Survey: Leakage and Privacy at Inference Time

Citation for published version:
Jegorova, M, Kaul, C, Mayor, C, O'Neil, AQ, Weir, A, Murray-Smith, R & Tsaftaris, SA 2021 'Survey: Leakage and Privacy at Inference Time' ArXiv. https://doi.org/10.48550/arXiv.2107.01614

Digital Object Identifier (DOI):
10.48550/arXiv.2107.01614

Survey: Leakage and Privacy at Inference Time

Marija Jegorova, Chaitanya Kaul, Charlie Mayor, Alison Q. O'Neil, Alexander Weir, Roderick Murray-Smith, and Sotirios A. Tsaftaris

Abstract—Leakage of data from publicly available Machine Learning (ML) models is an area of growing significance as commercial and government applications of ML can draw on multiple sources of data, potentially including users' and clients' sensitive data. We provide a comprehensive survey of contemporary advances on several fronts, covering involuntary data leakage which is natural to ML models, potential malevolent leakage which is caused by privacy attacks, and currently available defence mechanisms. We focus on inference-time leakage, as the most likely scenario for publicly available models. We first discuss what leakage is in the context of different data, tasks, and model architectures. We then propose a taxonomy across involuntary and malevolent leakage, available defences, followed by the currently available assessment metrics and applications. We conclude with outstanding challenges and open questions, outlining some promising directions for future research.

Index Terms—Data Leakage, Privacy, Inference-Time Attacks, Privacy Attacks and Defences, Feature Leakage, Membership Inference, Property Inference, Machine Unlearning, Data Anonymization, Adversarial Defences

1 Introduction

Machine Learning (ML) technologies have become prolific in modern day life, with many ML models made publicly available. Data leakage is an area of growing significance as commercial and government applications of ML can draw on multiple sources of data, potentially including users' and clients' sensitive data. Hence, it is important to understand the potential leakage scenarios and existing prevention mechanisms in order to safeguard against revealing information about models' training data, in particular data which breaches an individual's privacy.

To address this need, we present a comprehensive overview and unified perspective on data leakage in trained ML models: causes of involuntary leakage, the implications of these causes being exploited by malevolent users, the methods for measuring and preventing such attacks, and finally to identify challenges and opportunities for further research into data leakage. To the best of our knowledge, existing surveys on privacy focus on privacy attacks or some subset of them [1–10], whereas we examine the broader picture of data leakage. Since the interest of this survey lies primarily in data leakage from trained models, we review research focused on inference time leakage and attacks (see Figure 1). If the reader would like to examine training time interventions, there are a number of relevant surveys [4, 11–14]. Our contributions are as follows:

- *first comprehensive survey on data leakage*, including involuntary and malevolent leakage methodology, prevention and defences, assessment metrics, and applications
- Acknowledging that leakage is context-specific, we describe the data leakage research conducted in different task and data type contexts.
- in-depth presentation of current methodologies;
- summary of the challenges and open questions in the data leakage research field

M. Jegorova is now with Facebook FAIAR, London but work was completed when M. Jegorova was with the University of Edinburgh, UK. C. Kaul and R. Murray-Smith are with University of Glasgow. C. Mayor is with NHS Scotland. A. O'Neil and A. Weir are with Canon Medical Research Europe, Edinburgh, UK. S.A. Tsaftaris is with the University of Edinburgh, UK and The Alan Turing Institute, London, UK.
Manuscript revised May 17, 2021.

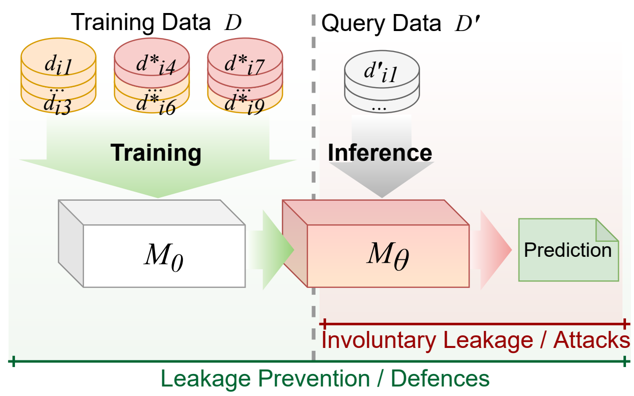

Fig. 1: This paper focuses on inference time involuntary leakage and attacks, concerning the query dataset D', trained model M_θ, its parameters θ, and output - predictions of M_θ or its confidence scores for the prediction options.

The paper is structured as follows: Section 2 provides definitions and notation, discusses what private and sensitive data are in the context of different data types, ML tasks and models, and defines leakage with respect to the actions of the user. Section 3 covers causes of natural *involuntary* data leakage, whilst Section 4 covers *malevolent* leakage, i.e., *privacy attacks*. Section 5 covers leakage prevention and defence mechanisms. Section 6 provides an aggregated picture of currently available metrics for assessing leakage and privacy. Section 7 outlines applications, such as Machine Learning as a Service (MLaaS), Data as a Service (DaaS), and mobile applications. Finally, we end with Sections 8 and 9 on remaining challenges, open questions, and conclusions.

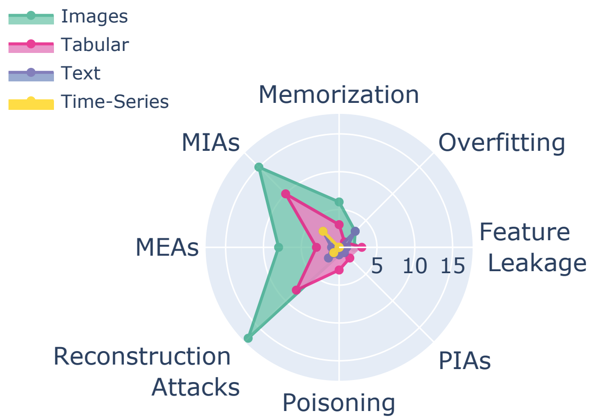

Fig. 2: Statistics on papers about data leakage, by data type of training dataset D and by type of leakage. It can be seen that imaging and time-series data are the most and least explored domains respectively. In terms of leakage types, current research is mostly focused around MIAs and Reconstruction Attacks (see Sec. 4.2 and 4.6).

2 Definitions

First of all let us define the notation for this article. Every ML model M_θ, regardless of its task, is trained on some data D, which consists of the individual data samples d_i which have features f_j, some of which are sensitive f_j^* ($i = 1, ...k$, where k is the number of data samples in D, and $j = 1, ..., n$, where n is number of features of D). With respect to this notation:

- Data leakage, differential privacy, membership inference attacks, and data reconstruction attacks, are all focused around the safety of the individual data samples d_i, i.e. the possibility of inferring these training samples from the model M_θ.
- Feature leakage and property inference attacks are concerned with inferring some properties of the sensitive features f_j^* of the training dataset D.
- Model extraction attacks are interested in inferring the parameters θ of the trained model M_θ (or their feasible approximations) in order to steal this model.

The end-user can have different levels of access to the trained model M_θ. Traditionally, these are separated into black- and white-box access. **Black-box access**, also known as **query access**, assumes that the user controls the input and has access to the output of the trained model M_θ. **White-box access** assumes that the user has full access to the trained model M_θ, including its input, output, architecture, and parameters θ. **Gray-box access** describes situations which are in between, e.g. user might not know the model's architecture and parameters θ but has access to outputs from the model's intermediate layers, or might not know the parameters θ, but has access to the architecture of the model M_θ, and so on.

2.1 What is personal (private) and sensitive data?

First of all, we distinguish between *personal*, *personal "sensitive"*, and *non-personal* data.

Personal data are defined in the Article $4(1)$ of the GDPR, [15], and, in loose terms, means data that directly or indirectly relates to an identified or identifiable natural person. Personal data may be collected routinely for legitimate ends. For example, a National Health Service (NHS) patient will have many examples of personal data processed during routine interactions with the healthcare system such as attendance at an appointment or address details which are registered with a GP.

Sensitive data are defined by the GDPR as the personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs; trade-union membership; genetic data, biometric data processed solely to identify a human being; health-related data; or data concerning a person's sex life or sexual orientation. Such data require stronger safeguards for data processing, storage, transfer, etc.

Non-personal data Any personal data is under GDPR [15] protection, which implies a dichotomy – everything outside the scope of personal is *non-personal data*. Thus, non-personal data become of the utmost importance for any data-driven research, analysis, and commercial applications.

There are methods to separate personal from sensitive data. However, since researchers often use unconsented data, such as healthcare data, it is common to adopt a cautious approach and assume that all of the data provided for research, even anonymised, falls under the special *sensitive data* category, i.e., $f_j = f_j^*$ and $d_i = d_i^*$ for all j and i.

The challenge is to mitigate against the risk of leaking any type of personal sensitive data that could be directly linked back to a real individual's identity (at the training sample d_i level of granularity, such as client/patient record, isolated sensitive data entries, user profile information, etc). In fact, we can imagine scenarios where real data is leaked, but nothing can be linked back to a real person e.g. a list of postcodes. The real-world risk of linkage back to identity is complex, and depends on multiple factors including the frequency of data points, the size of the source dataset, the availability of public datasets to support general re-identification strategies, and public domain information that makes it easier to identify specific individuals.

2.2 Leakage for different data types

Types of data leakage are largely data-specific; we provide illustrative examples below. Fig. 2 shows the number of publications per leakage/attack type covering different data types.

Data leakage in text data Examples are individuals' names (users, clients, patients, security personnel, etc.), dates of birth, full postcodes, full or partial addresses, telephone numbers, unique identity numbers, and job titles. In the context of training ML models on such data, one can imagine a predictive model, leaking specific sensitive data entries, features or full data records when deployed, [16, 17].

Data leakage in images Examples are individuals' faces or other identifying features, or embedded disclosive metadata (e.g. words overlaid on images containing sensitive information). When training an ML model with sensitive image data, a generative model such as [18], trained, for instance, on X-rays with hand-written notes on them or recognisable, re-identifiable bone/denture implants, might occasionally reproduce an identifiable training image look-alike. Similar type of leakage could apply for other types

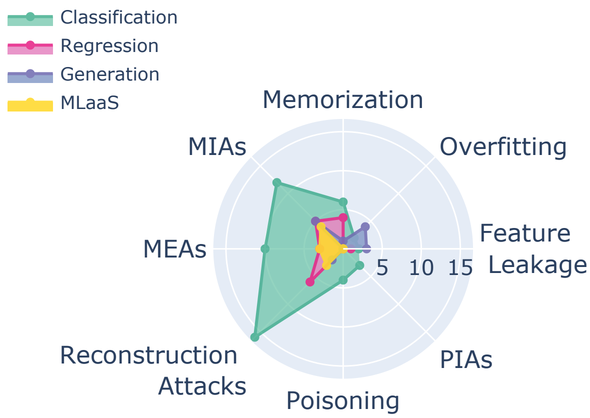

Fig. 3: Statistics on papers about types of data leakage across several ML tasks. Classification is the most explored type of ML task. Most current research is focused on MIAs, MEAs, and Reconstruction Attacks (Sec. 4.2, 4.3, 4.6). MLaaS (Sec. 7.2) can be of any type of task, but we would like to show attack distribution in the most practical setting.

of image text, such as personnel names on their security badges, car license plates, location postcodes.

Data leakage in tabular data Examples of information leakage are similar to text data; however in tabular data, the dataset is constrained to predefined variables and values, therefore the risk of identifying an individual can be more accurately estimated according to statistical disclosure risks, based on governing features such as the sensitivity of the tabular data, geography and population size, zero-value entries, and small group linkage to specific clinical providers (see [19]). Whilst, for instance, re-identifying patients from rare combinations of diagnoses is possible, statistical disclosure control, e.g. [19, 20], is relied upon to make such a possibility distant.

2.3 Leakage for different tasks

Privacy violations and mitigation of such violations is not only data-specific, but also task- and model-specific. Below is a detailed (but by no means exhaustive) overview of ML tasks and corresponding models for which some privacy-preservation and violation research has been conducted. Fig. 3 shows the number of publications per leakage/attack type with respect to some of the most common ML tasks.

Classification is widely used for real world applications such as retail goods databases [21], face recognition [22], autonomous target detection, and medical diagnostics [23]. Whilst it seems surprising that data samples might be reconstructed from as little information as a class label, there are a few methods – especially if more than plain black-box query access is possible – which will be described later: membership inference attacks (MIAs), property inference attacks, and model extraction.

Classification is the best researched task in terms of leakage and privacy attacks. A number of different kinds of attacks have been explored for image classification, on computer vision benchmarks like MNIST [17, 24–42], CIFAR-100 [17, 28–30, 43, 44] and ImageNet [31, 32, 45], as well as more applied datasets/tasks, such as classification of potential customer value [17, 41, 44], classification of the income level based on the Census data [24–26, 33], diagnosing breast cancer [24–26] and classifying X-rays [27, 46].

There has been less research on leakage from classifiers trained with tabular/mixed feature data (as opposed to image classification) [17, 24, 26, 28, 44, 47–51], and even less involving time-series classification. Notably, a number of works have targeted UCI's diabetes dataset [52], exploring predominantly model extraction attacks [24, 26, 49], and only touching upon binary classification of text data [50].

Regression / Prediction of unknown/future values of data samples has broad application in fields such as forecasting for financial and medical time-series, marketing trends, weather predictions, etc. A number of papers discuss model-level leakage for different sorts of data, including financial and medical time-series [24, 26, 49], numerical tabular data [29, 48, 49, 49], as well as mixed feature tabular data [29, 53].

Generation/Synthesis of realistic high quality data could solve the shortage of open-access data in the medical and financial domains. However, ensuring convincing privacy guarantees for generative methods is not a trivial demand [54, 55]. Part of the issue is that a good generative model, e.g. a well-trained generative adversarial network (GAN) [18], captures the underlying distribution of the real data, which means there is nothing stopping it from accidentally producing a doppelganger of a sensitive record (or a close enough sample), and simply sampling such a model could reveal much about both individual records and specific sensitive features of the training dataset [56, 57]. There are a number of linkage attacks specifically developed for GANs [35, 58–60], as well as some defences proposed for all data types (images, time-series, structured data) [56, 61–63].

