rahimian, Bernt Schiele, and Mario Fritz. 2020. Segmentations-leak: Membership inference attacks and defenses in semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV'20)*. Springer International Publishing, Cham, 519–535.

[41] Zecheng He, Tianwei Zhang, and Ruby B. Lee. 2019. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference (ACSAC'19)*. ACM, New York, NY, 148–162. https://doi.org/10.1145/3359789.3359824

[42] Walt Hickey. 2014. DataLab: How Americans like their steak. Retrieved April 17, 2020 from http://fivethirtyeight.com/datalab/how-americans-like-their-steak

[43] Seira Hidano, Takao Murakami, Shuichi Katsumata, Shinsaku Kiyomoto, and Goichiro Hanaoka. 2017. Model inversion attacks for prediction systems: Without knowledge of non-sensitive attributes. In *Proceedings of the 15th Annual Conference on Privacy, Security and Trust (PST'17)*. IEEE, 115–124.

[44] Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. 2019. Monte Carlo and reconstruction membership inference attacks against generative models. In *Proceedings on Privacy Enhancing Technologies*. 232–249. https://doi.org/10.2478/popets-2019-0067

[45] Geoffrey Hinton, Nitsh Srivastava, and Kevin Swersky. 2012. Neural networks for machine learning. *Coursera Video Lect.* 264, 1 (2012).

[46] Sorami Hisamoto, Matt Post, and Kevin Duh. 2020. Membership inference attacks on sequence-to-sequence models: Is my data in your machine translation system? *Trans. Assoc. Comput. Ling.* 8 (2020), 49–63. https://doi.org/10.1162/tacl_a_00299

[47] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. 2017. Deep models under the GAN: Information leakage from collaborative deep learning. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS'17)*. ACM, New York, NY, 603–618. https://doi.org/10.1145/3133956.3134012

[48] Gary B. Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. 2008. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Proceedings of the Workshop on Faces in "Real-life" Images: Detection, Alignment, and Recognition*. Erik Learned-Miller and Andras Ferencz and Frédéric Jurie, HAL. Retrieved from https://hal.inria.fr/inria-00321923

[49] International Conference on Spoken Language Translation 2015. IWSLT Evaluation 2015. Retrieved April 17, 2020 from https://sites.google.com/site/iwsltevaluation2015

[50] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. 2020. High accuracy and high fidelity extraction of neural networks. In *Proceedings of the 29th USENIX Security Symposium (USENIX Security'20)*. USENIX Association.

[51] Bargav Jayaraman and David Evans. 2019. Evaluating differentially private machine learning in practice. In *Proceedings of the 28th USENIX Security Symposium (USENIX Security'19)*. USENIX Association, 1895–1912.

[52] Malhar S. Jere, Tyler Farnan, and Farinaz Koushanfar. 2020. A taxonomy of attacks on federated learning. *IEEE Secur. Privacy* (2020), 0–0. https://doi.org/10.1109/MSEC.2020.3039941

[53] Jinyuan Jia and Neil Zhenqiang Gong. 2018. AttriGuard: A practical defense against attribute inference attacks via adversarial machine learning. In *Proceedings of the 27th USENIX Security Symposium (USENIX Security'18)*. USENIX Association, 513–529.

[54] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. 2019. MemGuard: Defending against black-box membership inference attacks via adversarial examples. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS'19)*. ACM, New York, NY, 259–274. https://doi.org/10.1145/3319535.3363201

[55] Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, H. Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. 2016. MIMIC-III, a freely accessible critical care database. *Sci. Data* 3 (2016), 160035.

[56] Mika Juuti, Sebastian Szyller, Samuel Marchal, and N. Asokan. 2019. PRADA: Protecting against DNN model stealing attacks. In *Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P'19)*. IEEE, 512–527.

[57] Kaggle. 2014. *Acquire Valued Shoppers Challenge*. Retrieved April 17, 2020 from https://www.kaggle.com/c/acquire-valued-shoppers-challenge/data

[58] Kaggle. 2015. Diabetic Retinopathy Detection. Retrieved April 17, 2020 from https://www.kaggle.com/c/diabetic-retinopathy-detection

[59] Yiping Kang, Johann Hauswald, Cao Gao, Austin Rovinski, Trevor Mudge, Jason Mars, and Lingjia Tang. 2017. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. In *Proceedings of the 22nd International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'17)*. ACM, New York, NY, 615–629. https://doi.org/10.1145/3037697.3037698

[60] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*. Curran Associates, 3149–3157.

[61] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. Retrieved from https://arXiv:1412.6980

[62] Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR'14)*, Vol. 1. ICLR.

