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 since 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 malicious 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 malicious leakage, followed by description of currently available defences, 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, Verifying Forgetting, 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, including: causes of involuntary leakage, the implications of these causes being exploited by malevolent users, the methods for measuring and preventing such attacks, and finally the 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 [1–11], whereas we examine the broader picture of data leakage.

This survey focuses on the inference time data leakage of trained models, such as they might be found in the wild, and the ways for defending against such kind of leakage and attacks (see Figure 1). For the training time interventions, we invite reader to check the relevant surveys on training-time attacks and defences in general [4], as well as specific types, such as adversarial attacks [12–15], or poisoning attacks [16].

Our contributions are as follows:

- *first comprehensive survey on data leakage*, including involuntary and malicious leakage methodology, prevention and defences, assessment metrics, and applications;

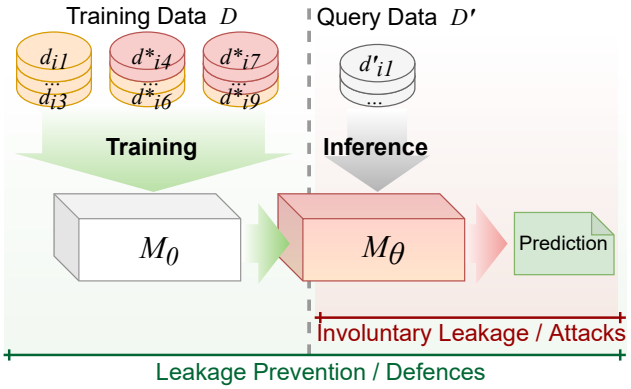


Fig. 1: We survey benign leakage and attacks at inference time, concerning the query dataset D', trained model M_θ, parameters θ, and predictions (or confidence scores) of M_θ.

- 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; and
- a summary of the challenges and open questions in the data leakage research field.

The paper is structured as follows: Section 2 provides definitions and notation, and discusses what private and sensitive data are in a variety of contexts. Section 3 covers causes of natural *involuntary* data leakage, whilst Section 4 covers *privacy attacks*. Sections 5 and 6 cover leakage prevention and defence mechanisms. Sections 7 provides an aggregated picture of currently available metrics for assessing leakage and privacy. Section 8 outlines applications. We conclude with remaining challenges and open questions in Sections 9 and 10.

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

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.

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 d_i 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**, or **query access**, assumes that the user controls the input and has access to the output of M_θ. **White-box access** assumes that the user has full access to M_θ, its input, output, architecture, and parameters θ. **Gray-box access** describes situations 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, *etc*.

2.1 What is personal (private) and sensitive data?

We distinguish between *personal*, *personal "sensitive"*, and *non-personal* data.

Personal data are defined in the Article $4(1)$ of the GDPR, [74], 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. E.g., a National Health Service (NHS) patients have their personal data processed during routine health checks.

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 and health-related data; data concerning person's sexual orientation. Such data require stronger safeguards for processing, storage, transfer, etc.

Non-personal data Any personal data fall under GDPR [74] protection, which implies a dichotomy – everything outside the scope of personal are *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 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 level of training sample d_i, such as patient record, user information, etc). The real-world risk of linkage back to identity is complex, and depends on multiple factors – the frequency of data points, the size of source datasets, and the availability of public data to support re-identification.

2.2 Leakage for different data types

Types of data leakage are largely data-specific (Table 1).

Data leakage in text data Examples are individuals' names, 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, [26, 34].

Data leakage in images Examples are individuals' faces or other identifying features, or embedded disclosive metadata (e.g. sensitive text on images). When training an ML model with sensitive image data, a generative model such as [75], trained on X-rays with hand-written notes on them or re-identifiable bone/denture implants, might occasionally reproduce an identifiable training image look-alike. This type of leakage could apply for other types of image text – names on security badges, car license plates, etc.

Data leakage in tabular data Examples of data leakage are similar to text data; however tabular datasets are constrained to predefined variables, so re-identification risks can be more accurately estimated according to statistical disclosure risks, based on features such as the data sensitivity, population size, zero-value entries, etc (see [76]). Whilst, for instance, re-identifying patients from rare combinations of diagnoses is possible, statistical disclosure control [76, 77] 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. Please refer to Table 1. Below is a detailed (but by no means exhaustive) overview of ML tasks and corresponding models researched for privacy-preservation and violation.

Classification is widely used for real world applications [78–80] and 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 [18, 21, 32–35, 37, 50, 51, 53–56, 59, 65, 81–85], CIFAR-100 [18, 20, 32, 34, 82, 86] and ImageNet [38, 51, 53], as well as more applied datasets/tasks, such as classification of potential customer value [20, 33, 34], classification of the income level based on the Census data [21, 50, 55, 59], diagnosing breast cancer [21, 50, 55] and classifying X-rays [81, 87]. There has been slightly less research on leakage from classifiers trained with tabular/mixed feature data [20, 30, 32, 34, 50, 55, 58, 64, 88, 89], and even less involving time-series. A number of works have targeted UCI's diabetes dataset [90], mainly for model extraction attacks [50, 55, 58], and also binary classification of text [30].

Regression/Prediction of unknown/future values of data samples has broad application in fields such as forecasting for financial and medical time-series. Leakage for regression was investigated for financial and medical time-series [50, 55, 58], numerical tabular data [18, 58, 58, 89], as well as mixed feature tabular data [18, 44].

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 problem

Type of Data Leakage	Type of Data				Type of Tasks				
	Images	Text	Tabular	Time-series	Classification	Regression	Generation	Seg.	MLaaS
Involuntary leakage:									
Overfitting Sec. 3.1	[17–21]	-	[18–22]	-	[17–21]	[18, 19, 21]	-	-	-
Memorization Sec. 3.2	[23–25]	[23, 26, 27]	[28]	-	[23, 24]	-	[25–28]	-	-
Feature Leakage Sec. 3.2	[22, 29, 30]	[26, 27, 30]	[22, 29, 30]	-	[22, 29, 30]	[22, 29]	[26, 27, 30]	-	-
Malicious Leakage / Attacks:									
Membership Infer. Sec. 4.1	[18–21] [31–40]	[30, 41, 42]	[18–21, 30] [32, 33, 43–45]	[44, 46] [47]	[18–21, 30] [32–36, 38, 41]	[18, 19, 45] [21, 44]	[30, 39, 40] [31, 37, 43]	[48]	[32–34, 41]
Model Extraction Sec. 4.2	[49–56]	[57]	[50, 55, 58]	-	[49–58]	[50, 55, 58]	-	-	[50, 55, 58]
Property Inference Sec. 4.3	[59–62]	[63]	[61–64]	-	[59–64]	-	-	-	-
Reconstruction Sec. 4.4	[22, 34, 50] [53, 54] [65–70]	[69]	[19, 22, 44, 71]	[44, 50]	[19, 22, 34, 50] [65, 67–70] [53, 54, 71–73]	[19, 44, 69] [22, 50]	-	-	[34, 50, 67]

TABLE 1: Taxonomy of the data leakage research, summarized by the type of the data and tasks. Sec. 2.2, 2.3, and 3.

[91, 92]. A good generative model should capture the underlying distribution of the real data [75], risking accidentally producing a doppelganger of a sensitive record (or a close enough sample). Simply sampling such a model could reveal individual records and attributes [25, 28]. A number of linkage attacks are developed for GANs [31, 39, 40, 83], with some defences proposed for all data types [28, 93–95].

Segmentation is crucial for computer vision tasks. The privacy risks of sharing a medical image segmentation model publicly have been studied, e.g. by [48], for linkage attacks, showing that most state-of-the-art semantic segmentation models are vulnerable. Segmentation models' vulnerability with less than white-box access remains unexplored.