Segmentation is important for computer vision tasks such as autonomous driving and medical imaging diagnostics. The privacy risks of sharing a medical image segmentation model publicly have been studied by [64] for linkage attacks, who showed that most state-of-the-art semantic segmentation models would be vulnerable. Segmentation models' vulnerability when less than white-box access is available remains unexplored.

Privacy preservation may also be important for other tasks such as clustering, translation, transfer, and collaborative learning. We observe that privacy-related research has been predominantly verified on image classification, and then somewhat on regression/prediction tasks, and very sparsely for any other group of tasks and models.

2.4 How do user actions affect leakage?

We differentiate between two types of user, defined below.

Passive / honest-but-curious user interacts with the trained model as intended by design and in compliance with protocols. All they can reveal is *involuntary / involuntary leakage*, if the model has any such vulnerability.

Malevolent user / an adversary attempts to take advantage of potential vulnerabilities in the trained model, such as memorization and overfitting, aiming to extract sensitive data via *privacy attacks*.

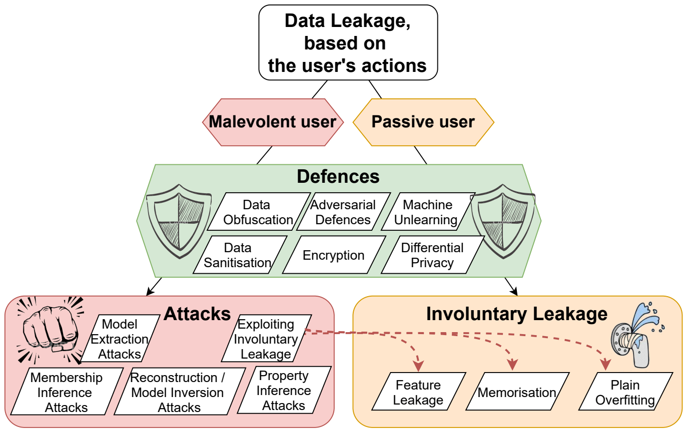

Fig. 4: Scope of the paper with respect to the user's actions. See Sec. 2.4 for more detail on the role of the user.

3 Involuntary Data Leakage

There are a few ways in which data can leak without any malicious intervention from the user, including memorization, overfitting, and feature leakage. *Feature leakage* (Sec. 3.3) is characterized by leakage of sensitive features f_j^* of the data D, whereas memorization (Sec. 3.1) and overfitting (Sec. 3.2) are mostly related to leakage of whole data samples d_i.

Although overfitting and memorization are both indicative of a lack of model generalization, it is important to understand the difference between them. *Overfitting and overtraining* [65], manifest in a trained model as higher accuracy on the training data than on the test data. This happens due to training an excessively complex model on a comparatively simple training data set, or training the model for too long, i.e., after the training loss function has converged. *Memorization* involves unintentionally storing training data samples "memorized" in the model parameters, with the potential to leak these at inference time. Whilst overfitting implies some degree of memorization, memorization can occur while the model is still learning, i.e., before the overfitting begins to happen [16].

3.1 Memorization

Memorization of specific training data samples occurs when the model assigns some sample a significantly higher likelihood than expected by random chance [16]. It raises serious privacy and legal concerns for sharing trained ML models publicly or providing them as a service [66, 67].

Measuring memorization In order to detect and prevent (or exploit) the memorization effect in trained models, one would need to first estimate it using one of the existing mechanisms. For example, a metric called *exposure*, proposed in [16, 68], aims to estimate a model's potential for memorizing rare and unique sequences in text data specifically. In a little more detail – [16] embeds the new unique sequences into the training text, which they call *canaries*, and later measure the probability of these canaries coming up at the inference time. Similarly, *déja vu* [69] focuses on estimating the memorization happening in the lower layers of convolutional neural networks, showing that in practice commonly applied fine-tuning of the upper layers of the neural networks is not enough to prevent memorization.

Potential risks of memorization are covered in detail in Section 4, and include membership inference attacks, sensitive attribute reconstruction, and even training dataset reconstructions, in the case of malicious intent from the ML-as-a-Service (MLaaS) provider [67].

Preventing memorization Although there is little research on explicitly preventing memorization, there is some evidence suggesting that data augmentation somewhat reduces (but does not eliminate) the memorization capacity of a network, whereas increasing the size of the architecture increases its memorization capacity [69]. More specifically for GANs [18, 70], [56] suggests that limiting the number of noise vectors at training time reduces memorization.

Nevertheless, as harmful as it is from the perspective of sensitive data leakage, some degree of memorization is necessary and unavoidable in certain scenarios. For instance, [57] builds on the premise that a well-trained generative adversarial network (GAN, [18, 70]) has to learn enough about the underlying training data distribution to function properly. It further utilizes that notion to memorize explicitly with a memory network module, which ensures stability of the training as well as better understanding of the separate distribution modes by unsupervised GANs.

3.2 Plain Overfitting

The hallmark of model overfitting is substantially higher accuracy on the training data than on the test data, usually caused by overtraining or unnecessarily large models being trained on smaller datasets [65]. The relationship between overfitting and privacy risks is not yet completely clear, due to the lack of research on exactly how overfitting aids various data and model attacks [29, 71].

Measuring and exploiting overfitting Overfitting has been shown to be a sufficient but not necessary condition for aiding membership inference and model inversion attacks (Sec. 4.2 and 4.6). According to [71], overfitting (high generalization error) inevitably results in a privacy loss for classification models – they formalize the connection between the inference advantage of the attacker model and the target model generalization error for both membership inference and attribute inference attacks (Sec. 4.2).

Preventing overfitting Preventing overfitting for most models (but not all GANs [18]) can be achieved through simply monitoring the generalization error. Nonetheless, overfitting is but one of the possible reasons for data leakage, and even stable, well-generalized models can leak sensitive data, e.g., due to memorization [16, 25, 44]. Specific model types and architectures, as well as the training dataset features also have an impact on leakage [17].

3.3 Feature Leakage

Feature leakage occurs when sensitive attributes/features f_j^* of the data D are unintentionally memorized and revealed by the trained model at inference time.

Measuring feature leakage. Explicit feature leakage through memorization is a concern for models which classify or predict natural language sequences. Hence, [16, 68] introduced an *exposure* metric (Sec. 3.1) and suggested training natural language models with this metric as a training guide to encourage training without memorization.

Another approach for estimating feature leakage is presented in [72], where they proposed a number of Bayesian metrics based on universally consistent nearest neighbor rules from which the metrics should be selected that converge fastest. This results in an estimate of the Bayes risk of the model in question, i.e. the error of the optimal (ideal world) classifier for predicting a sensitive attribute given an output observation from the model.

Potential Risks. Feature leakage implicitly enables property inference attacks (Sec. 4.4). For instance, [50] focuses primarily on leakage of *unintended* features, i.e. inferring properties that hold for some subset of the training data but not in general for the entire class, which are also not necessarily the properties that the target model intended to capture in the first place. They show that property inference attacks are a danger for collaborative learning models (Sec. 4.4 and 5.9).

Preventing feature leakage Interestingly, although perhaps not surprisingly, [73] discovers that *overlearning*, i.e. the model learning attributes that are not part of the original objective or that make it sensitive to certain biases, can have serious negative consequences, such as feature/attribute leakage, and the model capacity for being re-purposed for a privacy-violating agenda even in the absence of the original training data. Importantly, [73] also shows that overlearning cannot be prevented by merely censoring out the unnecessary attributes, meaning that certain defences, e.g., data obfuscation (Sec. 5.1) will not reliably prevent overlearning.

4 Malevolent Leakage / Privacy Attacks

To elaborate on Sec. 2.4, we define *malevolent leakage* (a term used interchangeably with *privacy attacks* in this survey) as the actions of a malevolent user, an adversary who tries to take an advantage of trained ML model M_θ, which we call the *target model*, at inference time.

In this section we assume that the adversary has no access either to the original training data or to the training process of the target model. We do not make any assumptions beyond this e.g. the adversary's access to the trained model M_θ can be either *black-box*, *white-box*, or anywhere in between. Note that some of the methods reviewed in this chapter also assume access to open-source data that might or might not come from a similar distribution to the original (potentially sensitive) training data.

4.1 Attacks Exploiting Memorization and Overfitting

This is not an explicit class of privacy attacks; rather, almost all methods of attack have a higher chance of success when overfitting comes into play, and several will implicitly exploit overfitting [74, 75]. A large amount of research has been conducted to show that overfitting alone is enough for membership inference attacks and more complex attribute inference attacks to succeed [29, 75] (see Sec. 4.2 for details).

Other examples of exploiting memorization and overfitting apply to settings such as *collaborative* (also known as *federated*) *learning* [50], where model gradient updates can be used by the adversary – the malicious participant – to leak sensitive information. Since the adversary provides part of the training data for the target model, the inference attacks (Sec. 4.2) are simplified to a supervised learning problem, i.e., poisoning attacks (Sec. 4.5).

Another malicious setting, explored by [67], features an adversary model provider (DaaS setting, Sec. 7.1), supplying the model M to a data owner, and receiving back a trained model M_θ. Model architectures designed by [67], could deliberately memorize the original training data, while maintaining reasonable performance on tasks like face recognition, image classification, and text analysis, even without the adversary directly controlling the training.

4.2 Membership Inference Attacks

ML models currently do not fall under GDPR protection. Nonetheless, advances in certain types of attack, such as *membership inference attacks* (MIAs, also sometimes called *"linkage attacks"*) and *reconstruction MIAs*[2] can be used to identify the individual records used for training open-access ML models (see Fig. 5). Hence, MIAs can threaten user data privacy, supporting the argument that ML models should be classified according to their sensitive training data content [66].

Formalization: An MIA is a type of attack (lying anywhere in the range from white-box to black-box), that assumes the attacker has access to both:

- *the trained target model* M_θ – the more information about the model that is available, the easier to attack. The adversary must at least have query access.
- *some query dataset* D' – ideally containing the training data samples d_i, that have potentially been used for training the target model M_θ, i.e. $d_i \in D$ (as well as $d_i \in D'$). The adversary must at least have a dataset containing samples d_i similar in distribution to those in the original training dataset D.

The target of MIA is to re-identify which of the samples d_i were used for the training of the target model M_θ.

Risks of query access to M_θ. While large companies take advantage of their user databases and deploy ML models on a large scale, there is always a risk of re-identification or misidentification of a user, given (even just query) access to the model. Offering ML as a service, i.e. providing the trained models in open and semi-open access, increases such risks. Vulnerability under MIAs is largely data-driven and hence data-specific. Also, it can be performed with just black-box access to the model [45], and without knowledge about the structure of the target model [28, 41].

MIAs and Overfitting. In addition to direct information about the model type, architecture, or parameter values (black- vs white-box MIAs), overfitting and poor generalization can significantly impact the vulnerability of a model. In fact, MIAs are likely to succeed on an overfitted model even with only black-box access. For larger class-balanced multi-class datasets, [17] reports over 70% attack accuracy for model overfit to a train-test accuracy gap of over 12%, and up to 100% attack accuracy for over 25% gap. Further, [29] and [25] provide theoretical and empirical evidence that

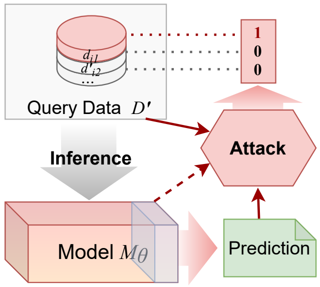

Fig. 5: **Membership inference attack (MIA):** at inference time the adversary uses query-prediction pairs obtained from M_θ to train an attack model capable of telling whether the data sample d_i has been included in the original data set D or not - correspondingly labelling them as 1 or 0.

overfitting alone is sufficient to increase the attacker's success in performing MIAs. The same is proven by [29] for the *attribute inference*[1] attacks, however controlling overfitting (by minimizing generalization error) does not necessarily prevent a successful membership inference. Furthermore, [25] presents strategies for attacking well-generalised models via identifying the vulnerable target records and exploiting their influences on the target model. Finally, [29] shows that the possibility of attribute inference implies the possibility of membership inference, thereby making a connection between MIAs and reconstruction attacks (see Sec. 4.6).

Potential defences and concerns. There are a variety of strategies that are usually proposed for defence against MIAs, such as differential privacy in a federated learning setting (Sec. 5.6 and 5.9), however none provide absolute guarantees. For instance, [40] systematically explores attacks on differentially private models, and finds that differentially private models can provide strong privacy guarantees at the expense of model utility. Furthermore, [44] shows that adversarial participants can run (white-box) MIAs against other participants in a federated learning setting even on well-generalized models, which is also evaluated for different levels of knowledge and interventions available to the adversary, coming to conclusions that gradients in the later layers of the neural nets leak more information compared to the earlier layers, that gradient norm highly correlates with the accuracy of membership inference, and that, predictably, increasing the size of the adversary's training dataset increases the precision of the attacks. Finally, [28] goes further and relaxes the assumptions on both the adversary and the data, showing that MIAs can succeed even without knowledge of the target model structure and without assuming that the query and original training datasets should come from the same or similar distributions using just the posterior $M_\theta(d_i)$ of the target point d_i and the empirically chosen threshold (based on attackers' priorities and query datasets available).

Alternatively, there is a class of adversarial defences (Sec. 5.5), that use potential attack models as a penalty when training the target model M_θ [76]. However, they should be used with great caution as [77] has shown that training with some of the state-of-the-art adversarial defense approaches will make the target ML model more susceptible to MIAs, compared to the original undefended training strategy.

Applications of MIAs are numerous, both in terms of the types of the data and the models on which they have been shown to succeed. Applications which have been explored include: medical data [78], location data [79], including time-series [80], translation systems [81], collaborative learning (especially the case where the adversary performs as one of the participants) [36, 41], and generative models for various types of the data synthesis. The latter are usually GAN-based, in which the discriminator of a *shadow model*[2] is often used for re-identifying the original training data samples in the query dataset [58–60].