[63] Kalpesh Krishna, Gaurav Singh Tomar, Ankur P. Parikh, Nicolas Papernot, and Mohit Iyyer. 2020. Thieves on sesame street! Model extraction of BERT-based APIs. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*. ICLR, Virtual Conference.

[64] Alex Krizhevsky, Geoffrey Hinton et al. 2009. Learning multiple layers of features from tiny images. MSc thesis. http://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf

[65] Yann LeCun, Corinna Cortes, and Christopher J. C. Burges. 1998. The MNIST database of handwritten digits. Retrieved April 17, 2020 from http://yann.lecun.com/exdb/mnist/

[66] Klas Leino and Matt Fredrikson. 2020. Stolen memories: Leveraging model memorization for calibrated white-box membership inference. In *Proceedings of the 29th USENIX Security Symposium (USENIX Security'20)*. 1605–1622.

[67] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2019. Federated learning: Challenges, methods, and future directions. Retrieved from https://1908.07873

[68] Zheng Li and Yang Zhang. 2021. Membership leakage in label-only exposures. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS'21)*. ACM, New York, NY, 880–895. https://doi.org/10.1145/3460120.3484575

[69] Dong C. Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Math. Program.* 45, 1-3 (1989), 503–528.

[70] Jian Liu, Mika Juuti, Yao Lu, and N. Asokan. 2017. Oblivious neural network predictions via MiniONN transformations. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS'17)*. ACM, New York, NY, 619–631. https://doi.org/10.1145/3133956.3134056

[71] Shuying Liu and Weihong Deng. 2015. Very deep convolutional neural network based image classification using small training sample size. In *Proceedings of the 3rd IAPR Asian Conference on Pattern Recognition (ACPR'15)*. IEEE, 730–734.

[72] Ximeng Liu, Lehui Xie, Yaopeng Wang, Jian Zou, Jinbo Xiong, Zuobin Ying, and Athanasios V. Vasilakos. 2021. Privacy and security issues in deep learning: A survey. *IEEE Access* 9 (2021), 4566–4593. https://doi.org/10.1109/ACCESS.2020.3045078

[73] Yugeng Liu, Rui Wen, Xinlei He, Ahmed Salem, Zhikun Zhang, Michael Backes, Emiliano De Cristofaro, Mario Fritz, and Yang Zhang. 2021. ML-doctor: Holistic risk assessment of inference attacks against machine learning models. Retrieved from https://arXiv:2102.02551

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

[75] Ellen Goeleven, Rudi De Raedt, Lemke Leyman, and Bruno Verschuere. 2008. The Karolinska directed emotional faces: A validation study. *Cognition and Emotion* 22, 6 (2008), 1094–1118. DOI: 10.1080/02699930701626582

[76] Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domain. In *Proceedings of the International Workshop on Spoken Language Translation.*

[77] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*. ICLR.

[78] S. Mahloujifar, E. Ghosh, and M. Chase. 2022. Property inference from poisoning. In *Proceedings of the IEEE Symposium on Security and Privacy (SP'22)*. IEEE Computer Society, 1569–1569. https://doi.org/10.1109/SP46214.2022.00140

[79] Matt Mahoney. [n.d.]. Large Text Compression Benchmark. Retrieved March 8, 2021 from http://mattmahoney.net/dc/text.html

[80] Davide Maiorca, Battista Biggio, and Giorgio Giacinto. 2019. Towards adversarial malware detection: Lessons learned from PDF-based attacks. *ACM Comput. Surv.* 52, 4, Article 78 (Aug. 2019), 36 pages. https://doi.org/10.1145/3332184

[81] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The Penn Treebank. *Comput. Linguist.* 19, 2 (June 1993), 313–330.

[82] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 54)*, Aarti Singh and Jerry Zhu (Eds.). PMLR, 1273–1282.

[83] Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning differentially private recurrent language models. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*. ICLR.

[84] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting unintended feature leakage in collaborative learning. In *Proceedings of the IEEE Symposium on Security and Privacy (SP'19)*. IEEE, San Francisco, CA, 691–706.

[85] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. Retrieved from https://arXiv:1609.07843

[86] Smitha Milli, Ludwig Schmidt, Anca D. Dragan, and Moritz Hardt. 2019. Model reconstruction from model explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*'19)*. ACM, New York, NY, 1–9. https://doi.org/10.1145/3287560.3287562

[87] Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. 2020. TUDataset: A collection of benchmark datasets for learning with graphs. In *Proceedings of the ICML Workshop on Graph Representation Learning and Beyond (GRL+'20)*. www.graphlearning.io

[88] Kevin P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. MIT Press, Cambridge, MA.