Privacy preservation has been very sparsely verified for any other tasks, but may also be important for tasks such as clustering, translation, transfer, and collaborative learning.

2.4 How do user actions affect leakage?

We differentiate between two types of users, 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 / benign 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*.

3 Involuntary Data Leakage

Ways in which data leak without malicious user intervention include overfitting and memorization. Note that whilst overfitting implies some degree of memorization, memorization can occur while the model is still learning, i.e., before overfitting begins to happen [26]. One or both of these can be the cause of data leakage.

3.1 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 [17]. The formalization of the relationship between overfitting and privacy risks lacks research on exactly how overfitting aids various data and model attacks [18, 19]. So far it has been shown to be a sufficient but not strictly necessary condition for aiding membership inference and model inversion attacks (Sec. 4.1 and 4.4). Overfitting has also been shown to impact the privacy properties of classifiers; [19] formalizes the connection between the attacker model's inference advantage and the target model's generalization error for both membership inference and attribute inference attacks (Sec. 4.1).

For most models (but not all GANs [75]) overfitting can be prevented simply monitoring the generalization error.

Nonetheless, overfitting is but one of the possible reasons for data leakage. Even stable, well-generalized models can leak sensitive data, e.g., due to memorization [20, 21, 26]. Specific model types and architectures, as well as the training dataset features also have an impact on leakage [34].

3.2 Memorization

Memorization of specific training data samples occurs when the model assigns some sample a significantly higher likelihood than expected by random chance [26]. It raises serious privacy and legal concerns for sharing trained ML models publicly or providing them as a service [23, 96]. Potential risks include membership inference attacks, sensitive attribute and training dataset reconstruction [23].

Although there is little research on preventing memorization, some evidence suggests that data augmentation reduces (but does not eliminate) the memorization capacity of a network, whereas increasing the size of the architecture increases it [24]. Further, [24] estimates memorization in lower layers of CNNs, showing that fine-tuning the upper layers can be insufficient to prevent memorization. For GANs [75, 97], [28] suggests that limiting the number of noise vectors at training time reduces memorization.

Feature Leakage Can be defined as a special case of memorization, which occurs when sensitive attributes/features f_j^* (rather than data samples d_i) of the training data D are unintentionally memorized and revealed by the trained model at inference time.

Feature leakage implicitly enables property inference attacks (Sec. 4.3). For instance, [30] 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. 4.3 and 6.5). Feature leakage can be detected even when present in a few [98] or a single image [99], with common techniques (e.g., augmentations, unlearning) shown to not offer protection.

Interestingly, [22] 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 lead to feature/attribute leakage, and the model vulnerability even in the absence of the original training data. Importantly, [22] 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 Malicious Leakage / Privacy Attacks

To elaborate on Sec. 2.4, we define *malicious 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. However, the adversary's access to the trained model M_θ can be *black-box*, *white-box*, or anywhere in between. Some methods in this chapter also assume access to open-source data D' that might or might not come from a similar distribution to the original (potentially sensitive) data D'.

Attacks Exploiting Memorization and Overfitting Most attacks have a higher chance of success when overfitting comes into play [100, 101]. Overfitting alone has been shown to be enough for membership inference and more complex attribute inference attacks to succeed [18, 101] (see Sec. 4.1).

Other examples of exploiting memorization and overfitting apply to settings such as *collaborative* (also known as *federated*) *learning* [30], 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. 4.1) are simplified to a supervised learning problem.

Another malicious setting, explored by [23], features an adversary model provider (DaaS setting, Sec. 8.1), supplying the model M to a data owner, and receiving back a trained model M_θ. Model architectures designed by [23], 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.

4.1 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*[1] can be used to identify the individual records used for training open-access ML models. Hence, MIAs can threaten user data privacy, supporting the argument that ML models should be classified according to their sensitive training data content [96].

Formalization: MIA (see Fig. 2a) 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 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 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 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.

MIAs are usually performed either via *shadow models*[1] [32–34, 102, 103] or based on comparison of *metrics of the target model predictions* with an empirically chosen threshold. Such metrics can include prediction correctness[18, 38, 104, 105], confidence level [32], entropy [106], etc. For a detailed review and taxonomy of MIAs please refer to [107].

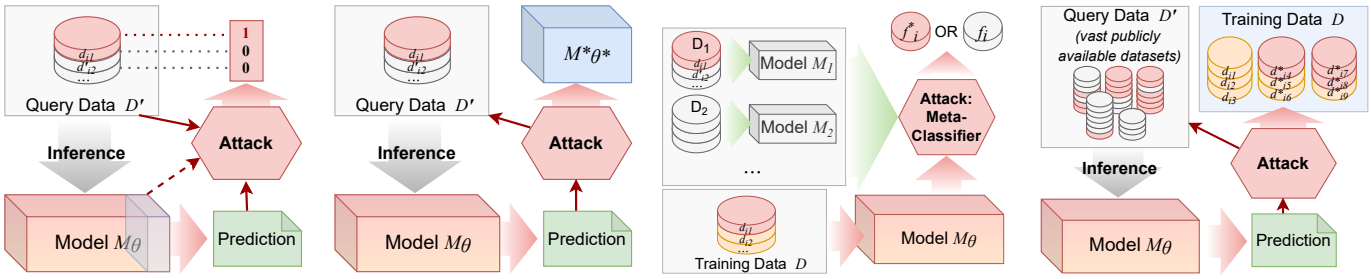
Risks of query access to M_θ. While large companies take advantage of their user databases and deploy ML models on a large scale, there is always a risk of re-identification or misidentification of a user, given (even just query) access to the model. Offering ML as a service, i.e. providing the trained models in open and semi-open access, increases such risks. Also, MIAs can be performed with just black-box access to the model [38], and without knowledge about the structure of the target model [32, 33]. For example, a metric-based [32] shows 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). Furthermore, [104] introduces two label-only MIAs that require no confidence scores of target model M_θ, instead directly assessing the robustness of the hard output labels under the input perturbations.

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, [34] 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, [18] and [21] provide theoretical and empirical evidence that overfitting alone is sufficient to increase the attacker's success in performing MIAs. The same is proven by [18] for the *attribute inference*[2] attacks. However controlling overfitting

1. *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.

2. *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 [44].

Name	Target	Assumptions	Input	Result
(a) Membership Inference, Sec.4.1	training samples d_i	black-box access to M_θ	query dataset D'	$d_i \in / \notin D$
(M.I.: Attribute Inference)			incomplete information about a data point d_i	missing details on d_i
(b) Model Extraction, Sec.4.2 (1)	trained model M_θ	model architecture or type known	labelled query dataset D'	model parameters θ
Sec.4.2 (2)		black-box access to M_θ, labelled query D'	labelled query dataset D'	model M_θ architecture
Sec.4.2 (3)	functionality of the trained model M_θ	black-box access to M_θ, unlabelled dataset D'	unlabelled query D'	a model $M^*_{\theta*}$, where $M^*_{\theta*}(x) \approx M_\theta(x)$
(c) Property Inference, Sec.4.3	whether M_θ exhibits feature f^*_i	white-box access to M_θ	weights θ of trained M_θ	$f^*_i \in / \notin D, M_\theta$
(d) Reconstruction, Sec.4.4	reconstructing D (fully or partially)	query access to M_θ, sometimes publicly available D' and M'_θ	publicly available or generated query dataset D'	training data D (full or partial)



(a) MIA (b) MEA (c) PIA (d) Reconstruction Attack

Fig. 2: Most common types of the inference-time attacks: **(a)** Membership inference attack (MIA), Sec. 4.1; **(b)** Model Extraction Attack (MEA), Sec. 4.2; **(c)** Property Inference Attack (PIA), Sec. 4.3; **(d)** Reconstruction Attack, Sec. 4.4.