Measuring the success of MIAs is easy compared to other privacy attacks. The common metric to use is re-identification score, i.e. the ratio of training/additional data samples in the query dataset, that have been correctly identified by an MIA, or some modification of this metric [82, 83].

4.3 Model Extraction Attacks

Model Extraction Attacks (MEAs) are not designed to steal the training data D (although it is often a by-product of this class of attacks [84]); instead their end-goal is to steal the trained model functionality, see Fig. 6.

Formalization of assumptions for different kinds of MEAs. Model functionality can be captured in a few ways. From most to least prior knowledge required, MEAs can:

1) steal just the (trained) model parameters θ, assuming the model architecture (or at least the class) is known to the attacker,
2) steal *the entire model architecture* M_θ when it is unknown – a black-box-style model extraction attack.
3) steal the model functionality – an extraction attack does not necessarily have to reverse engineer the target model itself. It might be enough to copy the functionality of it, e.g. *make a different model* $M^*_{\theta^*}$, *where* $M^*_{\theta^*}(x) \approx M_\theta(x)$, where x is some data plausible for a task domain at the inference time. This class of techniques can succeed without any assumptions on the model architecture or anything else except query access to the target model.

Below we present more detail on the above, ordering from greatest to least stringent requirements for the attacker's prior knowledge.

1) Stealing parameters θ and hyperparameters θ' of the ML models of the known class. This setting assumes that an attacker is in possession of the most granular level of knowledge about the target model M_θ across all ME types.

For instance, [49] assumes full white-box access to M_θ, i.e. Machine Learning as a Service setting (MLaaS), where the adversary knows everything: the original training dataset D, the ML algorithm (an objective function) of the target model, and (optionally) the learned parameters of

1. *Attribute inference attack* (or *reconstruction attack*) assumes access to the trained ML model and incomplete information about a data point, and aims to infer the missing information about that point [53].

2. *Shadow model* is a term used in privacy attacks, in which a new model is trained by an adversary to mimic the behaviour of the target model, based on its query-output pairs.

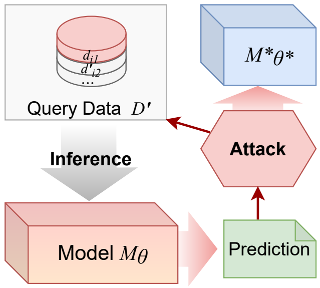

Fig. 6: **Model Extraction Attack (MEA):** at inference time the adversary uses query-prediction pairs either to train an attack model $M^*_{\theta^*}$ functioning identically to the target model M_θ, or to reveal the (hyper-)parameters θ of M_θ. A doppelganger of M_θ then can be attacked for the original training dataset D, e.g., with MIAs or reconstruction attacks.

the target model θ. Under these assumptions, a method is proposed for efficiently stealing the hyperparameters θ' of the target models with both theoretical assessment and empirical evaluation on Amazon Machine Learning service.

A black-box attack, stealing parameters θ, is possible without access to the original training data D, assuming knowledge about the model class, the confidence values provided as an output of the target model, and/or the ability to query arbitrary partial inputs. There are two efficient ways of stealing a trained model's hyperparameters with aforementioned assumptions introduced by [26]. These attacks are also successful when the confidence values are omitted from the target model output, as a privacy precaution. The reported speeds of extraction of the 100%-equivalent of the trained models from publicly available services, Amazon ML and BigML, (for logistic regression and decision tree target models), is between just over a minute to just over half an hour [26].

2) Reverse engineering black-box models or functionally equivalent model extraction. In this case the assumption of an adversary knowing the model architecture is relaxed, which makes the extraction attack much harder but not impossible [31, 84]. Additionally, there is still an implicit assumption that the adversary has access to some suitable unlabelled data for querying the target model, not necessarily from the same domain as the original training data, but from a rich enough distribution to expose the full target model functionality.

An intuitive approach in this setting, based on creating an imposter dataset D' and then training a functional equivalent $M^*_{\theta^*}$ of the target model M_θ on it, is offered by both [84] and [85]. Both papers query the target model (black-box CNN) M_θ with some random unlabelled data D', asking the target model itself to label the new dataset. This results in an imposter dataset D', theoretically containing the knowledge of the target network $M_\theta{}'$ the "copycat" network $M^*_{\theta^*}$ is then trained on this imposter dataset D', and should be able to reproduce the behaviour of the target model M_θ, i.e., $M^*_{\theta^*}(x) \approx M_\theta(x)$, where x is some data plausible for a task domain. The empirical results of [84] (for CNN class models) show at least 93.7% attack accuracy on a variety of problems (measured as the ability perform in the same way as the target model), and 97.3% of the performance when applied to the Microsoft Azure Emotion API. [85] shows between 92% and 105% performance of the target model. They explain the additional improvement on the target model by the regularizing effect of training on soft-labels, introduced as the "soft targets" in [86].

3) Stealing functionality with minimal assumptions. The next assumption to relax is access to the unlabeled data used for querying the target models. [32] assumes no prior data knowledge, as well as no knowledge of the target model class. Instead, they train a meta-model capable of inferring the target model architecture and training hyperparameters (such as the optimization algorithm and the training dataset) from a series of queries, hence turning the black-box target models into white-box models, which automatically makes the target models susceptible to all of the above mentioned attacks.

Last but not least amongst the minimal assumption methods, [31] explores the trade-off between accuracy and fidelity of MEAs, where accuracy stands for performing well on the underlying task, and fidelity for matching the target model predictions. They focus on high-fidelity, and claim the first practical functionally-equivalent model extraction, i.e. $M^*_{\theta^*}(x) = M_\theta(x)$, as well as faster querying, compared to competitors. This is achieved by a learning-based attack method, that utilizes the target model as an oracle for training the adversary model.

Model Extraction for some more specific applications. An important limitation of all of the aforementioned MEA-related research is that it focuses primarily on classification and prediction tasks. However, there are other interesting applications in the field. For example [38] investigates model extraction attacks in a setting where the target model provides not only traditional outputs, but the gradients with respect to the input data as an explanation for its outputs. Active learning for model extraction in MLaaS settings is covered by [24], both for implementing model extraction attacks and investigating possible defences. In fact they find that active learning is very similar to MEAs. There is also some exciting research on model extraction of natural language models, such as BERT [87], which finds not only that simple query access to the target model is sufficient, but also that no real or semantically plausible data is required for querying the target model. Random sequence querying paired with a task-specific heuristic is enough for extracting approximate models for natural language inference and question answering.

Model extraction for generative models remains unexplored. One can argue that a principle similar to [85] and [84] could work, i.e. sampling the target model for random inputs (for instance conditions for the generator in GANs) in order to create a fake dataset for training a functionally identical model. However, to our knowledge there is no published work confirming it in practice.

Defences against model extraction attacks A number of precautions can be taken in order to protect a model from MEAs, e.g., [26] originally suggested anything from rounding confidence scores outputs, to differential privacy, and/or using the ensemble methods. However, the efficiency of these would be model-specific, and they do not guarantee complete safety from the model extraction attacks. Addi-

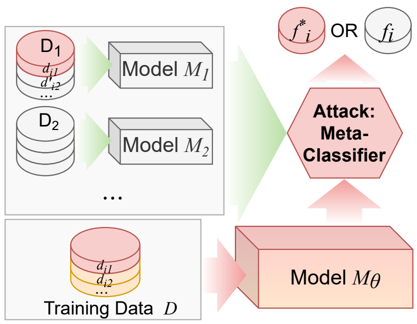

Fig. 7: **Property Inference Attack (PIA):** the adversary aims to train a meta-classifier to detect a sensitive feature f_i^* in a target model M_θ. To achieve that the adversary trains multiple shadow models on datasets with an without proxy sensitive features, using that as training data for the meta-classifier. The trained meta-classifier should be able to detect the presence of sensitive features f_i^* in the target model M_θ.

tionally, there is a more active way of defending from model stealing via open-access APIs called PRADA, [37]. It analyzes the distribution of consecutive API queries and raises an alarm when this distribution deviates from benevolent behaviour. According to [37], PRADA can detect all prior model extraction attacks with no false positives.

4.4 Property Inference Attacks

Property Inference Attacks (PIAs) can be seen as a sub-class of model extraction attacks, where instead of trying to learn all the attributes of a model, an attacker tries to extract a specific sensitive attribute or feature of interest f_i^* from a given target model M_θ. The overall structure of a Property Inference Attack is shown in Fig. 7.

Formalization of assumptions PIAs are generally white-box attribute inference attacks, that assume complete access to the target model, including its training information, model weights, etc. Property Inference Attacks can be formalized as follows,

- PIAs are based on the principle that similar models, trained on similar datasets, exhibit similar properties.
- The goal of PIA is to build a meta-classifier, MC, that is capable of classifying whether a ML classifier, M_i, contains the sensitive attribute of interest, f_i^*, or not.
- In order to train the MC, an attacker trains a series of Shadow Classifiers, $M = \{M_1, M_2, .., M_n\}$ on some shadow dataset, $D = \{D_1, D_2, .., D_n\}$, where only some of the subsets of D exhibit the property f_i^*.
- The shadow models are not explicitly trained to learn the property P, but learn it as a consequence of the bias introduced in the dataset.
- During inference, the target model, M_θ, trained on the original dataset, D_x, is passed into the MC, that classifies it as either exhibiting f_i^* or not.
- Generally, the weights and biases of the models are used as the features to train the meta-classifier.

The first work on PIA conducted the attack successfully on Hidden Markov Models and Support Vector Machines [51]. The weights of the hidden states were used as the inputs for the HMMs while the weights and biases of

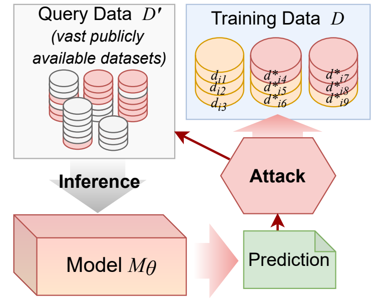

Fig. 8: **Reconstruction / Model Inversion Attacks:** at inference time the adversary leverages query access to the target model M_θ and to some aggregated publicly available data D' in order to reconstruct a private training dataset D, containing sensitive data d_i^* and sensitive features f_i^*.

the support vectors were used to train the meta-classifier for SVMs. The logical transition of this approach to fully connected networks was shown in [33], where the weights and biases of the neural networks were used as input to the meta-classifier. To account for the permutation invariance in the representations learnt by neural network nodes, the meta-classifier itself was a network that learnt to account for all permutations of a particular neural network layer's weights and biases. An architecture inspired by Deep Sets [88] was used for this task.

Applications of PIAs are so far somewhat limited - no existing approach has applied PIAs to models beyond fully connected neural networks. Moreover, no publications show PIAs applied to anything but classification tasks.

4.5 Poisoning Attacks

Depending on access to a target model M_θ, data D, training objective and other parameters θ, an adversary can potentially pollute the data D or the model M causing a bias in the target model output, that an adversary can use to their advantage. This class of attacks is called poisoning attacks.

Poisoning attacks are mostly done at model level [89–91]. Federated Learning models, for example, are often susceptible to such attacks due to the nature of their training algorithms. However, training time model manipulations by adversary are out of the scope of this survey; we are only interested in inference time leakage by poisoning attacks.

Data poisoning has also been used to increase the amount of sensitive information a model leaks about a particular sensitive attribute. After choosing the property to attack, an adversary submits the input data according to the poisoned data distribution. Given only black-box access (output labels) to a model, the adversary can then infer the frequency of the sensitive feature in the dataset [92].

4.6 Reconstruction / Model Inversion Attacks

Reconstruction / Model Inversion Attacks are a collection of methods for partially reconstructing a private dataset from aggregated publicly available information, which may include publicly-available or query-only trained ML models. See Figure 8.

Applications. Reconstruction attacks have been applied to a variety of scenarios, for instance, to the *federated learning* *setting*, [93], including an interesting application of GANs trained with a multitask discriminator that outputs the reality indicator for the data, its class, and user identity, [94].

A variety of applications of model inversion exist in the *general (centralized) setting*, e.g., [95] introduces a technique, where the second neural network is trained as an inverse of the target model to perform the inversion, with its performance validated on Amazon Rekognition (MLaaS setting). Yet another interesting example of GANs used for inversion attacks, called *generative model-inversion attack*, was proposed by [96]. It applies GANs to learn the distributional prior of the data, which later guides the inversion process. Finally, [97] explores the *Deep Leakage from Gradient*, an incredibly efficient inversion attack, accurate to the pixel-level for images and token-level for natural language, proving gradients of the model are unsafe to share publicly.

Reconstruction attacks in an online learning setting have been studied in depth in [98]. In this case, the adversary probes a model with a particular data point (*MIA*, Sec. 4.2) or a particular set of data points (*Group MIA*) before and after training the model with the additional data, in order to assess how the model's outcome changes as a result of the online training. The attacks proposed in [98] follow a general encoder-decoder structure. The encoder attempts to learn the target model's difference in prediction before and after being updated with the additional data. The decoder then generates information on the updated dataset.

Defences. Several defences have been proposed against reconstruction attacks: [99] suggested the "noise interference" technique, which can render an invertible model non-invertible by adding noise. Another noise-based defence, this time for the federated learning setting, has been recently proposed by [100]. They use a simple additive noise method and, interestingly, they find that pairing it with another existing method NoPeekNN, [101], improves the defence. For classifier target models, [102] suggests "purifying" the confidence score vectors of the target model by reducing their dispersion. This can help, since some of the MIAs and some of the reconstruction attacks use the target model confidence score vectors for guidance.

5 Current Methods of Defence

Defence methods aim to prevent privacy attacks launched by adversaries from succeeding. There are a few stages in model training and deployment where defences can be implemented. They can largely be dichotomized as applying some augmentations at the *training data level* versus training, tuning, and designing the models with inbuilt defence mechanisms – at the *model level*.

At the data level, simply deleting the sensitive features or entries is unwise for training data integrity and consistency reasons, and can represent a privacy risk of its own, as the pattern of "missingness" itself might allow for the inference of some of the data properties. Hence techniques such as data obfuscation and data sanitization are often applied in order to mask, scramble, or overwrite the sensitive information with a realistic fake rather than using simple deletion.