[89] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2018. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security* (Toronto, Canada) *(CCS'18)*. ACM, New York, NY, 634–646. https://doi.org/10.1145/3243734.3243855

[90] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *Proceedings of the IEEE Symposium on Security and Privacy (SP'19)*. IEEE, 739–753.

[91] Netflix 2009. Netflix prize. Retrieved April 17, 2020 from https://www.netflixprize.com

[92] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. 2011. Reading digits in natural images with unsupervised feature learning. In *Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning*. NIPS, Granada, Spain.

[93] Hong-Wei Ng and Stefan Winkler. 2014. A data-driven approach to cleaning large face datasets. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'14)*. IEEE, 343–347.

[94] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. 2016. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16)*. Curran Associates, 3395–3403.

[95] Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*.

[96] Jorge Nocedal and Stephen J Wright. 2006. *Numerical Optimization*. Springer, Berlin.

[97] Seong Joon Oh, Max Augustin, Mario Fritz, and Bernt Schiele. 2018. Towards reverse-engineering black-box neural networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*. ICLR.

[98] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2019. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 4954–4963.

[99] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2020. Prediction poisoning: Towards defenses against DNN model stealing attacks. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*.

[100] Soham Pal, Yash Gupta, Aditya Shukla, Aditya Kanade, Shirish Shevade, and Vinod Ganapathy. 2020. ActiveThief: Model extraction using active learning and unannotated public data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. AAAI, 865–872. https://doi.org/10.1609/aaai.v34i01.5432

[101] Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. 2020. Privacy risks of general-purpose language models. In *Proceedings of the IEEE Symposium on Security and Privacy (SP'20)*. IEEE, 1314–1331.

[102] Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL'05)*. Association for Computational Linguistics, 115–124. https://doi.org/10.3115/1219840.1219855

[103] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the ACM on Asia Conference on Computer and Communications Security (ASIA CCS'17)*. ACM, New York, NY, 506–519. https://doi.org/10.1145/3052973.3053009

[104] Nicholas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael P. Wellman. 2018. SoK: Security and privacy in machine learning. In *Proceedings of the IEEE European Symposium on Security and Privacy (EuroSP'18)*. IEEE, 399–414.

[105] Boris T. Polyak. 1964. Some methods of speeding up the convergence of iteration methods. *Ussr Computational Mathematics and Mathematical Physics* 4, 5 (1964), 1–17.

[106] Md Atiqur Rahman, Tanzila Rahman, Robert Laganière, Noman Mohammed, and Yang Wang. 2018. Membership inference attack against differentially private deep learning model. *Trans. Data Priv.* 11, 1 (2018), 61–79.

[107] Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *Ann. Math. Stat.* (1951), 400–407.

[108] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Herve Jegou. 2019. White-box vs black-box: Bayes optimal strategies for membership inference. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 5558–5567.

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

[110] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2019. ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS'19)*.

[111] Todd E. Scheetz, Kwang-Youn A. Kim, Ruth E. Swiderski, Alisdair R. Philp, Terry A. Braun, Kevin L. Knudtson, Anne M. Dorrance, Gerald F. DiBona, Jian Huang, Thomas L. Casavant et al. 2006. Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proc. Natl. Acad. Sci. U.S.A.* 103, 39 (2006), 14429–14434.

[112] Ozan Sener and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. In *Proceedings of the 6th International Conference on Learning Representations (ICLR'18)*.

[113] Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.

[114] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. 2016. Accessorize to a Crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS'16)*. ACM, New York, NY, 1528–1540. https://doi.org/10.1145/2976749.2978392

[115] Reza Shokri and Vitaly Shmatikov. 2015. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS'15)*. ACM, New York, NY, 1310–1321. https://doi.org/10.1145/2810103.2813687

[116] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *Proceedings of the IEEE Symposium on Security and Privacy (SP'17)*. IEEE, 3–18.

[117] Congzheng Song and Ananth Raghunathan. 2020. Information leakage in embedding models. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS'20)*. ACM, New York, NY, 377–390. https://doi.org/10.1145/3372297.3417270

[118] Congzheng Song and Vitaly Shmatikov. 2019. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'19)*. ACM, New York, NY, 196–206. https://doi.org/10.1145/3292500.3330885

[119] Congzheng Song and Vitaly Shmatikov. 2020. Overlearning reveals sensitive attributes. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*.

[120] Liwei Song, Reza Shokri, and Prateek Mittal. 2019. Privacy risks of securing machine learning models against adversarial examples. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS'19)*. ACM, New York, NY, 241–257. https://doi.org/10.1145/3319535.3354211

[121] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1 (2014), 1929–1958.

[122] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. 2011. The German traffic sign recognition benchmark: A multi-class classification competition. In *Proceedings of the International Joint Conference on Neural Networks*. IEEE, 1453–1460.