(by minimizing generalization error) does not necessarily prevent a successful membership inference. Further, strategies for attacking well-generalised models via identifying the vulnerable target records and exploiting their influences on the target model are presented in [21]. Finally, [18] shows that the possibility of attribute inference implies the possibility of membership inference, thereby making a connection between MIAs and reconstruction attacks (see Sec. 4.4).

Applications of MIAs are numerous, both in terms of the types of the data and the models vulnerable to them. Applications which have been explored include: medical data [41], location data [47], including time-series [46], translation systems [42], collaborative learning (especially when the adversary performs as one of the participants) [33, 37], and generative models for various types of data synthesis. The latter are usually GAN-based, where the discriminator of a *shadow model* is often used for re-identifying the original training data samples in the query dataset [31, 39, 40].

Measuring the success of MIAs is easy compared to other privacy attacks. The common metric is re-identification score – the ratio of training and additional data samples in the query dataset, that have been correctly identified by MIA, or some modification of this metric [43, 45].

4.2 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 [49]); instead their end-goal is to steal the trained model functionality, see Fig. 2b.

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 the model parameters θ, assuming the model architecture (or at least the type) 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.

We now expand the above, ordering from greatest to least stringent requirements for 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, [58] 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 the target model θ. Under these assumptions, [58] proposes a method for efficiently stealing the hyperparameters θ' of the target models with both theoretical assessment and empirical evaluation on Amazon Machine Learning service. Meta-model introduced in [53] can predict some of the training parameters, i.e. batch size or optimisation algorithm.

A black-box attack on parameters θ is possible even without an 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. Two efficient ways of stealing hyperparameters with aforementioned assumptions are introduced in [50]. 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 [50].

2) Reverse engineering black-box models or functionally equivalent model extraction. Here the assumption of an adversary knowing the model architecture is relaxed, making the extraction attack much harder but not impossible [49, 51]. However, 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 [49] and [52]. 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_θ. The "copycat" network $M^*_{\theta^*}$ is then trained on the 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 [49] (on CNN models) show at least 93.7% attack accuracy on various problems (measured as the ability to mimic the target model), and 97.3% of the performance when applied to the Microsoft Azure Emotion API. [52] shows between 92% and 105% performance of the target model. They explain the additional improvement by the regularizing effect of training on soft-labels, introduced as the "soft targets" in [108].

3) Stealing functionality with minimal assumptions. The next assumption to relax is access to the unlabeled query data. [53] assumes no prior data knowledge and no knowledge of the class of M_θ. Instead, they train a meta-model capable of inferring the architecture of M_θ and training hyperparameters (e.g. the optimization algorithm and training dataset) from a series of queries, hence turning the black-box target models into white-box models, making the target models susceptible to all of the above mentioned attacks.

Last but not least , [51] 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. $M^*_{\theta^*}(x) = M_\theta(x)$, and faster querying, compared to competitors. It is achieved by a learning-based attack method, utilizing the target model as a guide 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, e.g. [54] 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 [55], both for implementing model extraction attacks and investigating possible defences. 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 – [57] finds that not only 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 [52] and [49] 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 this in practice.

4.3 Property Inference Attacks

Property Inference Attacks (PIAs) constitute a type of attacks where an attacker tries to extract a specific sensitive attribute or feature of interest f^*_i from a given target model M_θ. See Fig. 2c for the overall structure.

Assumptions and Formalization. PIAs are white-box attribute inference attacks, assuming complete access to M_θ and its training information. They are based on the principle that similar models, trained on similar datasets, exhibit similar properties. The goal is to build a meta-classifier MC, capable of telling whether a model M_i contains an attribute f^*_i. In order to train MC, an attacker trains a series of shadow classifiers, $M = \{M_1, .., M_n\}$ on some dataset, $D = \{D_1, .., D_n\}$, where only some of the subsets D_i exhibit the property f^*_i. The shadow models are not explicitly trained to learn the property f^*_i, but learn it as a consequence of the bias introduced in the dataset. At inference time the M_θ, trained on the original dataset D_x, is classified by MC as either exhibiting f^*_i or not. Weights and biases of M_θ are often used as features in MC training.

The first PIA was conducted on Hidden Markov Models and Support Vector Machines [64]. Weights of the hidden states served as inputs for the HMMs while weights and biases of support vectors were used to train the meta-classifier for SVMs. The logical transition of this approach to fully connected networks was shown in [59], 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, the meta-classifier itself was a network that learnt to account for all permutations of a particular neural network layer's weights and biases. This architecture was inspired by Deep Sets [109].

Poisoning Attacks is a special case of PIAs, where an adversary pollutes the data D or the model M during training, resulting in a bias in the M_θ output and in a leakage. (As *training time* attacks are out of scope here, see [16].)

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

4.4 Reconstruction / Model Inversion Attacks

Reconstruction / Model Inversion Attacks are methods for partial reconstruction of private datasets from aggregated publicly available information D', including open-access or query-only trained ML models M_θ. See Figure 2d.

Applications. Reconstruction attacks have been applied to a variety of scenarios, for instance, to the *federated learning setting*, [65], including an interesting application of GANs trained with a multitask discriminator that outputs the reality indicator for the data, its class, and user identity, [66].

A variety of applications of model inversion exist in the *general (centralized) setting*. E.g., [67] trains a second neural network as an inverse of the target model to perform the inversion, with its performance validated on Amazon Rekognition. Another GAN-based method, called *generative model-inversion attack* [68] uses GANs to learn the distributional prior of the data, which later guides the inversion process. Finally, [69] 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 [70], where the adversary probes a model with a particular data point (*MIA*, Sec. 4.1) or a particular set of data points (*Group MIA*) before and after training a model with additional data, to assess how the model's outcome changes as a result of online training. The attacks follow a general encoder-decoder structure.

Defences. Several defences have been proposed against reconstruction attacks: [110] 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 [72]. They use a simple additive noise method and, interestingly, they find that pairing it with another existing method NoPeekNN, [73], improves the defence. For classifier target models, [71] 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 Defences at Data Level

Defence methods aim to prevent malicious privacy attacks from succeeding. There are a few stages in model training and deployment where defences can be implemented. They can largely be dichotomized as applying augmentations at the *training data level* versus training, tuning, and designing models with inbuilt defence mechanisms – at the *model level*. Refer to Figure 3 for the proposed taxonomy of defences.

At data level, simply deleting sensitive features / entries can violate the data integrity and consistency and represent a privacy risk of its own, since the pattern of "missingness" might reveal some data properties. Hence data obfuscation and sanitization are often applied to mask, scramble, or overwrite the sensitive information with a realistic fake.

5.1 Data Obfuscation

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

For instance, [111] introduces an obfuscation function that addresses the trade-off between user privacy and service quality, which is dependent on the severity of the data perturbation. The adaptive mechanism anticipates and protects against optimal inference algorithms by designing a game between the obfuscation designer and the potential inference attack. Meanwhile, [112] 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 mitigating 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 model.

5.2 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 randomization algorithms satisfying the ϵ-differential data privacy criteria [113]. Data sanitization is often a natural precaution for *adversarial attacks* [114] (a large class of training time attacks, which is out of the scope of this paper).

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

Sanitization is a potential defence against the inference attacks on the social media networks, e.g., [115] utilizes a collective manipulation sanitization techniques on the user profile and friend connection data to prevent inference attacks from identifying users from their friend connections. Additionally, [116] 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 applications include self-destruct data-processing cycles [117]. These use threshold cryptography to overwrite data enough times 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 solve training data shortage problems for a wide variety of applications. It would allow open access to realistic synthetic data for researchers and facilitate internal data transfers within the organizations, where clients data cannot be shared across branches, divisions, hospitals, etc.

There is evidence suggesting some already successful applications - generating high quality synthetic patient data [91] 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 [118].