5.1 At Data Level: Data Obfuscation

Data obfuscation perturbs the sensitive information in the data through either *scrambling or masking* of some sort.

For instance, [103] introduces an obfuscation function to pre-process data before submitting it for training to a downstream model. This is achieved by adding noise to the data or augmenting it with the new data samples. It addresses the trade-off between user privacy and service quality, which is dependent on the severity of the data perturbation. They build adaptive mechanisms that anticipate and protect against optimal inference algorithms by designing a game between the designer of the obfuscation mechanism and the potential inference attack. Meanwhile, [104] is concerned with the difference between a trained model's predictions on training and test data and the inference risks this difference presents. They suggest mitigation of those risks by narrowing the dynamic ranges of the sensitive features in the training data, such that the training, test, and synthetic data are forced to have similar predictions by the same ML model.

5.2 At Data Level: Data Sanitization

Data Sanitization aims to disguise the sensitive information within the data by overwriting it with realistic-looking synthetic data, using techniques like flipping labels or adding noise of certain specifications. Recent developments also include, for instance, [105] randomization algorithms satisfying the ϵ-differential data privacy criteria. Data sanitization is often a natural precaution for adversarial attacks [106] (adversarial attacks are a large class of training time attacks, which are outside of the scope of this paper).

Limitations The aforementioned data modifications are limited by the assumptions made about data complexity.

Applications Sanitization is widely applied to social networks' privacy. For instance, [107] argues that nouns convey most of the information in a sentence, hence sanitization can be conducted by treating nouns in the sensitive sentences as keywords that need overwriting with random entries. Sanitization is a potential defence against the inference attacks on the social media networks, e.g., [108] utilizes a collective manipulation sanitization techniques on the user profile and friend connection data to prevent inference attacks from successfully identifying social network users from the open-source data using their friend connections.

Further sanitization implementations include the self-destruct data-processing cycles proposed in [109]. These overwrite data enough times, using threshold cryptography, to render it non-recoverable and hence ensure user data self-erase after a certain validity period.

5.3 At Data Level: Learning with Synthetic Data

Learning with synthetic data can be viewed as a natural extension to both data obfuscation and sanitization, since it involves perturbing/disguising the sensitive information. High-fidelity synthetic data, generated with privacy guarantees, could potentially solve a number of problems for training ML models across a wide variety of applications, from healthcare to financial data analysis. It would not only allow open access to realistic synthetic data for ML researchers, but could also facilitate the internal data transfers within the organizations, in situations where clients/patients data cannot be shared across the branches or divisions of a company, or across different hospitals, etc.

Applications There is evidence suggesting some already successful applications - generating high quality synthetic patient data [54] for testing ML healthcare software, using a combination of techniques including probabilistic graphical modelling. Another potentially useful approach is data synthesis via a differentially private autoencoder with empirical assessment of both the utility and quality of the results [110].

There are several GAN-based models that are designed to produce synthetic data with certain privacy guarantees. For example, [111], meant for generation of time-series type of data with DP guarantees, and [112] designed to preserve privacy under MIAs at a small performance trade-off.

Limitations and Risks Although synthetic data might seem appealing as a remedy for the sensitive data leakage problem, it is not the case in reality, largely because a good generative model captures the underlying training data distribution, and might leak at the very least some of the properties of the dataset into its generated data, enabling property inference attacks. Moreover, [55] finds that generative models tend to store richer information, enabling attribute inference. Furthermore, they show that generative models are vulnerable under membership inference attacks (sometimes called "linkage attacks"), even when trained under differential privacy guarantees, perhaps because memorization issues in models like GANs and VAEs cannot be fully eradicated.

Thus far synthetic data generation with privacy guarantees remains elusive.

5.4 At Model Level: Machine Unlearning / Forgetting

The General Data Protection Regulation (GDPR), [15], enforced by the European Union in May 2018, is aimed at protecting user privacy. Amongst other things, GDPR ensures the user's right "for the explanation" about how their data are being stored and used, as well as the right "to be forgotten", i.e. a user can request their data to be deleted from a database. The natural next question: *What if these data also had to be "forgotten" by the AI models powering a service?*

The obvious course of action would be to remove the user data that needs to be forgotten from the training dataset and retrain the model from scratch. However, often the computational costs involved would make this an infeasible solution, creating a demand for techniques to unlearn the requested data and its traces from the trained models.

There are other use cases for the ability to unlearn certain data, irrespective of the user's privacy. For instance, in the case of adversarial attacks (outside the scope of this survey, but relevant to machine unlearning), if an adversary pollutes the model (e.g., anomaly detector) with hand-crafted faulty data, the model might have to "forget"/"unlearn" such data in order to restore its security [113].

Machine Unlearning. This term was introduced by [113] who proposed the need for a "forgetting system", and introduced one of the first unlearning algorithms based on converting learning algorithms into *summation form*[3] for efficiently forgetting data traces. This method also works against data pollution attacks. [114] is the first framework for instantaneous data summarization with machine unlearning using a resilient streaming algorithm, involving submodular optimization; it comes with a constant factor approximation guarantee to the optimum solution. [115] provides formalization for machine unlearning in a variety of instances, and proposes an efficient unlearning algorithm for k-means clustering, with accompanying statistical analysis of the results. [116] provides an unlearning algorithm for linear regression methods, based on the projective residual update and use of synthetic data points. [117] proposes to limit the effect that a single training data sample can have in the training process. They achieve this by training multiple models on subsets of the training dataset, which would imply storage and computational costs for retraining multiple models. In a similar attempt to limit the effect of a single data point at training time, [118] and [119] suggest a Newton-based estimation of the effect of such a training point on the model predictions. This estimate can be immediately used for guiding the machine unlearning.

A comparatively computationally light method [120] suggests forgetting logit-based classifiers through linear transformation to the output logits. This method, however would leave a data sample trace in the weights of a neural network model. [121] focuses on data removal from the differential privacy perspective, and provides an algorithm for convex problems, based on a second order Newton update, to be layered over a differentially private DNN.

An algorithm proposed by [122] conducts unlearning for DNNs trained with SGD, and is based on shifting the weight space of the model by adding noise to the weights. Specifically, [122] focuses on selective forgetting by "scrubbing" the weights of the neural net, so that it need not be trained from scratch, without requiring the access of data to be forgotten. Further, [123] proposes weight scrubbing based on the Neural Tangent Kernel at the level of the model activations, which allows not only better handling of the null-spaces in network weights (which is essential for over-parameterised models like DNNs), but also for the "one-shot" forgetting to work better than [122]. This work also introduces a new set of bounds that quantifies the average information per query an attacker can extract from the model.

Verifying forgetting. There is a difference between deliberately unlearning the traces of information from an ML system versus verifying it has indeed been forgotten (intentionally or otherwise). There are additional considerations also: *1)* Forgetting can occasionally happen on its own (such as *"catastrophic forgetting"* in reinforcement learning); *2)* different data samples bear varying amounts of unique information and contribute to the model final weights differently [124]; *3)* forgetting a specific data entry (a single person's entry) in the training set and consequently its trace in the

3. *The summation form* is a technique where model weights are not trained on each data sample, instead they are trained on a small number of sums of the data sample transforms. Aforementioned transforms are achieved through pre-defined efficiently computable transformation functions. When the data sample is erased, these sums get re-computed, and the model is efficiently updated.

system is non-trivial, because of the possible *trace overlap*. In the case of trace overlap, the updates (the knowledge) that the system extracted from the entry to-be-forgotten are exactly the same as some knowledge obtained from a similar record that is still a legitimate training data point, and hence this knowledge should not be forgotten. In light of GDPR [15], and the "right to be forgotten", there is a lot of focus on formalization and a good way of performing verification of forgetting [125].

These intricacies have led to several directions of verifying forgetting. For instance, [126] develops a forgetting verification technique based on backdoor attacks (data poisoning attacks), i.e. some fraction of users of MLaaS can choose to insert backdoor triggers in a fraction of their data; training MLaaS on such data will have a high backdoor success rate. So after the user's request to be forgotten they could rely on a simple check of the backdoor success rate to verify whether that has been done. [127] focuses primarily on applying statistical methods, i.e. Kolmogorov-Smirnov distance, to find a discrepancy in the output distributions between a model that has supposedly "forgotten" certain traces and a reference shadow model [4], trained on different datasets to model forgetting with and without a *trace overlap*.

In case if the "core" dataset, that should not be forgotten, is known, [128] offers an effective method of forgetting the traces of the additional data, that involves replacing a standard deep network with a suitable linear approximation.

There are also plenty of context-specific applications, including forgetting data for neural network predictors [129] by applying carefully engineered oblivious protocols for commonly used neural network operations on trained networks. For network embeddings [130] investigates the forgetting of a single node, by removing the representation vector from the network embedding, and finds that often this is not sufficient since the information can be still encoded in the embedding vectors of the remaining nodes. And finally, for text generation models [131] suggests at black-box model-auditing technique successful on well-generalized models that are not overfitted to their training data, and [132] proposes a model-auditing method based on the model distillation and model comparison techniques.

Limitations and Risks. Despite the benevolent intentions of machine unlearning, it should be applied with caution due to the risks involved. For example, [133] focuses on analyzing the risks of data leakage (through MIA) for black-box classifiers, that has been through the machine unlearning procedure. They find that in some cases the *unlearnt* model can leak information about the forgotten data, even when the original *non-unlearnt* model did not leak information.

5.5 At Model Level: Adversarial Defences

Adversarial Defences use an adversary as a penalty during training of the target model M_θ. Although in theory most privacy attacks can be used in some way during the training of M_θ as potential adversaries to defend against, in practice this setting has been mostly explored for MIAs.

Adversarial Defences for MIAs. There has been a lot of research conducted on protecting against black-box MIAs with adversarial examples. For example, [76] anticipates a MIA, and regularizes the target model during the training via min-max game-based adversarial regularization, so that predictions of the target model on its training data are indistinguishable from its predictions on other data points from the same distribution. This technique not only claims membership privacy, but also – good target model generalization. Memguard, [134], has been the first defence with formal utility-loss guarantees against black-box MIAs. Instead of fiddling with target model regularization, like [76] does, it proposed adding carefully designed noise to the target model confidence score vectors, turning these into adversarial examples, that a MIA classifier would be vulnerable to. [134]

Limitations and Risks. Interestingly, some of the proposed adversarial defence methods, such as projective gradient descent (PGD) adversarial training [135], on the contrary increase the model's susceptibility to membership inference attacks.

Theoretically, many of the privacy attacks could be used as potential adversaries to improve against during training of M_θ. Nonetheless, one has to be careful with this setting, as it has been proven by [77] that using some state-of-the-art attacks as penalties during defensive training can weaken the defence against some or all of these and new attacks compared to even completely undefended training.

5.6 At Model Level: Training with Differential Privacy

The idea behind *differential privacy* (DP) is to gather confidential user data for analysis without compromising the confidentiality of each individual user. It was formally defined in 2006 by [136] – the algorithm K is considered to be ϵ-private if for all datasets D_1 and D_2 differing in at most one data entry and all events S

$$Pr[K(D_1) \in S] \leq exp(\epsilon) + Pr[K(D_2) \in S].$$

This can be interpreted as follows: a differentially private algorithm's functionality should remain unchanged whether any single entry is or is not present in its training dataset. In other words, unlike for some of the other defences, DP provides a guarantee on the maximum privacy loss: the maximum divergence between these two distributions (or a maximum log odds ratio for any event S) is bounded by the privacy parameter ϵ.[5] This guarantee is also known as "pure" differential privacy.

Concentrated DP and Rényi DP. There exist generalizations and relaxations of DP methods, that tend to enjoy higher accuracy than "pure" DP. For instance, (ϵ, δ)-differential privacy, [137], guarantees that with probability of at most $(1 - \delta)$ the privacy loss does not exceed ϵ. Typically this helps with the trade-off between privacy and accuracy of the model, and "pure" DP can be viewed as a special case when $\delta = 0$. However, in the case of multiple queries, the bound grows, which is why [138] proposed *Concentrated Differential Privacy (CDP)* relaxation, not only improving on the accuracy but also offering tighter bounds on the expected privacy loss for *group privacy*. The privacy loss accounting, training efficiency and model quality can

4. See Sec. 4.2 for explanation of the *shadow model* training.

5. ϵ is also sometimes called a "privacy budget".

be improved using two different data batching techniques proposed by [139] as an extension to classic CDP. Further quantitative results for CDP were provided in [140] by redefining the concept of DP in terms of the Réényi divergence between the distributions obtained by running an algorithm on neighboring input, and defining *zero-Concentrated Differential Privacy (zCDP)* with its corresponding lower bounds. An alternative approach is to adopt *Rényi Differential Privacy (RDP)* proposed by [141], which claims more accurate analysis of the privacy loss due to another relaxation – CDP requires a linear bound on all positive moments of a privacy loss variable, whereas [141] definition applies to one moment at a time. Further, [142] proves a tight upper bound on RDP for subsampling in DP, it also generalizes the results of the *moments accounting technique* [143], to any RDP algorithm. The *moments accounting technique* [143], is a DP framework for deep learning, that improved training computational efficiency by introducing algorithms for efficient gradient computation for individual training examples, sharding tasks into smaller batches to reduce memory footprint, and applying differentially private principal projection at the input layer. Tool-wise, it builds its DP training framework on top of Tensorflow [144].

Differential Privacy Surveys. In addition to some of the aforementioned previous privacy reviews, [4, 5, 5], there exist several surveys focusing specifically on DP, from early works such as [145], to [146, 147]. In this and the following subsections we refer the reader to these to create as complete a picture of the field as possible.

Applications of DP with respect to different tasks. The more traditional applications of DP, outlined in [136] are the DP online learning, [148–150] and DP empirical risk minimization, [151–157]. However, the range of learning tasks that DP was applied to has widened and now includes nearly anything from the federated ML setting [158] to differentially private recurrent language models [159], and even differentially private generative adversarial networks [61, 63], specific DP-GAN applications for generating time-series [62, 111], and tabular mixed feature datasets [160].