[123] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

[124] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction APIs. In *Proceedings of the 25th USENIX Security Symposium (USENIX Security'16)*. USENIX Association, 601–618.

[125] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. 2021. Demystifying membership Inference attacks in machine learning as a service. *IEEE Transactions on Services Computing* 14, 6 (2021), 2073–2089. DOI: 10.1109/TSC.2019.2897554

[126] V. Vapnik. 1991. Principles of risk minimization for learning theory. In *Proceedings of the 4th International Conference on Neural Information Processing Systems (NIPS'91)*. Morgan Kaufmann Publishers, 831–838.

[127] Michael Veale, Reuben Binns, and Lilian Edwards. 2018. Algorithms that remember: Model inversion attacks and data protection law. *Philos. Trans. Roy. Soc. A: Math., Phys. Eng. Sci.* 376, 2133 (2018), 20180083.

[128] Ben Verhoeven and Walter Daelemans. 2014. CLiPS stylometry investigation (CSI) corpus: A Dutch corpus for the detection of age, gender, personality, sentiment and deception in text. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14).* European Language Resources Association (ELRA), 3081–3085.

[129] VoxForge 2009. VoxForge Speech Corpus. Retrieved April 17, 2020 from http://www.voxforge.org/

[130] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. 2011. *The Caltech UCSD Birds-200-2011 Dataset.* Technical Report. Pasadena, CA.

[131] Binghui Wang and Neil Zhenqiang Gong. 2018. Stealing hyperparameters in machine learning. In *Proceedings of the IEEE Symposium on Security and Privacy (SP'18).* IEEE, 36–52.

[132] Xianmin Wang, Jing Li, Xiaohui Kuang, Yu-an Tan, and Jin Li. 2019. The security of machine learning in an adversarial setting: A survey. *J. Parallel Distrib. Comput.* 130 (2019), 12–23.

[133] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. 2017. ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17).* IEEE.

[134] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. 2019. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'19).* IEEE, 2512–2520.

[135] Xi Wu, Matthew Fredrikson, Somesh Jha, and Jeffrey F. Naughton. 2016. A methodology for formalizing model-inversion attacks. In *Proceedings of the 29th Computer Security Foundations Symposium (CSF'16).* IEEE, 355–370.

[136] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. Retrieved from https://arXiv:1708.07747

[137] Dingqi Yang, Daqing Zhang, Longbiao Chen, and Bingqing Qu. 2015. Nationtelescope: Monitoring and visualizing large-scale collective behavior in LBSNs. *J. Netw. Comput. Appl.* 55 (2015), 170–180.

[138] Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, and Zhenkai Liang. 2019. Neural network inversion in adversarial setting via background knowledge alignment. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS'19).* ACM, New York, NY, 225–240. https://doi.org/10.1145/3319535.3354261

[139] Yelp. [n.d.]. Yelp Open Dataset. Retrieved April 17, 2020 from https://www.yelp.com/dataset

[140] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *Proceedings of the IEEE 31st Computer Security Foundations Symposium (CSF'18).* IEEE, 268–282.

[141] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. 2015. Understanding neural networks through deep visualization. Retrieved from https://arXiv:1506.06579

[142] Honggang Yu, Kaichen Yang, Teng Zhang, Yun-Yun Tsai, Tsung-Yi Ho, and Yier Jin. 2020. CloudLeak: Large-scale deep learning models stealing through adversarial examples. In *Proceedings of the Network and Distributed Systems Security Symposium (NDSS'20).* The Internet Society. https://doi.org/10.14722/ndss.2020.24178

[143] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. 2019. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE International Conference on Computer Vision.* IEEE, 1476–1485.

[144] Ning Zhang, Manohar Paluri, Yaniv Taigman, Rob Fergus, and Lubomir Bourdev. 2015. Beyond frontal faces: Improving person recognition using multiple cues. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* IEEE, 4804–4813.

[145] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. 2020. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* IEEE, 253–261.

[146] Zhikun Zhang, Min Chen, Michael Backes, Yun Shen, and Yang Zhang. 2022. Inference attacks against graph neural networks. In *Proceedings of the 31st USENIX Security Symposium (USENIX Security'22).* USENIX Association.

[147] Huadi Zheng, Qingqing Ye, Haibo Hu, Chengfang Fang, and Jie Shi. 2019. BDPL: A boundary differentially private layer against machine learning model extraction attacks. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS'19).* Springer International Publishing, Cham, 66–83.

[148] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. *Deep Leakage from Gradients.* Curran Associates.

[149] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision.* 19–27.

Received 28 July 2020; revised 18 May 2022; accepted 2 September 2023