Several GAN-based models are designed to produce synthetic data with privacy guarantees: [119] is meant for generating time-series with DP guarantees, [120] is privacy-preserving under MIAs at a small performance trade-off.

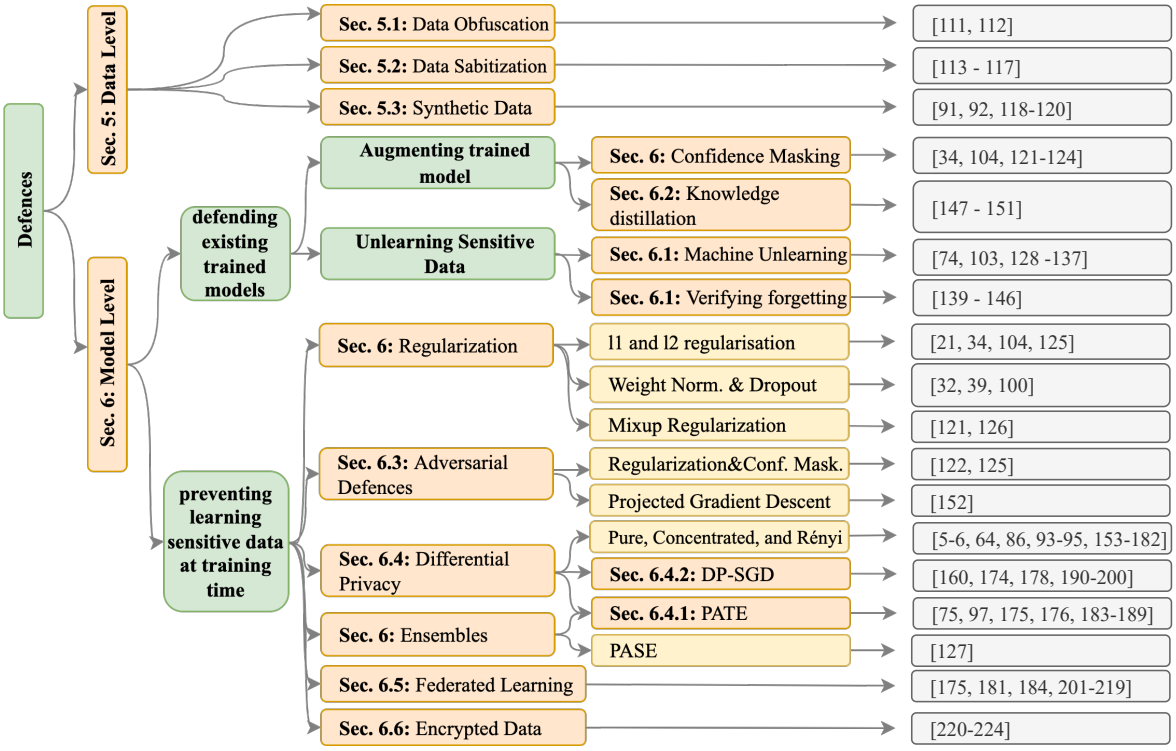


Fig. 3: Taxonomy of the defence sections (Sections 5 and 6).

Although synthetic data might seem appealing as a remedy for the sensitive data leakage problem, it is not in reality. A good generative model captures the underlying training data distribution, and might leak some of its properties into the generated data, enabling PIAs. Moreover, [92] finds that generative models tend to store richer information, enabling attribute inference. They show that generative models are vulnerable to MIAs, even when trained under DP guarantees, perhaps because memorization in models like GANs and VAEs cannot be fully eradicated. Thus far synthetic data generation with privacy guarantees remains elusive.

6 Current Defences at Model Level

Model defence techniques can be loosely separated into those protecting an existing trained model from leaking learnt sensitive information vs those preventing models from learning such information in the first place (Fig. 3).

As many of the inference time attacks rely on the confidence scores of the target model, defence can be simply perturbing these via confidence masking or regularization.

Confidence Masking usually comprises one of: hiding confidence scores of the model output and providing only the final label [104], or showing only top-K confidence [34, 121], perturbing the confidence scores directly [122]. Finally, prediction purification [123, 124] used against inversion and membership inference attacks replaces the confidence scores with reconstructed privacy-preserving representations.

Regularisation is a standard way to prevent overfitting and hence can be considered a defence against data leakage [34]. Standard types of regularization include l1 and l2 regularization [21, 34, 104, 125], weight normalization and dropout [32, 39, 100], and mixup regularization [121, 126].

Ensembles have also been used against privacy attacks, for example switching ensembles (PASE) have been shown to work against MIAs [127], and teaching ensembles (Sec. 6.4.1) work against a broad spectrum of attacks.

The rest of this section covers more complex defences.

6.1 Machine Unlearning / Forgetting

The General Data Protection Regulation (GDPR), [74], 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.

Machine Unlearning. This term was introduced by [128] 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. The first framework for instantaneous data summarization with machine unlearning

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.

using a resilient streaming algorithm, involving submodular optimization was presented in [129]; it comes with a constant factor approximation guarantee to the optimum solution. Further, [130] 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. An unlearning algorithm for linear regression methods, based on the projective residual update and use of synthetic data points, was introduced by [131]. Later, [132] proposed to limit the effect that a single data sample can have on the training process by training multiple models on subsets of the training dataset. This would imply storage and computational costs for retraining multiple models. In a similar attempt to limit the effect of a single data point on training [133] and [134] suggest a Newton-based estimation of the effect of such data point on the model predictions. This estimate can be immediately used for guiding the machine unlearning.

A comparatively computationally light method [135] suggests forgetting logit-based classifiers through linear transformation to the output logits. This would leave a data sample trace in the weights of a neural network model. [136] focuses on data removal from differential privacy perspective, and provides an algorithm for convex problems, based on a second order Newton update, layered over a DP DNN.

An algorithm proposed by [137] 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, [137] 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 to data to be forgotten. Further, [138] 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 [137]. This work introduces a new set of bounds that quantifies the average information per query an attacker can extract from a model.

Verifying forgetting. There is a difference between deliberately unlearning the traces of information from a model versus verifying it has indeed been forgotten (intentionally or otherwise). Additional considerations: *1)* Forgetting can occasionally happen on its own (*"catastrophic forgetting"* in reinforcement learning); *2)* different data samples bear varying amounts of unique information and contribute to the model final weights differently [139]; *3)* forgetting a specific data entry (a single person's entry) in the training set and consequently its trace in the system is non-trivial, because of the possible *trace overlap*. Trace overlap means the updates (the trace) extracted from the entry to-be-forgotten is equivalent to the trace obtained from another record that is still a legitimate training sample, hence this trace should be kept in the model. In light of GDPR [74] and the "right to be forgotten" there is a lot of focus is on formalization and good ways of performing verification of forgetting [140].

These intricacies have led to several directions of verifying forgetting. For instance, [141] 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 (See Sec. 4.1 about *shadow models*), 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, [142] 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 [143] by applying carefully engineered oblivious protocols for commonly used operations on trained networks. For network embeddings [144] investigates forgetting 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. For text generation models [145] suggests black-box model-auditing technique successful on well-generalized models (not overfitted to their training data), and [146] 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. Analyzing the risks of data leakage (under MIA) for black-box classifiers after machine unlearning, [103] finds that in some cases the *unlearnt* model can leak information about the forgotten data, even when the original *non-unlearnt* model did not leak such information.

6.2 Knowledge Distillation

Knowledge distillation has been actively used to compress models and thus facilitate deployment on resource constrained devices, however, it can also be applied to preserve privacy. Distillation for Membership Privacy uses distillation to train models with membership privacy by leveraging various sources of noise in the model distillation process [147]. Distillation-based methods based on the fast gradient sign method [148] and the Jacobian attack [149] 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, [150] showed that distillation fails to mitigate attacks proposed in [151].