Evaluation and Utility-Privacy Trade-Off of DP methods. The utility vs privacy trade-off has been one of the most important topics in Differential Privacy, partly due to the lack of formal utility-loss guarantees [134, 161].

Evaluation of privacy guarantees for DP is more established compared to some of the other defence methods. However, despite the various DP methods, and the provable upper bounds on the(ir) maximum privacy loss, there remains relatively little understanding of the trade-off between the size of the privacy budget ϵ and the utility of the resulting model. It is typical in DP works to select large values for ϵ to show reasonable utility scores [43, 162]. Practically, [43] finds that there is a huge gap between the upper bounds on privacy loss that can be guaranteed, and the effective privacy loss that can be measured using current inference attacks. Moreover, there is no agreed upon threshold for ϵ, at which privacy guarantees are rendered meaningless. Their empirical assessment shows that for an acceptable utility level the privacy guarantees are practically meaningless, although the observed level for leakage under the inference attacks is still low.

Advancing further on DP under inference attacks, [163] offers more empirical assessment of data leakage under inference attacks, considering single and joint decoding (MIA, see Sec. 4.2 for single data instance at a time vs a subset of data instances at a time), finding that the joint decoding is more powerful, and offering a method to empirically decide on the size of the privacy budget ϵ.

Some research has been conducted on eliminating the privacy-utility trade-off and replacing it with privacy-computational cost trade-off instead by [164]. They propose a stochastic gradient descent-based DP (Sec. 5.8) for recurrent language models in a federated learning setting (Sec. 5.8).

Risks. DP has been shown to be insecure under PIAs (see Sec. 4.4), because of the different types of data leakage considered by PIA and DP [51].

Moreover, (ϵ, δ)-differential privacy retains the possibility of failures, i.e. a DP algorithm can in theory reveal the sensitive data it has been trained on. No mechanism has been proposed for detection and reporting of this kind of leakage, which is a serious issue [165].

For neural networks, two more recent approaches of implementing DP are particularly relevant, and the next two subsections are dedicated to these. Note that these are merely sub-classes of DP methods, and share general limitations and vulnerabilities of DP methods.

5.7 At Model Level: Private Aggregation of Teaching Ensembles (PATE)

Private Aggregation of Teaching Ensembles (PATE), [158, 166, 167], and its modification *PATE-G*, [168], is a subset of differential privacy techniques based on the teacher-student approach, using ensemble methods ([169]) aggregation and some of the GAN-based architecture for PATE-G, [18, 70].

At training time the ensemble of teacher networks is trained on the disjoint subsets of the training dataset with strong privacy guarantees, and then the student network is used to aggregate the teacher network's knowledge in a noisy fashion, i.e. the student is black-box-querying the teacher ensemble, receiving the noisy labels. PATE methods train the student only on the labelled training data, whilst PATE-G also uses the unlabelled data (via GANs or Virtual Adversarial training). At inference time, only the student model is used. The teacher models are never publicly shared, and the student model never comes in contact with the training dataset, thus the noisy aggregation of the teacher ensemble at the training time provides the privacy guarantees [165].

The scalability of PATE methods has been practically confirmed by [170] (on SVHN and the UCI Adult datasets). They further proposed to use concentrated noise (swapping Laplacian for Gaussian noise during aggregation) for further improvement of the teacher ensemble results, as well as not returning an answer to the student network at training time in the absence of teacher ensemble consensus. They report both high utility and privacy guarantees for $\epsilon < 1$.

Applications Theoretically, PATE can be universally applied to a variety of models, so, although more classical works are concerned with the classifiers, [171, 172] focus specifically on the data generation with DP guarantees. G-PATE [171],[6] trains a student-generator with an ensemble of teacher discriminators. PATE-GAN [172] trains a student classifier on synthetically generated data, using a noisy aggregation of the teacher-discriminator labels.

5.8 At Model Level: the Gradient Descent Perturbations

Neural network training relies on gradient descent, and adding noise is a popular technique both for better generalization [173, 174], and for ensuring differential privacy, if the noise is appropriately calibrated [161]. Since weight changes with respect to the training data occur through a gradient update, both gradient clipping and adding noise to gradient computations are valid techniques for ensuring DP, explored by variety of methods [157, 175–177].

More recent advances of the noisy SGD include extension with the moments accounting technique [143], a scalable and computationally efficient "bolt-on" output perturbation technique by [178], and DP-LSSGD [179], based on Laplacian smoothing SGD, that stabilizes the training of DP models, leading to better generalization and higher utility of the resulting DP models. Finally, adaptive allocation of the privacy budget at the iteration level [180], and [181] applying the control variates technique [182, 183] to stochastic gradient descent update are both compatible with zCDP (see Sec. 5.6 for more details on zCDP).

5.9 At Model Level: Federated / Collaborative Learning

Federated (or collaborative) Learning (FL) trains an ML model on a central server, across multiple decentralized databases, holding local data samples, without exchanging them directly [184–186], thus, potentially mitigating risks of the direct data leakage.

Surveys. There are a number of surveys covering FL in general, [187–190]. We would like to refer the reader to [191] which focuses mainly on privacy concerns for FL. It shows some evidence that FL is not always able to provide good privacy guarantees, as well as, outlines two major challenges to the classic federated learning setting: poisoning and inference attacks (see Sections 4.2 and 4.5).

FL vulnerability to privacy attacks. FL is sometimes offered as a solution to the problem of balancing user data privacy requirements (such as GDPR [15]) with the benefits of learning from multiple data sources, [187]. However, FL does not provide foolproof privacy guarantees. Successful white-box MIAs have been performed by [44] against both centralised and federated learning, even for cases with well-generalised target models. These attacks leverage stochastic gradient descent (SGD) vulnerabilities; specifically they compute membership probability for each data point based on the gradient vector of all parameters with respect to this data point. Furthermore, [191] not only concludes that classic FL frameworks are often vulnerable to inference and poisoning attacks (Sections 4.2 and 4.5), it also expresses concerns with the current methods of defences against these attacks for FL.

6. G-PATE not to be confused with PATE-G acronym, G-PATE is merely one of the PATE-G methods.

Malicious servers An alternative to a malicious user is a malicious server provider aiming to steal client's data. Recently [192] proposed the first ever attack from the perspective of such a malicious server. It uses a GAN [18, 70] multi-task discriminator, designed to recover the category and the client identity of the input data. It is designed to run "invisibly" on a server leaving the clients unaware.

Differential privacy for FL. Efforts have been made to secure the classic FL framework relying on differential privacy [158, 167, 193–196]. Some concerns remain on privacy-utility trade-offs [164], and property inference attacks for groups of records (rather than a single record) [50].

Other defences for FL. Another important point raised in [191] is the lack of clarity on whether certain defences, such as adversarial defences, could be applied for privacy protection of FL systems. A more traditional alternative defence is homomorphic encryption, used to mask the local gradient updates, either individually, e.g. [197–199], or in batches in order to reduce the computation costs [200].

Applications. Federated learning is widely applied in applications involving the use of sensitive data, e.g., recommendation systems, mobile applications, transaction fraud detection, and healthcare [187, 190, 199, 201]. Nevertheless, according to [190], there are not many FL applications that explicitly focus on privacy preservation. Still, there are some examples of privacy-preserving recommendation systems [202, 203], that rely primarily on data encryption (see the next section) for their privacy guarantees.

5.10 At Model Level: Operating on Encrypted Data

Traditional encryption requires the sharing of the key amongst the parties involved, which interferes with individual privacy. However, *Homomorphic Encryption (HE)* techniques allow any third party to operate on the encrypted data without decrypting it in advance, and, furthermore, *Fully Homomorphic Encryption(FHE)*, [204], allows for any computable function to perform on the encrypted data [205].

Surveys. Homomorphic Encryption is a vast and well-established field, hence, for the sake of brevity, we refer the reader to the relevant surveys [205, 206].

Limitations. Operating on encrypted data could alleviate privacy issues, but unfortunately its low efficiency often makes FHE impractical in the real world [207]. However, there are a number of advances, involving somewhat homomorphic encryption, aimed to improving efficiency – please refer to [208] for a detailed overview.

5.11 At Model Level: Knowledge Distillation

Knowledge distillation has been actively used to compress models and thus facilitate deployment on resource constrained devices, however can also be applied to preserve privacy. E.g., Distillation for Membership Privacy (DMP) uses distillation to train models with membership privacy by leveraging various sources of noise in the model distillation process [209]. Distillation-based methods based on the fast gradient sign method [210] and the Jacobian attack [211] have been shown to train privacy preserving models where large perturbations to the input are required to make a distilled model cause a wrong prediction. However, [212] showed that distillation fails to mitigate attach variants proposed in [213].

5.12 At Model Level: Other Privacy-preserving ML

Various other methodologies exist that work at model level to protect against adversary attacks. PRADA [37] defends against model extraction attacks by flagging multiple queries made against a model when they deviate against general inference behaviour. [214] designed a privacy preserving framework to protect ML algorithms such as linear and logistic regression, as well as neural networks at training time itself. [215] and [216] presented a privacy preserving alternative to SGD when multiple data owners wish to train a model combining their data without sharing the data with each other, sharing weight parameters instead of gradient updates. FPPDL [217] is a decentralized privacy preserving framework based on Blockchain for decentralization, and differential privacy (DPGAN) along with a 3 layer onion encryption to facilitate fairness. VIPS [218] overcomes the high amount of additional noise needed to make variational Bayes privacy preserving by combining a moment accountant to get a tight bound on the privacy cost of multiple VB iterations. [25] show how even well-generalized models can leak data and that overfitting is important, but not a necessary condition for information leakage. Prediction purification was shown to protect models from inversion and membership inference attacks [219]. This is done using a purifier network, which is adversarial in nature and maps the confidence scores of a classifier to a reconstructed privacy-preserving representation.

6 Metrics

Assessment of the data leakage in trained machine learning models remains an open area of research. Measuring leakage is case-specific, as it depends on the data type and the type of malevolent/involuntary leakage in question. Further, any knowledge about the exact type and architecture of the attack used by the adversary might be crucial for the ability to protect against it.

Assessing involuntary leakage may be easy for some components e.g. overfitting via generalization error. Others, such as memorization and feature leakage, are harder to troubleshoot. An *exposure* metric suggested by [16, 68] estimates a model's potential for memorizing rare and unique sequences in text data (thus far, there are no extensions to other data types). Additionally, an assessment proposed by [69] focuses on estimating memorization in the lower layers of convolutional neural networks.

Assessing data leakage via attacks can be occasionally be straightforward, e.g., for MIAs the membership inference easily translates into the re-identification score [82, 83].

Assessing data leakage for defence purposes Examples of this include Kolmogorov-Smirnov distance used for verifying forgetting in [127], metrics proposed by [133] for assessing machine unlearning leakage under MIAs, as well as some work on estimating the Bayes risk of the system via universally consistent nearest neighbor (ML) rules [72], improving upon more naïve min-entropy approaches.

Learning metrics as a fairness constraint Most literature in fair ML deals with learning fair classifiers due to the prominence of classification as a learning task. Most proposed methods treat solving for fairness based on the definition of fairness tailored to their specific objective. Of considerable importance are techniques such as those proposed by [220] which not only satisfy fairness constraints, but also tend to be stable towards adversarial attacks and variations in datasets during testing. Regression-based fairness techniques eliminate bias at training time by hand-crafting loss functions that conform to group fairness, individual fairness or hybrid fairness, although they have not received a lot of attention in research [221].

Metrics in Differential Privacy In this setting, due to the provable privacy upper bounds, empirical assessment of both utility and quality guarantees is possible [43, 110]. The Rényi Divergence can be used as a metric to bound any arbitrary privacy loss [43]. The resulting Rényi differential privacy works by creating a bound on each individual moment of the privacy loss, leading to other variants of differential privacy, and to a more accurate numerical analysis of the privacy loss. A synthetic data generating deep learning model with privacy guarantees (DP-SYN) was proposed in [110]. Evaluation of DP-SYN was done using carefully crafted metrics based on ML (misclassification rate), statistics (Total Variation Distance [222] between the noisy and original marginals of the data distributions) and agreement rate (the percentage of records to which two classifiers assign the same prediction [223]).

Limitations. First of all, there are a number of attacks/leakages that can be hard to trace. For instance, there is currently no single reliable way to verify how much of the training data is memorized by a GAN, or how much a property inference attack could infer even from sanitized data, since it would change, depending on the design of the attack and the type of the data in question.

Secondly, there is no universal robust framework for detecting and reporting model plainly revealing the sensitive data (more likely for predictive or generative models), [165], and although it does not necessarily seem like a big issue at first glance, it does impede open access trained model sharing in a commercial setting, as companies will require guarantees on the privacy of their data.

7 Applications

7.1 Data as a Service (DaaS) / Safe Havens

Data as a Service offers an appealing solution to limited data availability in both data-driven research and data-intensive commercial applications, given sufficient privacy guarantees. However, current proposed implementations, e.g. [224], provide no leakage assessment.

Significant efforts are afoot to create a national research infrastructure across the United Kingdom[7] to support data-driven knowledge discovery, including data analysis, statistics, and ML. Organizations such as Health Data Research UK and Research Data Scotland are designing services to facilitate identification of health research datasets, their description, permissions, and accessibility.

7. We are certain similar activities exist across the world.

Federated access models are favoured by data holders, with data scientists invited to access data within Trusted Research Environments, of which Safe Havens in Scotland are but one example. However, with the intellectual and economic benefits of more access to data comes an escalating risk of data leakage. The proliferation of data access tools, environments, protocols, and transfers, coupled with escalating volumes of data, is driving persistent privacy concerns.

The current official protocols, in the UK, rely on statistical disclosure control, [19, 20], data pseudoanonymization, [225], and true data anonymization, [226], since fewer legal restrictions apply to anonymized data. However, from a legal perspective, anonymized data lies in a gray area according to [227]. In fact some regulations, such as Data Protection Directive (1995) [228], Data Protection Act 1998 (DPA) [229], and GDPR [15], for example, do not require strictly risk-free data protection, however the risk of the re-identification should be mitigated to the extent when it is remote. GDPR regulations do not apply to truly anonymized data either – Recital 26 defines the anonymous information, as " *information which does not relate to an identified or identifiable natural person or to personal data rendered anonymous in such a manner that the data subject is not or no longer identifiable*", [230]. It does, however, apply to pseudoanonymized and non-anonymized, both of which are often more useful for practical purposes, and are preferable for both statistical analysis and machine learning techniques in Data as a Service (DaaS) setting, such as Safe Havens.

If Data as a Service (DaaS) using linked and unconsented public data, meeting the requisite GDPR safeguards and standards for privacy is to continue, techniques to mitigate data linkage are imperative. Therefore custodians considering DaaS, especially with sensitive data categories, have a difficult dual duty to both respect public privacy, and to foster public benefit through research.

From an ML perspective, this results in a growing demand for the implementation of reliable checks on models exported from DaaS facilities. This requires furthering our understanding and control over involuntary data leakage, and progressively more reliable methods of defence from malevolent attackers, such as MIAs and PIAs (Section 3, and Subsections 4.2 and 4.4 respectively).

7.2 ML Models as a Service (MLaaS)

Machine Learning as a Service (MLaaS) represents an extended privacy risk, further to that posed by Data as a Service. The development of ML models can risk perpetuating bias, state intrusion, inequalities, and the potential erosion of privacy.

Whilst the separation of source data from MLaaS could ameliorate data leakage concerns, the outputs, decisions, and unintended applications of MLaaS add complexity to the tracing of potential leakage. Quarantining MLaaS to Cloud deployments may insulate personal data from interrogants; however, benign or deliberate data leakage remain a potential threat.

Certain settings of MLaaS, including federated learning, can be vulnerable to inference type attacks, e.g., MIAs [41, 104], with defence mechanisms shown to mitigate those risks explored for classification models [104, 139]. Moreover, [95] showed that Amazon Rekognition, a commercial MLaaS API, can be vulnerable to model inversion attacks.

Research in MLaaS data safety remains important to understand the risks posed by models as they are deployed, trained, and evolve on exposure to new data. Presently a number of providers such as Amazon ML [231], Google Cloud [232], and IBM [233] are providing MLaaS for public and commercial use.

Sections 3 and 4 of this survey cover the implications of sharing ML models trained on sensitive data in open access, whereas Section 5 touched upon current defence methods and their shortcomings.

7.3 ML models in Mobile Applications

ML methods are commonly used to support mobile applications. Thus, privacy attacks, e.g., MIAs [17], attribute inference attacks, and PIAs (see Sec. 4.2 and 4.4) are a possibility. Sensitive information might involve anything from the full user profile (under MIA) to the user's location [79, 80], or gender and sexual orientation [234].

Federated learning (see Sec. 5.9 for more details on risks and defences) appeals in this context as means of privacy protection. Although some research for protecting mobile users specifically exist, e.g., [234] and [201], this field is still somewhat in its adolescence.

8 Challenges and Opportunities

Our findings thus far can be summarized as follows:

Attacks are not evenly explored across different data types or tasks. For instance, MIAs (Sec. 4.2) are not well investigated for tasks such as regression or segmentation, MEAs (Sec. 4.3) have not been verified for generative models, and PIAs have only been applied to classification tasks.

This points to the need to uniformly probe weaknesses of leakage across several tasks and data types via advancements in attacks.

Defences at the data level lie in between data being potentially anonymized (or sanitised, obfuscated) to the point where they are no longer useful, and data being likely re-identifiable through inference attacks. Replacing real personal data with synthetic data could be a promising direction, albeit they remain vulnerable to property and attribute inference attacks [55].

Data privacy can be largely contextual, i.e., in certain situations a publicly accessible dataset can potentially enable recovering individuals' identities, when combined with other supposedly public datasets.

Defences via model have not been evenly explored across different tasks, data, and attacks/leakage, and may often work only for specific settings. For example, adversarial defences are mostly explored for MIA-type attacks, DP-based defences may not universally succeed against MIAs, and the privacy guarantees of classic FL may not be as strong as we desire.

Homomorphic Encryption remains a promising direction for privacy-preserving FL; however, its practical implementation is not straightforward and requires compute power in a homogenous setup across all parties involved.

We remain in need of computationally efficient defences, which can offer a wide range of privacy guarantees.

Detection and Assessment of Leakage and Tools Furthermore, uniform mechanisms for reporting data leakage are lacking. For instance, in a DaaS scenario a malicious user could potentially encode sensitive data within the NN model weights – yet a check / mechanism to reliably detect even such a simple form of leakage is lacking.

We find that established tools, and practical and universally applicable software packages developed from already existing research are lacking. This results in an opportunity to develop mechanisms for transparent reporting and equally need to develop robust software tools that can help bridge the gap between proof-of-concept and practical utility.

9 Conclusion

While data leakage research is not new, the field is ever-evolving due to the dynamic (and rapid) nature of machine learning development. New privacy risks and attacks arise, which are then met by new efforts to protect against them, resulting in a constant adversarial game. This survey unifies the currently available research and summarizes our understanding of inference-time information leakage in ML, both involuntary and malevolent, as well as the means which are currently available to measure and prevent such leakage. It results in a rich comprehensive taxonomy of the broad field of privacy in ML.

We find that, first of all, understanding of data leakage, its causes and implications, is unexplored and our hope is that this survey will positively contribute towards furthering our appreciation of leakage. Our survey reveals opportunities to improve the means to measure, detect and report sensitive data leakage. Secondly, the privacy attacks exploration has been uneven in its coverage of ML tasks and architectures, data types, and attack structures.

And finally, we find that most available defences are case-specific, and scaling to larger datasets with guarantees on performance remains a challenge. Overall these findings indicate that leakage, privacy, and the necessary defenses remains an area which is fertile for further research and development.

Acknowledgment

This work is supported by iCAIRD, which is funded by Innovate UK on behalf of UK Research and Innovation (UKRI) [project number 104690]. S.A. Tsaftaris acknowledges also support by a Canon Medical / Royal Academy of Engineering Research Chair under Grant RCSRF1819\8\25.

References

[1] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognit.*, vol. 84, pp. 317–331, 2018.

[2] D. Maiorca, B. Biggio, and G. Giacinto, "Towards adversarial malware detection: Lessons learned from pdf-based attacks," *ACM Comput. Surv.*, vol. 52, 2019.

[3] X. Wang, J. Li, X. Kuang, T. yu an, and J. Li, "The security of machine learning in an adversarial setting: A survey," *Journal of Parallel and Distributed Computing*, vol. 130, 2019.

[4] N. Papernot, P. McDaniel, A. Sinha, and M. P. Wellman, "SoK: Security and privacy in machine learning," in *IEEE EuroSP*, 2018, pp. 399–414.

[5] E. De Cristofaro, "An overview of privacy in machine learning," *arXiv*, 05 2020.

[6] M. Rigaki and S. Garcia, "A survey of privacy attacks in machine learning," *arXiv*, 07 2020.

[7] S. Chang and C. Li, "Privacy in neural network learning: Threats and countermeasures," *IEEE Network*, vol. 32, pp. 61–67, 2018.

[8] H. C. Tanuwidjaja, R. Choi, and K. Kim, "A survey on deep learning techniques for privacy-preserving," in *ML4CS*, 2019.

[9] J. Zhang, C. Li, J. Ye, and G. Qu, "Privacy threats and protection in machine learning," *2020 Great Lakes Symposium on VLSI*, 2020.

[10] S. Rezaei and X. Liu, "Security of deep learning methodologies: Challenges and opportunities," *ArXiv*, 2019.

[11] M. Ozdag, "Adversarial attacks and defenses against deep neural networks: A survey," *Procedia Computer Science*, vol. 140, pp. 152–161, 2018.

[12] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," *CoRR*, 2018.

[13] H. Xu, Y. Ma, H. Liu, D. Deb, H. Liu, J. Tang, and A. K. Jain, "Adversarial attacks and defenses in images, graphs and text: A review," *CoRR*, 2019.

[14] K. Ren, T. Zheng, Z. Qin, and X. Liu, "Adversarial attacks and defenses in deep learning," *Engineering*, vol. 6, no. 3, pp. 346–360, 2020.

[15] P. Voigt and A. v. d. Bussche, *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer, 2017.

[16] N. Carlini, J. Kos, Ú. Erlingsson, and X. Song, "The Secret Sharer : Evaluating and Testing Unintended Memorization in Neural Networks," *USENIX*, pp. 267–284, 2019.

[17] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," in *IEEE SSP*, 2017, pp. 3–18.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, vol. 27. Curran Associates, Inc., 2014.

[19] *Statistical Disclosure Control Protocol*, NHS National Services Scotland, 2009.

[20] *Guidance on intruder testing*, www.ons.gov.uk, Office for National Statistics.

[21] F. Yang, A. Kale, Y. Bubnov, L. Stein, Q. Wang, H. Kiapour, and R. Piramuthu, "Visual search at ebay," ser. KDD '17. ACM, 2017, p. 2101–2110.

[22] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *IEEE/CVF CVPR*. IEEE, 2015, pp. 815–823.

[23] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image*

Analysis, vol. 42, pp. 60–88, 2017.

[24] V. Chandrasekaran, K. Chaudhuri, I. Giacomelli, S. Jha, and S. Yan, “Exploring connections between active learning and model extraction,” in *USENIX*, 2020, pp. 1309–1326.

[25] Y. Long, V. Bindschaedler, L. Wang, D. Bu, X. Wang, H. Tang, C. A. Gunter, and K. Chen, “Understanding membership inferences on well-generalized learning models,” *CoRR*, 2018.

[26] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, “Stealing machine learning models via prediction APIs,” in *USENIX*, 2016, pp. 601–618.

[27] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, “The secret revealer: Generative model-inversion attacks against deep neural networks,” *IEEE/CVF CVPR*, pp. 250–258, 2020.

[28] A. Salem, Y. Zhang, M. Humbert, M. Fritz, and M. Backes, “ML-Leaks: Model and data independent membership inference attacks and defenses on machine learning models,” *ArXiv*, 2019.

[29] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, “Privacy risk in machine learning: Analyzing the connection to overfitting,” in *IEEE Computer Security Foundations Symposium*. IEEE, 2018, pp. 268–282.

[30] L. Zhu, Z. Liu, and S. Han, “Deep leakage from gradients,” in *NeurIPS*, vol. 32. Curran Associates, Inc., 2019, pp. 14 774–14 784.

[31] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, and N. Papernot, “High accuracy and high fidelity extraction of neural networks,” in *USENIX*, 2020.

[32] S. J. Oh, M. Augustin, M. Fritz, and B. Schiele, “Towards reverse-engineering black-box neural networks,” in *ICLR*, 2018.

[33] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, and N. Borisov, “Property inference attacks on fully connected neural networks using permutation invariant representations,” *SIGSAC*, 2018.

[34] Z. He, T. Zhang, and R. Lee, “Model inversion attacks against collaborative inference,” *Computer Security Applications Conference*, 2019.

[35] B. Hilprecht, M. Härterich, and D. Bernau, “Monte Carlo and reconstruction membership inference attacks against generative models,” *Proc. Priv. Enhancing Technol.*, pp. 232–249, 2019.

[36] B. Hitaj, G. Ateniese, and F. Pérez-Cruz, “Deep models under the GAN: information leakage from collaborative deep learning,” in *SIGSAC*. ACM, 2017, pp. 603–618.

[37] M. Juuti, S. Szyller, S. Marchal, and N. Asokan, “PRADA: Protecting against DNN model stealing attacks,” in *IEEE EuroSSP*, 2019, pp. 512–527.

[38] S. Milli, L. Schmidt, A. D. Dragan, and M. Hardt, “Model reconstruction from model explanations,” in *Conference on Fairness, Accountability, and Transparency*. ACM, 2019.

[39] N. Papernot, P. McDaniel, I. J. Goodfellow, S. Jha, Z. Y. Celik, and A. Swami, “Practical black-box attacks against machine learning,” *ACM on Asia Conference on Computer and Communications Security*, 2017.

[40] M. A. Rahman, T. Rahman, R. Laganière, and N. Mohammed, “Membership inference attack against differentially private deep learning model,” *Trans. Data Priv.*, vol. 11, pp. 61–79, 2018.

[41] S. Truex, L. Liu, M. Gursoy, L. Yu, and W. Wei, “Demystifying membership inference attacks in machine learning as a service,” *IEEE Transactions on Services Computing*, 2019.

[42] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, “Beyond inferring class representatives: User-level privacy leakage from federated learning,” *IEEE INFOCOM*, pp. 2512–2520, 2019.

[43] B. Jayaraman and D. Evans, “Evaluating differentially private machine learning in practice,” in *USENIX*, 2019.

[44] M. Nasr, R. Shokri, and A. Houmansadr, “Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning,” *IEEE SSP*, pp. 739–753, 2019.

[45] A. Sablayrolles, M. Douze, C. Schmid, Y. Ollivier, and H. Jegou, “White-box vs black-box: Bayes optimal strategies for membership inference,” ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 2019, pp. 5558–5567.

[46] G. Kaissis, A. Ziller, J. Passerat-Palmbach, T. Ryffel, D. Usynin, A. Trask, I. Lima, J. Mancuso, F. Jungmann, M.-M. Steinborn, A. Saleh, M. Makowski, D. Rueckert, and R. Braren, “End-to-end privacy preserving deep learning on multi-institutional medical imaging,” *Nature Machine Intelligence*, 2021.

[47] M. Fredrikson, S. Jha, and T. Ristenpart, “Model inversion attacks that exploit confidence information and basic countermeasures,” in *SIGSAC*. ACM, 2015, pp. 1322–1333.

[48] S. Hidano, T. Murakami, S. Katsumata, S. Kiyomoto, and G. Hanaoka, “Model inversion attacks for prediction systems: Without knowledge of non-sensitive attributes,” in *Conference on Privacy, Security and Trust (PST)*, 2017.