6.3 Adversarial Defences

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

Adversarial Defences for MIAs. A lot of research is conducted on protecting against black-box MIAs with adversarial examples. For example, [125] anticipates a MIA, and regularizes the target model during 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 claims membership privacy and good target model generalization. Memguard, [122], has been the first defence with formal utility-loss guarantees

against black-box MIAs. It adds carefully designed noise to the target model confidence score vectors, turning these into adversarial examples, making MIA classifier vulnerable.

Limitations and Risks. Interestingly, some of the proposed adversarial defences, such as projective gradient descent (PGD) adversarial training [152], on the contrary increase the model's susceptibility to MIAs. In theory, many of the privacy attacks could be used as adversaries to improve against during training of M_θ. However, [36] has proven that using some state-of-the-art attacks as adversaries during training can weaken the defence against these and the new attacks compared even to original undefended models.

6.4 Training with Differential Privacy

Differential privacy (DP) gathers confidential user data for analysis without compromising the confidentiality of individual users. Formally defined in 2006 by [153]: 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. Thus, unlike some 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 budget" ϵ. This guarantee is 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, [154], 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 [155] 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 be improved using two different data batching techniques proposed by [156] as an extension to classic CDP. Further quantitative results for CDP were provided in [157] by redefining the concept of DP in terms of the Rééнyi 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 [158], 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 [158] definition applies to one moment at a time. Further, [159] proves a tight upper bound on RDP for subsampling in DP, it also generalizes the results of the *moments accounting technique* [160], to any RDP algorithm. The *moments accounting technique* [160] 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 DP principal projection at the input layer. Tool-wise, this framework is built in Tensorflow [161].

Differential Privacy Surveys. In addition to some of the aforementioned previous privacy reviews, [4, 5], there exist several surveys focusing specifically on DP, from early works such as [162], to [163, 164]. We refer readers to these specialised surveys for a more complete picture of the field.

Applications of DP with respect to different tasks. The more traditional applications of DP, outlined in [153] are the DP online learning, [165–167] and DP empirical risk minimization, [168–174]. However, the range of learning tasks that DP was applied to has widened and now includes nearly anything from the federated ML setting [175] to recurrent language models [176], and even GANs [93, 95], with specific DP-GAN applications for generating time-series [94, 119], and tabular mixed feature datasets [177].

Evaluation and Utility-Privacy Trade-Off of DP methods. The utility vs privacy trade-off is one of the most important topics in DP, partly due to the lack of formal utility-loss guarantees [122, 178]. Evaluation of privacy guarantees for DP is more established compared to some of the other defence methods. However, despite the provable upper bounds on maximum privacy loss, there is still 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 to select large values for ϵ to show reasonable utility scores [86, 179]. Practically, [86] finds a huge gap between the guaranteed upper bounds on privacy loss, and the effective privacy loss measured using inference attacks. Moreover, there is no agreed upon threshold for ϵ, at which privacy guarantees are considered meaningless. The empirical assessment shows that for an acceptable utility level privacy guarantees are often meaningless, although the observed level of leakage under the inference attacks is still low. Advancing further, [180] offers more empirical assessment of leakage under inference attacks, considering single and joint decoding (MIA, see Sec. 4.1 for single data instance vs a subset of instances at a time), finding the joint decoding is more powerful, and offering a method to empirically choose the size of privacy budget ϵ.

Some research has been conducted on replacing privacy-utility trade-off with privacy-computational cost trade-off [181]. Proposing an SGD-based DP (Sec. 6.4.2) for recurrent language models in a federated learning setting (Sec. 6.4.2).

Risks. DP has been shown to be insecure under PIAs (see Sec. 4.3), because of the different types of data leakage considered by PIA and DP [64]. Further, (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. According to [182], no mechanism has been proposed for detection and reporting of this kind of leakage.

For neural networks, two more recent approaches of implementing DP are particularly relevant. We cover them in the next two subsections. As sub-classes of DP methods these share their general limitations and vulnerabilities.

6.4.1 DP: Private Aggregation of Teaching Ensembles

Private Aggregation of Teaching Ensembles (PATE), [175, 183, 184], and its modification *PATE-G*, [185], is a subset of differential privacy techniques based on the teacher-student approach, using ensemble methods ([186]) aggregation and some of the GAN-based architecture for PATE-G, [75, 97].

At training time an ensemble of teacher networks is trained on disjoint subsets of the training dataset with strong privacy guarantees, then the student network aggregates the teacher network's knowledge in a noisy fashion, i.e. the student black-box-queries the teacher ensemble, receiving the noisy labels. PATE methods train only on labelled training data, whilst PATE-G also uses unlabelled data (via GANs or Virtual Adversarial training). At inference time, only the student model is used, and as it has never seen the training data, the noisy aggregation of the teacher ensemble provides privacy guarantees [182].

The scalability of PATE methods has been practically confirmed by [187] (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 withholding 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. Although more classical works are applied to classifiers, [188, 189] focus on the data generation with DP guarantees. G-PATE [188], trains a student-generator with an ensemble of teacher discriminators. Note, G-PATE and PATE-G should not be confused – G-PATE is merely one of the PATE-G methods. PATE-GAN [189] trains a student classifier on synthetically generated data, using a noisy aggregation of the teacher-discriminator labels.

6.4.2 DP: the Gradient Descent Perturbations

Neural network training relies on gradient descent, so adding noise is a popular technique for better generalization [190, 191] and (with appropriate calibration) for ensuring differential privacy [178]. 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 privacy-preserving techniques [174, 192–194].

More recent advances of the noisy SGD include extension with the moments accounting technique [160], a scalable and computationally efficient "bolt-on" output perturbation technique by [195], and DP-LSSGD [196], 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 [197], and [198] applying the control variates technique [199, 200] to stochastic gradient descent update are both compatible with zCDP (see Sec. 6.4 for more details on zCDP).

6.5 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 [201–203], thus, potentially mitigating risks of the direct data leakage. It is considered a popular but not completely reliable defence against MIAs [20, 35].

Surveys. There are a number of surveys covering FL in general, [204–207]. We would like to refer the reader to [208] which focuses mainly on privacy concerns for FL.

FL vulnerability to privacy attacks. FL is sometimes offered as a solution to the problem of balancing user data privacy requirements (such as GDPR [74]) with the benefits of learning from multiple data sources, [204]. However, FL does not provide foolproof privacy guarantees. Successful white-box MIAs have been performed by [20] 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, [208] concludes that classic FL frameworks are often vulnerable to inference and poisoning attacks (Sec. 4.1), also expressing concerns with the current methods of defences against these attacks for FL.

Malicious servers An alternative to a malicious user is a malicious server provider aiming to steal client's data. Recently [209] proposed the first ever attack from the perspective of such a malicious server. It uses a GAN [75, 97] 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 [175, 184, 210–213]. Some concerns remain on privacy-utility trade-offs [181], and property inference attacks for groups of records (rather than a single record) [30].

Other defences for FL. An important point raised in [208] is the lack of clarity on whether certain defences, such as adversarial defences, can be used for FL systems. A traditional defence for FL is homomorphic encryption (Sec. 6.6), used to mask the local gradient updates, either individually [214, 215] or in batches for reducing computation costs [216].

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 [204, 207, 215, 217]. Nevertheless, according to [207], there are not many FL applications that explicitly focus on privacy preservation. Still, there are some examples of privacy-preserving recommendation systems [218, 219], that rely primarily on data encryption (see the next section) for their privacy guarantees.

6.6 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)*, [220], allows for any computable function to perform on the encrypted data [221].

Surveys. Homomorphic Encryption is a vast and well-established field, hence, for the sake of brevity, we refer the reader to the relevant surveys [221, 222].

Limitations. Operating on encrypted data could alleviate privacy issues, but unfortunately its low efficiency often makes FHE impractical in the real world [223]. However, newer advances, e.g., somewhat homomorphic encryption aim to improve efficiency – refer to [224] for more details.

6.7 Other Privacy-preserving ML

Various other methodologies and applications exist for protecting against malicious leakage at model level. PRADA [56] defends against model extraction attacks by flagging multiple queries made against a model when they deviate against general inference behaviour. Privacy preserving alternatives to SGD, introduced in [225] and [226], apply to a case when multiple data owners wish to train models combining their data keeping individual privacy, based on sharing weight parameters instead of gradient updates. FP-PDL [227] 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 [228] 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.

7 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 malicious/involuntary leakage in question. Further, any knowledge about the exact type and architecture of the attack might be crucial for the defence.

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 [26, 27], explained in Sec. 3.2 estimates model memorization for text data (thus far, no extensions to other data types exist). An assessment proposed by [24] 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 [43, 45]. Further, [106] introduces a *privacy risk score* to measure an individual sample's likelihood of being a training member, to identify samples with high privacy risks under MIAs.

Assessing data leakage for defence purposes Examples of this include Kolmogorov-Smirnov distance used for verifying forgetting in [141], metrics proposed by [103] for assessing machine unlearning leakage under MIAs, as well as some work on estimating the Bayes risk of the system [29], improving upon more naïve min-entropy approaches. More specifically, [29] proposes a number of Bayesian metrics based on universally consistent nearest neighbor rules, from which metrics that converge the fastest should be selected. This provides an estimate of the Bayes risk of the model i.e., the error of the optimal classifier for predicting a sensitive attribute given an output observation from the model.

Learning metrics as a fairness constraint Most literature in fair ML deals with learning fair classifiers. 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 [229] 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, individual or hybrid fairness, although they have not received a lot of attention so far [230].

Metrics in Differential Privacy In this setting, due to the provable privacy upper bounds, empirical assessment of both utility and quality guarantees is possible [86, 118]. The Rényi Divergence can be used as a metric to bound any arbitrary privacy loss [86]. 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 [118]. Evaluation of DP-SYN was done using carefully crafted metrics based on ML (misclassification rate), statistics (Total Variation Distance [231] 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 [232]).

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), [182], 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.

8 Applications

8.1 Data as a Service (DaaS)

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. [233], provide no leakage assessment. Significant efforts are afoot to create national research infrastructures across the world to support data-driven research. Organizations such as Health Data Research UK and Research Data Scotland design services for identification of health research datasets, their description, permissions, and accessibility.

Federated access models are favoured by data holders, with data scientists invited to access data within Trusted Research Environments, for example Data Safe Havens. However, with the intellectual and economic benefits of more access to data comes an escalating risk of data leakage, driving persistent privacy concerns.

The current official protocols in the UK, rely on statistical disclosure control [76, 77], data pseudoanonymization [234],

and true anonymization [235], since fewer legal restrictions apply to anonymized data. However, from a legal perspective, anonymized data is a gray area [236]. In fact, regulations such as Data Protection Directive (1995) [237], Data Protection Act 1998 (DPA) [238], and GDPR [74], do not require strictly risk-free data protection, but the risk of re-identification should be mitigated to the extent when it is remote. GDPR does 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*", [239]. However it applies to pseudo- and non- anonymized data, often more useful in practice, and preferable for statistical analysis and ML applications in DaaS setting.

If DaaS using linked and unconsented public data, under the GDPR safeguards and standards for privacy is to continue, techniques to mitigate data linkage are imperative. Hence custodians considering DaaS, especially with sensitive datasets, 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 reliable checks on models exported from DaaS facilities, including more reliable methods of defence from privacy attacks, such as MIAs and PIAs (Sec. 3, 4.1 and 4.3).

8.2 ML Models as a Service (MLaaS)

Machine Learning as a Service (MLaaS) represents an extended privacy risk, further to that posed by DaaS. The development of ML models can risk perpetuating bias, state intrusion, inequalities, and erosion of privacy. Whilst the separation of source data from MLaaS could ameliorate data leakage concerns, the outputs, decisions, and unintended applications of MLaaS complicate tracing potential leakage.

Certain settings of MLaaS, including federated learning, can be vulnerable to inference type attacks, e.g., MIAs [33, 112], with defence mechanisms shown to mitigate those risks explored mainly for classification [112, 156]. Moreover, [67] showed that Amazon Rekognition, a commercial MLaaS API, can be vulnerable to model inversion attacks. Research in MLaaS data safety remains important for understanding the risks posed by models as they evolve on exposure to new data. Presently a number of providers such as Amazon ML [240], Google Cloud [241], and IBM [242] are providing MLaaS for public and commercial use. Sections 3 – 6 cover the implications of open-access-sharing ML models trained on sensitive data and current defences.

8.3 ML models in Mobile Applications

ML methods are often used to support mobile applications. Thus, privacy attacks, e.g., MIAs [34], attribute inference attacks, and PIAs (Sec. 4.1 and 4.3) are a possibility. Sensitive information could be anything from the full user profile or location [46, 47] to gender and sexual orientation [243].

Federated learning (see Sec. 6.5 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., [243] and [217], this field is still somewhat in its adolescence.

9 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. 4.1) are not well investigated for tasks such as regression or segmentation, MEAs (Sec. 4.2) have not been verified for generative models, and PIAs have only been applied to classification. 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 [92]. 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 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 mainly 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 is a promising direction for privacy-preserving FL; however, its practical implementation is not straightforward and requires compute power and 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 and universally applicable software packages based on already existing research are lacking. This results in an opportunity to develop mechanisms for transparent reporting and create robust software that can help bridge the gap between theory and practical utility.

10 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, prompting new efforts to protect against them, resulting in a constant adversarial game. This survey unifies and summarizes current research into inference-time information leakage in ML, both involuntary and malicious, as well as the means currently available to measure and prevent such leakage. This results in a rich comprehensive taxonomy of the broad field of privacy in ML.

We find, first of all, that 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 data leakage. Our survey reveals opportunities to improve the means to measure, detect

and report sensitive data leakage. Secondly, privacy attacks exploration has been uneven in its coverage of ML tasks and architectures, data types, and attack structures. Finally, we find that most available defences are case-specific, and scaling to larger datasets with performance guarantees remains a challenge. Overall this indicates that leakage, privacy, and the necessary defenses, remain areas which are fertile for further research and development.

ACKNOWLEDGMENT

This work is supported by iCAIRD, funded by Innovate UK, UK Research and Innovation (UKRI)[104690]. S.A. Tsaftaris acknowledges support by Canon Medical / Royal Academy of Engineering Research Chair, 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] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, and Z. Lin, "When machine learning meets privacy: A survey and outlook," *ACM Comput. Surv.*, vol. 54.

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

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

[14] 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.

[15] K. Ren, T. Zheng, Z. Qin, and X. Liu, "Adversarial attacks and defenses in deep learning," *Engineering*, vol. 6, pp. 346–360, 2020.

[16] M. A. Ramírez, S.-K. Kim, H. A. Hamadi, E. Damiani, Y.-J. Byon, T.-Y. Kim, C.-S. Cho, and C. Y. Yeun, "Poisoning attacks and defenses on artificial intelligence: A survey," *ArXiv*, vol. abs/2202.10276, 2022.

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

[18] 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.

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

[20] 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.

[21] 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," *arXiv*, 2018.

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

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

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

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

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

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

[28] V. Nagarajan, C. Raffel, and I. J. Goodfellow, "Theoretical insights into memorization in GANs," *NeurIPS Workshop*, 2018.

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

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

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

[32] 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.

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

[34] R. Shokri, M. Stronati, C. Song, and V. Shmatikov,

"Membership Inference Attacks Against Machine Learning Models," in *IEEE SSP*, 2017, pp. 3–18.