[49] B. Wang and N. Z. Gong, “Stealing hyperparameters in machine learning,” in *IEEE SSP*, 2018, pp. 36–52.

[50] L. Melis, C. Song, E. D. Cristofaro, and V. Shmatikov, “Exploiting unintended feature leakage in collaborative learning,” in *IEEE SSP*, 2019, pp. 691–706.

[51] G. Ateniese, G. Felici, L. Mancini, A. Spognardi, A. Villani, and D. Vitali, “Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers,” *International Journal of Security and Networks*, vol. 10, 06 2013.

[52] D. Dua and C. Graff, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[53] M. Fredrikson, E. Lantz, S. Jha, S. M. Lin, D. Page, and T. Ristenpart, “Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing,” in *USENIX*, 2014, pp. 17–32.

[54] A. Tucker, Z. Wang, Y. Rotalinti, and P. Myles, “Generating high-fidelity synthetic patient data for assessing machine learning healthcare software,” *NPJ Digital Medicine*, vol. 3, 2020.

[55] T. Stadler, B. Oprisanu, and C. Troncoso, “Synthetic data - A privacy mirage,” *CoRR*, 2020.

[56] V. Nagarajan, C. Raffel, and I. J. Goodfellow, “Theo-

retical insights into memorization in GANs," *NeurIPS Workshop*.

[57] Y. Kim, M. Kim, and G. Kim, "Memorization precedes generation: Learning unsupervised GANs with memory networks," *CoRR*, 2018.

[58] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "LOGAN: evaluating privacy leakage of generative models using generative adversarial networks," 2017.

[59] J. Hayes, L. Melis, G. Danezis, and E. D. Cristofaro, "LOGAN: membership inference attacks against generative models," *Proc. Priv. Enhancing Technol.*, pp. 133–152, 2019.

[60] D. Chen, N. Yu, Y. Zhang, and M. Fritz, "GAN-Leaks: A taxonomy of membership inference attacks against GANs," *CoRR*, 2019.

[61] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou, "Differentially private generative adversarial network," *CoRR*, 2018.

[62] L. Frigerio, A. S. de Oliveira, L. Gomez, and P. Duverger, "Differentially private generative adversarial networks for time series, continuous, and discrete open data," in *ICT Systems Security and Privacy Protection*, ser. IFIP Advances in Information and Communication Technology, vol. 562. Springer, 2019, pp. 151–164.

[63] X. Zhang, J. Ding, S. M. Errapotu, X. Huang, P. Li, and M. Pan, "Differentially private functional mechanism for generative adversarial networks," in *GLOBECOM*. IEEE, 2019.

[64] Y. He, S. Rahimian, B. Schiele, and M. Fritz, "Segmentations-leak: Membership inference attacks and defenses in semantic image segmentation," *CoRR*, 2019.

[65] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *ICLR*, 2017.

[66] M. Veale, R. Binns, and L. Edwards, "Algorithms that remember: model inversion attacks and data protection law," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 376, no. 2133, 2018.

[67] C. Song, T. Ristenpart, and V. Shmatikov, "Machine learning models that remember too much," in *SIGSAC*. ACM, 2017, pp. 587–601.

[68] N. Carlini, C. Liu, J. Kos, Ú. Erlingsson, and D. Song, "The secret sharer: Measuring unintended neural network memorization & extracting secrets," *CoRR*, 2018.

[69] A. Sablayrolles, M. Douze, C. Schmid, and H. Jégou, "Déjà Vu: an empirical evaluation of the memorization properties of ConvNets," *CoRR*, 2018.

[70] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved Techniques for Training GANs," in *NeurIPS*, vol. 29. Curran Associates, Inc., 2016.

[71] S. Yeom, M. Fredrikson, and S. Jha, "The unintended consequences of overfitting: Training data inference attacks," *CoRR*, 2017.

[72] G. Cherubin, K. Chatzikokolakis, and C. Palamidessi, "F-BLEAU: fast black-box leakage estimation," in *IEEE SSP*, 2019, pp. 835–852.

[73] C. Song and V. Shmatikov, "Overlearning reveals sensitive attributes," *ArXiv*, 2020.

[74] K. Leino and M. Fredrikson, "Stolen memories: Leveraging model memorization for calibrated white-box membership inference," *ArXiv*, 2020.

[75] S. Yeom, I. Giacomelli, A. Menaged, M. Fredrikson, and S. Jha, "Overfitting, robustness, and malicious algorithms: A study of potential causes of privacy risk in machine learning," *J. Comput. Secur.*, vol. 28, pp. 35–70, 2020.

[76] M. Nasr, R. Shokri, and A. Houmansadr, "Machine learning with membership privacy using adversarial regularization," in *SIGSAC*. ACM, 2018, pp. 634–646.

[77] L. Song, R. Shokri, and P. Mittal, "Privacy risks of securing machine learning models against adversarial examples," in *SIGSAC*, ser. CCS '19. ACM, 2019, p. 241–257.

[78] G. Liu, C. Wang, K. Peng, H. Huang, Y. Li, and W. Cheng, "SocInf: membership inference attacks on social media health data with machine learning," *IEEE Transactions on Computational Social Systems*, vol. 6, pp. 907–921, 2019.

[79] A. Pyrgelis, C. Troncoso, and E. D. Cristofaro, "Knock knock, who's there? Membership inference on aggregate location data," *CoRR*, 2017.

[80] A. Pyrgelis, C. Troncoso, and E. De Cristofaro, "Under the hood of membership inference attacks on aggregate location time-series," *CoRR*, 2019.

[81] S. Hisamoto, M. Post, and K. Duh, "Membership inference attacks on sequence-to-sequence models: Is my data in your machine translation system?" *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 49–63, 2020.

[82] Y. Long, V. Bindschaedler, and C. A. Gunter, "Towards measuring membership privacy," *ArXiv*, 2017.

[83] L. Rocher, J. Hendrickx, and M. Y.-A. de, "Estimating the success of re-identifications in incomplete datasets using generative models," *Nature Communications*, vol. 10, 2019.

[84] J. R. Correia-Silva, R. F. Berriel, C. Badue, A. F. de Souza, and T. Oliveira-Santos, "Copycat CNN: stealing knowledge by persuading confession with random non-labeled data," in *International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8.

[85] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff nets: Stealing functionality of black-box models," *IEEE/CVF CVPR*, pp. 4949–4958, 2019.

[86] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[87] K. Krishna, G. S. Tomar, A. P. Parikh, N. Papernot, and M. Iyyer, "Thieves on Sesame Street! model extraction of BERT-based APIs," *ArXiv*, 2020.

[88] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Póczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," in *NIPS*, 2017.

[89] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 2938–2948.

[90] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial

lens," in *ICML*. PMLR, 2019, pp. 634–643.

[91] F. Suya, S. Mahloujifar, A. Suri, D. Evans, and Y. Tian, "Model-targeted poisoning attacks with provable convergence," *ArXiv*, 2020.

[92] M. Chase, E. Ghosh, and S. Mahloujifar, "Property inference from poisoning," *ArXiv*, 2021.

[93] Z. He, T. Zhang, and R. B. Lee, "Model inversion attacks against collaborative inference," in *Computer Security Applications Conference*, ser. ACSAC '19. ACM, 2019, p. 148–162.

[94] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE INFOCOM*, 2019, pp. 2512–2520.

[95] Z. Yang, J. Zhang, E.-C. Chang, and Z. Liang, "Neural network inversion in adversarial setting via background knowledge alignment," in *SIGSAC*, ser. CCS '19. ACM, 2019, p. 225–240.

[96] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, "The secret revealer: Generative model-inversion attacks against deep neural networks," in *IEEE/CVF CVPR*, 2020, pp. 250–258.

[97] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *NeurIPS*, vol. 32. Curran Associates, Inc., 2019.

[98] A. Salem, A. Bhattacharyya, M. Backes, M. Fritz, and Y. Zhang, "Updates-leak: Data set inference and reconstruction attacks in online learning," *ArXiv*, 2020.

[99] X. Wu, M. Fredrikson, S. Jha, and J. F. Naughton, "A methodology for formalizing model-inversion attacks," in *IEEE CSF*, 2016, pp. 355–370.

[100] T. Titcombe, A. J. Hall, P. Papadopoulos, and D. Romanini, "Practical defences against model inversion attacks for split neural networks," *ArXiv*, 2021.

[101] P. Vepakomma, A. Singh, O. Gupta, and R. Raskar, "NoPeek: information leakage reduction to share activations in distributed deep learning," in *ICDM Workshops*. IEEE, 2020, pp. 933–942.

[102] Z. Yang, B. Shao, B. Xuan, E. Chang, and F. Zhang, "Defending model inversion and membership inference attacks via prediction purification," *CoRR*, 2020.

[103] T. Zhang, Z. He, and R. Lee, "Privacy-preserving machine learning through data obfuscation," *ArXiv*, 2018.

[104] C. Wang, G. Liu, H. Huang, W. Feng, K. Peng, and L. Wang, "MIASec: Enabling data indistinguishability against membership inference attacks in MLaaS," *IEEE Transactions on Sustainable Computing*, vol. 5, pp. 365–376, 2020.

[105] A. K. Zaman, C. Obimbo, and R. A. Dara, "An improved data sanitization algorithm for privacy preserving medical data publishing," in *Canadian Conference on AI*, 2017.

[106] P. Chan, Z.-M. He, H. Li, and C.-C. Hsu, "Data sanitization against adversarial label contamination based on data complexity," *International Journal of Machine Learning and Cybernetics*, vol. 9, pp. 1039–1052, 2018.

[107] P. Tambe and D. Vora, "Data sanitization for privacy preservation on social network," *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, pp. 972–976, 2016.

[108] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, pp. 577–590, 2018.

[109] Y. Zhu, S. Yang, C. Chu, and R. Feng, "FlashGhost: Data sanitization with privacy protection based on frequent colliding hash table," *2019 IEEE International Conference on Services Computing (SCC)*, pp. 90–99, 2019.

[110] N. C. Abay, Y. Zhou, M. Kantarcioglu, B. Thuraisingham, and L. Sweeney, "Privacy preserving synthetic data release using deep learning," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2019, pp. 510–526.

[111] J. Yoon, D. Jarrett, and M. van der Schaar, "Time-series generative adversarial networks," in *NeurIPS*, vol. 32. Curran Associates, Inc., 2019, pp. 5508–5518.

[112] S. Mukherjee, Y. Xu, A. Trivedi, N. Patowary, and J. L. Ferres, "privGAN: Protecting GANs from membership inference attacks at low cost to utility," *Proc. Priv. Enhancing Technol.*, vol. 2021, no. 3, 2021.

[113] Y. Cao and J. Yang, "Towards making systems forget with machine unlearning," in *IEEE SSP*. IEEE, 2015, pp. 463–480.

[114] B. Mirzasoleiman, A. Karbasi, and A. Krause, "Deletion-robust submodular maximization: Data summarization with "the right to be forgotten"," in *ICML*, vol. 70. PMLR, 2017, pp. 2449–2458.

[115] A. Ginart, M. Guan, G. Valiant, and J. Y. Zou, "Making ai forget you: Data deletion in machine learning," in *NeurIPS*, vol. 32, 2019, pp. 3518–3531.

[116] Z. Izzo, M. A. Smart, K. Chaudhuri, and J. Y. Zou, "Approximate data deletion from machine learning models: Algorithms and evaluations," *CoRR*, 2020.

[117] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot, "Machine unlearning," *CoRR*, 2019.

[118] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," *CoRR*, 2017.

[119] R. Giordano, W. T. Stephenson, R. Liu, M. I. Jordan, and T. Broderick, "A Swiss Army Infinitesimal Jackknife," in *AISTATS*, vol. 89. PMLR, 2019, pp. 1139–47.

[120] T. Baumhauer, P. Schöttle, and M. Zeppelzauer, "Machine unlearning: Linear filtration for logit-based classifiers," *CoRR*, 2020.

[121] C. Guo, T. Goldstein, A. Y. Hannun, and L. van der Maaten, "Certified data removal from machine learning models," *CoRR*, 2019.

[122] A. Golatkar, A. Achille, and S. Soatto, "Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks," *ArXiv*, 2019.

[123] A. Golatkar, A. Achille, and S. Soatto, "Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations," *ECCV*, vol. 12374, pp. 383–398, 2020.

[124] H. Harutyunyan, A. Achille, G. Paolini, O. Majumder, A. Ravichandran, R. Bhotika, and S. Soatto, "Estimating informativeness of samples with smooth unique information," in *ICLR*, 2021.