[35] 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.

[36] L. Song, R. Shokri, and P. Mittal, "Privacy risks of securing machine learning models against adversarial examples," in *SIGSAC*. ACM, 2019.

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

[38] A. Sablayrolles, M. Douze, C. Schmid, Y. Ollivier, and H. Jegou, "White-box vs black-box: Bayes optimal strategies for membership inference," vol. 97, 2019.

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

[40] 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.

[41] 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 Trans on Computational Social Systems*, vol. 6, pp. 907–921, 2019.

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

[43] 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.

[44] 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.

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

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

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

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

[49] 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.

[50] 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.

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

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

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

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

[55] 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.

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

[57] 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.

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

[59] 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.

[60] 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.

[61] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *ICML*. PMLR, 2019, pp. 634–643.

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

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

[64] 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.

[65] Z. He, T. Zhang, and R. B. Lee, "Model inversion attacks against collaborative inference," in *Computer Security Applications Conference*. ACM, 2019, p. 148–162.

[66] 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.

[67] 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.

[68] 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.

[69] L. Zhu, Z. Liu, and S. Han, in *NeurIPS*.

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

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

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

[73] 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.

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

[75] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, in *NeurIPS*.

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

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

[78] 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.

[79] 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.

[80] 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.

[81] 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.

[82] L. Zhu, Z. Liu, and S. Han, in *NeurIPS*, pp. 14 774–14 784.

[83] 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.

[84] 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.

[85] 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.

[86] B. Jayaraman and D. Evans, "Evaluating differentially private machine learning in practice," in *USENIX*, 2019.

[87] 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.

[88] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *SIGSAC*. ACM, 2015.

[89] 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.

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

[91] 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.

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

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

[94] 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.

[95] 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.

[96] M. Veale, R. Binns, and L. Edwards, "Algorithms that remember: model inversion attacks and data protection law," *Philos. Trans. R. Soc. A: Mathematical, Physical and Engineering Sciences*, vol. 376, 2018.

[97] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, in *NeurIPS*.

[98] Y.-Y. Yang and K. Chaudhuri, "Understanding rare spurious correlations in neural networks," *arXiv*, 2022.

[99] J. Hartley and S. A. Tsaftaris, "Measuring unintended memorisation of unique private features in neural networks," *arXiv*, 2022.

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

[101] 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, 2020.

[102] R. Shokri, M. Strobel, and Y. Zick, "On the privacy risks of model explanations," in *AIES '21: AAAI/ACM*, M. Fourcade, B. Kuipers, S. Lazar, and D. K. Mulligan, Eds. ACM, 2021, pp. 231–241.

[103] M. Chen, Z. Zhang, T. Wang, M. Backes, M. Humbert, and Y. Zhang, "When machine unlearning jeopardizes privacy," *CoRR*, 2020.

[104] C. A. Choquette-Choo, F. Tramèr, N. Carlini, and N. Papernot, "Label-only membership inference attacks," in *ICML*, M. Meila and T. Zhang, Eds.

[105] J. W. Bentley, D. Gibney, G. Hoppenworth, and S. K. Jha, "Quantifying Membership Inference Vulnerability via Generalization Gap and Other Model Metrics," 2020.

[106] L. Song and P. Mittal, "Systematic evaluation of privacy risks of machine learning models," in *USENIX Security Symposium*, M. Bailey and R. Greenstadt, Eds., 2021, pp. 2615–2632.

[107] H. Hu, Z. Salcic, G. Dobbie, and X. Zhang, "Membership inference attacks on machine learning: A survey," 2021.

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

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

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

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

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

[113] 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.

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

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

[116] P. Tambe and D. Vora, "Data sanitization for privacy preservation on social network," *ICACDOT*, pp. 972–976, 2016.

[117] 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*, pp. 90–99, 2019.

[118] 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.

[119] 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.

[120] 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.*, 2021.

[121] J. Li, N. Li, and B. Ribeiro, "Membership inference attacks and defenses in classification models," in *CODASPY*, A. Joshi, B. Carminati, and R. M. Verma, Eds. ACM, 2021, pp. 5–16.

[122] 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.

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

[124] L. Hanzlik, Y. Zhang, K. Grosse, A. Salem, M. Augustin, M. Backes, and M. Fritz, "Mlcapsule: Guarded offline deployment of machine learning as a service," in *IEEE CVPR*, 2021.

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

[126] Y. Yin, K. Chen, L. Shou, and G. Chen, "Defending privacy against more knowledgeable membership inference attackers," in *SIGKDD*, F. Zhu, B. C. Ooi, and C. Miao, Eds. ACM, 2021, pp. 2026–2036.

[127] R. Izmailov, P. Lin, C. Mesterharm, and S. Basu, "Privacy leakage avoidance with switching ensembles," in *IEEE MILCOM*, 2021, pp. 981–986.

[128] Y. Cao and J. Yang, "Towards making systems forget with machine unlearning," in *IEEE SSP*. IEEE, 2015.

[129] 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.

[130] 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.

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

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

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

[134] 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.

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

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

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

[138] 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.

[139] 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.

[140] S. Garg, S. Goldwasser, and P. N. Vasudevan, "Formalizing data deletion in the context of the right to be forgotten," in *EUROCRYPT*. Springer, 2020.

[141] X. Liu and S. A. Tsaftaris, "Have you forgotten? A method to assess if machine learning models have

forgotten data," *MICCAI*, 2020.
[142] A. Golatkar, A. Achille, A. Ravichandran, M. Polito, and S. Soatto, "Mixed-privacy forgetting in deep networks," *ArXiv*, vol. abs/2012.13431, 2020.
[143] J. Liu, M. Juuti, Y. Lu, and N. Asokan, "Oblivious neural network predictions via MiniONN transformations," *SIGSAC*, 2017.
[144] M. Ellers, M. Cochez, T. Schumacher, M. Strohmaier, and F. Lemmerich, "Privacy attacks on network embeddings," *CoRR*, 2019.
[145] C. Song and V. Shmatikov, "Auditing data provenance in text-generation models," ser. KDD '19. ACM, 2019.
[146] 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.
[147] V. Shejwalkar and A. Houmansadr, "Reconciling utility and membership privacy via knowledge distillation," *ArXiv*, 2019.
[148] 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.
[149] N. Papernot and P. McDaniel, "On the effectiveness of defensive distillation," *ArXiv*, 2016.
[150] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE SSP*, 2017.
[151] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *ArXiv*, 2013.
[152] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *ArXiv*, 2018.
[153] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, pp. 211–407, 2014.
[154] 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.
[155] C. Dwork and G. N. Rothblum, "Concentrated differential privacy," *CoRR*, 2016.
[156] 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.
[157] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," *ArXiv*, 2016.
[158] I. Mironov, "Rényi differential privacy," *IEEE CSF*, pp. 263–275, 2017.
[159] Y. Wang, B. Balle, and S. P. Kasiviswanathan, "Subsampled renyi differential privacy and analytical moments accountant," in *AISTATS*.
[160] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," *SIGSAC*, 2016.
[161] 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/
[162] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*. Springer, 2008.
[163] Z. Ji, Z. C. Lipton, and C. Elkan, "Differential privacy and machine learning: a survey and review," *ArXiv*, 2014.
[164] 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.
[165] P. Jain, P. Kothari, and A. Thakurta, "Differentially private online learning," in *COLT*, 2012.
[166] A. Thakurta and A. D. Smith, "(Nearly) Optimal Algorithms for Private Online Learning in Full-information and Bandit Settings," in *NeurIPS*, 2013.
[167] N. Agarwal and K. Singh, "The price of differential privacy for online learning," in *ICML*. PMLR, 2017.
[168] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *J. Mach. Learn. Res.*, vol. 12, pp. 1069–1109, 2011.
[169] D. Wang, M. Ye, and J. Xu, "Differentially private empirical risk minimization revisited: Faster and more general," in *NeurIPS*, 2017, pp. 2722–2731.
[170] Y. Wang, D. Kifer, and J. Lee, "Differentially private confidence intervals for empirical risk minimization," *J. Priv. Confidentiality*, vol. 9, 2019.
[171] D. Wang, C. Chen, and J. Xu, "Differentially private empirical risk minimization with non-convex loss functions," in *ICML*.
[172] 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.
[173] K. S. S. Kumar and M. P. Deisenroth, "Differentially private empirical risk minimization with sparsity-inducing norms," *CoRR*, 2019.
[174] 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.
[175] J. Hamm, Y. Cao, and M. Belkin, "Learning privately from multiparty data," in *ICML*, ser. JMLR Workshop and Conference Proceedings, vol. 48. JMLR, 2016.
[176] H. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *ICLR*, 2018.
[177] A. Triastcyn and B. Faltings, "Generating differentially private datasets using GANs," *CoRR*, 2018.
[178] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith, "Calibrating noise to sensitivity in private data analysis," *J. Priv. Confidentiality*, vol. 7, 2016.
[179] B. Jayaraman and D. Evans, "When relaxations go bad: "differentially-private" machine learning," *ArXiv*, 2019.
[180] R. Balu, T. Furon, and S. Gambs, "Challenging differential privacy:the case of non-interactive mechanisms," in *ESORICS*. Springer, 2014, pp. 146–164.
[181] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private language models without losing accuracy," *CoRR*, 2017.
[182] M. Abadi, Ú. Erlingsson, I. J. Goodfellow, H. B. McMa-