[125] S. Garg, S. Goldwasser, and P. N. Vasudevan, "For-

malizing data deletion in the context of the right to be forgotten," in *EUROCRYPT*, vol. 12106. Springer, 2020, pp. 373–402.
[126] D. M. Sommer, L. Song, S. Wagh, and P. Mittal, "Towards probabilistic verification of machine unlearning," *CoRR*, 2020.
[127] X. Liu and S. A. Tsaftaris, "Have you forgotten? A method to assess if machine learning models have forgotten data," *International Conference on Medical Image Computing and Computer Assisted Intervention*, 2020.
[128] A. Golatkar, A. Achille, A. Ravichandran, M. Polito, and S. Soatto, "Mixed-privacy forgetting in deep networks," *ArXiv*, vol. abs/2012.13431, 2020.
[129] J. Liu, M. Juuti, Y. Lu, and N. Asokan, "Oblivious neural network predictions via MiniONN transformations," *SIGSAC*, 2017.
[130] M. Ellers, M. Cochez, T. Schumacher, M. Strohmaier, and F. Lemmerich, "Privacy attacks on network embeddings," *CoRR*, 2019.
[131] C. Song and V. Shmatikov, "Auditing data provenance in text-generation models," ser. KDD '19. ACM, 2019, p. 196–206.
[132] S. Tan, R. Caruana, G. Hooker, and Y. Lou, "Distill-and-compare: Auditing black-box models using transparent model distillation," in *AAAI/ACM Conference on AI, Ethics, and Society*, ser. AIES'18. ACM, 2018, p. 303–310.
[133] M. Chen, Z. Zhang, T. Wang, M. Backes, M. Humbert, and Y. Zhang, "When machine unlearning jeopardizes privacy," *CoRR*, 2020.
[134] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "MemGuard: defending against black-box membership inference attacks via adversarial examples," in *SIGSAC*, 2019, pp. 259–274.
[135] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *ArXiv*, 2018.
[136] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014.
[137] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *EUROCRYPT*. Springer, 2006, pp. 486–503.
[138] C. Dwork and G. N. Rothblum, "Concentrated differential privacy," *CoRR*, 2016.
[139] L. Yu, L. Liu, C. Pu, M. E. Gursoy, and S. Truex, "Differentially private model publishing for deep learning," in *IEEE SSP*. IEEE, pp. 332–349.
[140] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," *ArXiv*, 2016.
[141] I. Mironov, "Rényi differential privacy," *IEEE CSF*, pp. 263–275, 2017.
[142] Y. Wang, B. Balle, and S. P. Kasiviswanathan, "Subsampled renyi differential privacy and analytical moments accountant," in *AISTATS*, ser. Proceedings of Machine Learning Research, vol. 89. PMLR, 2019, pp. 1226–1235.
[143] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," *SIGSAC*, 2016.
[144] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, and A. D. et al., "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2015. [Online]. Available: http://download.tensorflow.org/paper/whitepaper2015.pdf
[145] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*. Springer, 2008.
[146] Z. Ji, Z. C. Lipton, and C. Elkan, "Differential privacy and machine learning: a survey and review," *ArXiv*, 2014.
[147] G. Kaissis, M. Makowski, D. Rückert, and R. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Machine Intelligence*, vol. 2, pp. 305–311, 2020.
[148] P. Jain, P. Kothari, and A. Thakurta, "Differentially private online learning," in *COLT*, 2012.
[149] A. Thakurta and A. D. Smith, "(nearly) optimal algorithms for private online learning in full-information and bandit settings," in *NeurIPS*, 2013.
[150] N. Agarwal and K. Singh, "The price of differential privacy for online learning," in *ICML*, vol. 70. PMLR, 2017, pp. 32–40.
[151] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *J. Mach. Learn. Res.*, vol. 12, pp. 1069–1109, 2011.
[152] D. Wang, M. Ye, and J. Xu, "Differentially private empirical risk minimization revisited: Faster and more general," in *NeurIPS*, 2017, pp. 2722–2731.
[153] Y. Wang, D. Kifer, and J. Lee, "Differentially private confidence intervals for empirical risk minimization," *J. Priv. Confidentiality*, vol. 9, no. 1, 2019.
[154] D. Wang, C. Chen, and J. Xu, "Differentially private empirical risk minimization with non-convex loss functions," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 2019, pp. 6526–6535.
[155] D. Wang and J. Xu, "Differentially private empirical risk minimization with smooth non-convex loss functions: A non-stationary view," in *AAAI Conference on Artificial Intelligence*. AAAI Press, 2019, pp. 1182–89.
[156] K. S. S. Kumar and M. P. Deisenroth, "Differentially private empirical risk minimization with sparsity-inducing norms," *CoRR*, 2019.
[157] R. Bassily, A. D. Smith, and A. Thakurta, "Private empirical risk minimization: Efficient algorithms and tight error bounds," in *IEEE Symposium on Foundations of Computer Science, FOCS*. IEEE, 2014, pp. 464–473.
[158] J. Hamm, Y. Cao, and M. Belkin, "Learning privately from multiparty data," in *ICML*, ser. JMLR Workshop and Conference Proceedings, vol. 48. JMLR, 2016, pp. 555–563.
[159] H. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *ICLR*, 2018.
[160] A. Triastcyn and B. Faltings, "Generating differentially private datasets using GANs," *CoRR*, 2018.
[161] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith, "Calibrating noise to sensitivity in private data anal-

ysis," *J. Priv. Confidentiality*, vol. 7, no. 3, pp. 17–51, 2016.

[162] B. Jayaraman and D. Evans, "When relaxations go bad: "differentially-private" machine learning," *ArXiv*, 2019.

[163] R. Balu, T. Furon, and S. Gambs, "Challenging differential privacy:the case of non-interactive mechanisms," in *ESORICS*. Springer, 2014, pp. 146–164.

[164] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private language models without losing accuracy," *CoRR*, 2017.

[165] M. Abadi, Ú. Erlingsson, I. J. Goodfellow, H. B. McMahan, I. Mironov, N. Papernot, K. Talwar, and L. Zhang, "On the protection of private information in machine learning systems: Two recent approaches," in *IEEE Computer Security Foundations Symposium*. IEEE, 2017.

[166] K. Nissim, "Smooth sensitivity and sampling in private data analysis," in *STOC*. ACM, 2007, pp. 75–84.

[167] M. A. Pathak, S. Rane, and B. Raj, "Multiparty differential privacy via aggregation of locally trained classifiers," in *NeurIPS*. Curran Associates, Inc., 2010, pp. 1876–1884.

[168] N. Papernot, M. Abadi, Ú. Erlingsson, I. J. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," *ArXiv*, 2017.

[169] T. G. Dietterich, "Ensemble methods in machine learning," in *First International Workshop on Multiple Classifier Systems*, ser. MCS '00. Springer, 2000.

[170] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson, "Scalable private learning with PATE," in *ICLR*, 2018.

[171] Y. Long, S. Lin, Z. Yang, C. A. Gunter, and B. Li, "Scalable differentially private generative student model via pate," *ArXiv*, 2019.

[172] J. Jordon, J. Yoon, and M. Schaar, "Pate-gan: Generating synthetic data with differential privacy guarantees," in *ICLR*, 2019.

[173] A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser, K. Kurach, and J. Martens, "Adding gradient noise improves learning for very deep networks," *CoRR*, 2015.

[174] S. Smith, E. Elsen, and S. De, "On the generalization benefit of noise in stochastic gradient descent," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 9058–9067.

[175] S. Song, K. Chaudhuri, and A. D. Sarwate, "Stochastic gradient descent with differentially private updates," in *IEEE GlobalSIP*. IEEE, 2013, pp. 245–248.

[176] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *53rd Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 2015, pp. 909–910.

[177] J. Jälkö, A. Honkela, and O. Dikmen, "Differentially private variational inference for non-conjugate models," in *Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2017.

[178] X. Wu, F. Li, A. Kumar, K. Chaudhuri, S. Jha, and J. F. Naughton, "Bolt-on differential privacy for scalable stochastic gradient descent-based analytics," in *SIGMOD*. ACM, 2017, pp. 1307–1322.

[179] B. Wang, Q. Gu, M. Boedihardjo, L. Wang, F. Barekat, and S. J. Osher, "DP-LSSGD: A stochastic optimization method to lift the utility in privacy-preserving ERM," in *Proceedings of Mathematical and Scientific Machine Learning, MSML*, ser. Proceedings of Machine Learning Research, vol. 107. PMLR, 2020, pp. 328–351.

[180] J. Lee and D. Kifer, "Concentrated differentially private gradient descent with adaptive per-iteration privacy budget," in *SIGKDD*. ACM, 2018, pp. 1656–1665.

[181] J. Lee, "Differentially private variance reduced stochastic gradient descent," *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, pp. 161–166, 2017.

[182] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *NeurIPS*, ser. NIPS'13. Curran Associates Inc., 2013, p. 315–323.

[183] N. Roux, M. Schmidt, and F. Bach, "A stochastic gradient method with an exponential convergence _rate for finite training sets," in *NeurIPS*, vol. 25. Curran Associates, Inc., 2012.

[184] J. Konecný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *CoRR*, 2016.

[185] J. Konecný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *CoRR*, 2016.

[186] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *CoRR*, 2016.

[187] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *arXiv: Artificial Intelligence*, 2019.

[188] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *CoRR*, 2019.

[189] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, and M. Bennis, "Advances and open problems in federated learning," *CoRR*, 2019.

[190] Q. Li, Z. Wen, and B. He, "Federated learning systems: Vision, hype and reality for data privacy and protection," *CoRR*, 2019.

[191] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," *CoRR*, 2020.

[192] M. Song, Z. Wang, Z. Zhang, Y. Song, Q. Wang, J. Ren, and H. Qi, "Analyzing user-level privacy attack against federated learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, pp. 2430–2444, 2020.

[193] N. Papernot, M. Abadi, Ú. Erlingsson, I. J. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," in *ICLR*, 2017.

[194] A. Bellet, R. Guerraoui, M. Taziki, and M. Tommasi, "Fast and differentially private algorithms for decentralized collaborative machine learning," *CoRR*, 2017.

[195] M. A. Heikkilä, E. Lagerspetz, S. Kaski, K. Shimizu, S. Tarkoma, and A. Honkela, "Differentially private bayesian learning on distributed data," in *NeurIPS*, 2017, pp. 3226–3235.

[196] B. Li, C. Chen, H. Liu, and L. Carin, "On connecting

stochastic gradient MCMC and differential privacy," in *AISTATS*, ser. Proceedings of Machine Learning Research, vol. 89. PMLR, 2019, pp. 557–566.

[197] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, and Q. Yang, "SecureBoost: A lossless federated learning framework," *CoRR*, 2019.

[198] S. Hardy, W. Henecka, H. Ivey-Law, R. Nock, G. Patrini, G. Smith, and B. Thorne, "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," *CoRR*, 2017.

[199] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," *IEEE Intelligent Systems*, vol. 35, pp. 70–82, 2020.

[200] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, Y. Liu, and et al., "BatchCrypt: Efficient homomorphic encryption for cross-silo federated learning," in *USENIX*, 2020, pp. 493–506.

[201] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," *SIGSAC*, 2017.

[202] A. Jalalirad, M. Scavuzzo, C. Capota, and M. Sprague, "A simple and efficient federated recommender system," ser. BDCAT '19. ACM, 2019, p. 53–58.

[203] M. Ammad-ud-din, E. Ivannikova, S. A. Khan, W. Oyomno, Q. Fu, K. E. Tan, and A. Flanagan, "Federated collaborative filtering for privacy-preserving personalized recommendation system," *CoRR*, 2019.

[204] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009, crypto.stanford.edu/craig.

[205] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Comput. Surv.*, vol. 51, no. 4, 2018.

[206] Z. Brakerski, "Fundamentals of fully homomorphic encryption - a survey," *Electron. Colloquium Comput. Complex.*, vol. 25, 2018.

[207] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" ser. CCSW '11. ACM, 2011, p. 113–124.

[208] "Privacy enhancing technologies for solving the privacy-personalization paradox: Taxonomy and survey," *Journal of Network and Computer Applications*, vol. 171, 2020.

[209] V. Shejwalkar and A. Houmansadr, "Reconciling utility and membership privacy via knowledge distillation," *ArXiv*, 2019.

[210] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *IEEE SSP*. IEEE, 2016, pp. 582–597.

[211] N. Papernot and P. McDaniel, "On the effectiveness of defensive distillation," *ArXiv*, 2016.

[212] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE SSP*. IEEE, 2017, pp. 39–57.

[213] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *ArXiv*, 2013.

[214] P. Mohassel and P. Rindal, "ABY^3: A mixed protocol framework for machine learning." ACM, 2018.

[215] L. Phong and T. T. Phuong, "Privacy-preserving deep learning for any activation function," *ArXiv*, 2018.

[216] L. Phong and T. T. Phuong, "Privacy-preserving deep learning via weight transmission," *IEEE Transactions on Information Forensics and Security*, vol. 14, pp. 3003–3015, 2019.

[217] L. Lyu, J. Yu, K. Nandakumar, Y. Li, X. Ma, and J. Jin, "Towards fair and decentralized privacy-preserving deep learning with blockchain," *ArXiv*, 2019.

[218] M. Park, J. R. Foulds, K. Chaudhuri, and M. Welling, "Variational Bayes In Private Settings (VIPS)," *J. Artif. Intell. Res.*, vol. 68, pp. 109–157, 2020.

[219] Z. Yang, B. Shao, B. Xuan, E.-C. Chang, and F. Zhang, "Defending model inversion and membership inference attacks via prediction purification," *ArXiv*, 2020.

[220] L. Huang and N. K. Vishnoi, "Stable and fair classification," *arXiv preprint arXiv:1902.07823*, 2019.

[221] R. Berk, H. Heidari, S. Jabbari, M. Joseph, M. Kearns, J. Morgenstern, S. Neel, and A. Roth, "A convex framework for fair regression," *ArXiv*, 2017.

[222] A. B. Tsybakov, *Introduction to Nonparametric Estimation*. Springer, 2008.

[223] V. Bindschaedler, R. Shokri, and C. A. Gunter, "Plausible deniability for privacy-preserving data synthesis," *Proc. VLDB Endow.*, vol. 10, no. 5, p. 481–492, 2017.

[224] J. O. Abe and B. B. Ustundaug, "A data as a service (DaaS) model for GPU-based data analytics," *CoRR*, 2018.

[225] *Pseudonymisation techniques and best practices*, https://www.enisa.europa.eu/, European Union Agency for Cybersecurity, 2019.

[226] *Anonymisation: managing data protection risk code of practice*, https://ico.org.uk, ICO: Information Commissioner's Office, 2012.

[227] M. Finck and F. Pallas, "They who must not be identified—distinguishing personal from non-personal data under the GDPR," *International Data Privacy Law*, vol. 10, pp. 11–36, 2020.

[228] *Data Protection Directive*, https://eur-lex.europa.eu/, European Parliament and Council, 1995.

[229] *Data Protection Act (DPA)*, Parliament of the United Kingdom, 1998.

[230] *The EU General Data Protection Regulation 2016/679 (GDPR): Recital 26*, Parliament of the EU, 2016.

[231] *Amazon Machine Learning: Developer Guide*, Amazon Web Services, 2018.

[232] *Cloud Machine Learning Engine Documentation*, Google, 2018.

[233] *IBM Data Science and Machine Learning*, IBM, 2018.

[234] J. Jia and N. Z. Gong, "AttriGuard: A practical defense against attribute inference attacks via adversarial machine learning," in *USENIX*, 2018, pp. 513–529.