han, 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.
[183] K. Nissim, "Smooth sensitivity and sampling in private data analysis," in *STOC*. ACM, 2007, pp. 75–84.
[184] M. A. Pathak, S. Rane, and B. Raj, "Multiparty differential privacy via aggregation of locally trained classifiers," in *NeurIPS*.
[185] 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.
[186] T. G. Dietterich, "Ensemble methods in machine learning," in *First International Workshop on Multiple Classifier Systems*. Springer, 2000.
[187] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson, "Scalable private learning with PATE," in *ICLR*, 2018.
[188] Y. Long, S. Lin, Z. Yang, C. A. Gunter, and B. Li, "Scalable Differentially Private Generative Student Model via PATE," *ArXiv*, 2019.
[189] J. Jordon, J. Yoon, and M. Schaar, "Pate-gan: Generating synthetic data with differential privacy guarantees," in *ICLR*, 2019.
[190] 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.
[191] S. Smith, E. Elsen, and S. De, "On the generalization benefit of noise in stochastic gradient descent," in *ICML*, vol. 119, 2020, pp. 9058–9067.
[192] S. Song, K. Chaudhuri, and A. D. Sarwate, "Stochastic gradient descent with differentially private updates," in *IEEE GlobalSIP*. IEEE, 2013, pp. 245–248.
[193] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *53rd Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 2015.
[194] 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.
[195] 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.
[196] 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*.
[197] J. Lee and D. Kifer, "Concentrated differentially private gradient descent with adaptive per-iteration privacy budget," in *SIGKDD*. ACM, 2018.
[198] J. Lee, "Differentially private variance reduced stochastic gradient descent," *International Conference on New Trends in Computing Sciences*, pp. 161–166, 2017.
[199] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," 2013.
[200] N. Roux, M. Schmidt, and F. Bach, in *NeurIPS*.
[201] J. Konecný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *CoRR*, 2016.
[202] 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.
[203] H. B. McMahan, E. Moore, D. Ramage, and B. A. Arcas, "Federated learning of deep networks using model averaging," *CoRR*, 2016.
[204] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *arXiv: Artificial Intelligence*, 2019.
[205] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *CoRR*, 2019.
[206] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, and M. Bennis, "Advances and open problems in federated learning," *CoRR*, 2019.
[207] Q. Li, Z. Wen, and B. He, "Federated learning systems: Vision, hype and reality for data privacy and protection," *CoRR*, 2019.
[208] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," *CoRR*, 2020.
[209] 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.
[210] 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.
[211] A. Bellet, R. Guerraoui, M. Taziki, and M. Tommasi, "Fast and differentially private algorithms for decentralized collaborative machine learning," *CoRR*, 2017.
[212] 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.
[213] B. Li, C. Chen, H. Liu, and L. Carin, "On connecting stochastic gradient MCMC and differential privacy," in *AISTATS*.
[214] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, and Q. Yang, "SecureBoost: A lossless federated learning framework," *CoRR*, 2019.
[215] 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.
[216] 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.
[217] 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.
[218] A. Jalalirad, M. Scavuzzo, C. Capota, and M. Sprague, "A simple and efficient federated recommender system," ser. BDCAT '19. ACM, 2019, p. 53–58.
[219] 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.

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

[221] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Comput. Surv.*, 2018.

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

[223] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" ACM, 2011.

[224] N. Kaaniche, M. Laurent, and S. Belguith, "Privacy enhancing technologies for solving the privacy-personalization paradox: Taxonomy and survey," *J. of Network and Comp. App.*, vol. 171, 2020.

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

[226] L. Phong and T. T. Phuong, "Privacy-preserving deep learning via weight transmission," *IEEE Trans on Information Forensics and Security*, vol. 14, 2019.

[227] 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.

[228] 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.

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

[230] 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.

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

[232] V. Bindschaedler, R. Shokri, and C. A. Gunter, "Plausible deniability for privacy-preserving data synthesis," *Proc. VLDB Endow.*, vol. 10, p. 481–492, 2017.

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

[234] *Pseudonymisation techniques and best practices*, https://www.enisa.europa.eu/, EUAC, 2019.

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

[236] 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.

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

[238] *Data Protection Act*, Parliament of the UK, 1998.

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

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

[241] *Cloud ML Engine Documentation*, Google, 2018.

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

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

Dr Marija Jegorova is a postdoc at Facebook AI (London) and a former postdoc at School of Engineering, the University of Edinburgh. Her PhD focused on realistic artificial data synthesis for ML applications to robotics. Current research interests include semi-supervised learning and video emotion recognition.

Dr Chaitanya Kaul is a Research Associate in the School of Computing Science at the University of Glasgow. His research interests include medical image analysis, attention models for deep neural networks, generative modelling, and 3D shape analysis. He works on private and explainable machine learning.

Dr Alison Q. O'Neil is a Senior Scientist in the AI Research Team at Canon Medical Research Europe and Honorary Research Fellow at the University of Edinburgh. She leads a team working on machine learning for industrial healthcare applications, such as medical imaging, NLP, and electronic health records.

Dr Charlie Mayor is a Manager at Safe Havens Scotland, NHS Greater Glasgow and Clyde. Current research interests include development and deployment of privacy-preserving methodologies and safety checks in order to meet the potential future requirements for the safeguards of the Safe Havens.

Dr Alexander Weir is a Senior Technical Manager at Canon Medical Research (Europe). He leads specialist project research teams developing new medical systems for patient medical data management, deploying and developing state of the art applications to industrial healthcare technology.

Prof. Roderick Murray-Smith is a Professor of Computing Science at the University of Glasgow, in the *Inference, Dynamics and Interaction* group. He works in the overlap between machine learning, interaction design and control theory. His more recent interests include quantum imaging, multimodal sensor-based interaction with mobile devices, mobile spatial interaction, Brain-Computer interaction, and nonparametric machine learning.

Prof. Sotirios A. Tsaftaris is a Chair in Machine Learning and Computer Vision at the University of Edinburgh, and holds the Canon Medical/Royal Academy of Engineering Research Chair in Healthcare AI. His research interests include machine learning, computer vision, distributed computing and applications in healthcare and other domains.