Computers & Security 137 (2024) 103605

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

# Preserving data privacy in machine learning systems

Soumia Zohra El Mestari *, Gabriele Lenzini, Huseyin Demirci

*SnT, University of Luxembourg, 2 Av. de l'Universite, Esch-sur-Alzette, L-4365, Luxembourg*

ARTICLE INFO

*Keywords:*
Privacy enhancing technologies
Trustworthy machine learning
Machine learning
Differential privacy
Homomorphic encryption
Functional encryption
Secure multiparty computation
Privacy threats

ABSTRACT

The wide adoption of Machine Learning to solve a large set of real-life problems came with the need to collect and process large volumes of data, some of which are considered personal and sensitive, raising serious concerns about data protection. Privacy-enhancing technologies (PETs) are often indicated as a solution to protect personal data and to achieve a general trustworthiness as required by current EU regulations on data protection and AI. However, an off-the-shelf application of PETs is insufficient to ensure a high-quality of data protection, which one needs to understand. This work systematically discusses the risks against data protection in modern Machine Learning systems taking the original perspective of the *data owners*, who are those who hold the various data sets, data models, or both, throughout the machine learning life cycle and considering the different Machine Learning architectures. It argues that the origin of the threats, the risks against the data, and the level of protection offered by PETs depend on the data processing phase, the role of the parties involved, and the architecture where the machine learning systems are deployed. By offering a framework in which to discuss privacy and confidentiality risks for data owners and by identifying and assessing privacy-preserving countermeasures for machine learning, this work could facilitate the discussion about compliance with EU regulations and directives.

We discuss current challenges and research questions that are still unsolved in the field. In this respect, this paper provides researchers and developers working on machine learning with a comprehensive body of knowledge to let them advance in the science of data protection in machine learning field as well as in closely related fields such as Artificial Intelligence.

## 1. Introduction

Machine Learning (ML) systems process data to learn valuable patterns that solve and improve the performance of a specific task at hand.

These systems have demonstrated high performance and accuracy, which have led them to become drivers of innovation in several disciplines and sectors such as computer vision (Chai et al., 2021), autonomous transportation, health care (Ghassemi et al., 2020), biomedicine (Mamoshina et al., 2016) and law (Surden, 2014).

Nowadays, ML systems are at the core of common technologies such as automatic handwriting, natural language recognition (Md Ali et al., 2021), speech processing (Vila et al., 2018), and biometric data analysis.

A performant ML system needs two critical resources: (i) massive volumes of datasets from multiple sources to represent data at various circumstances to be used in the training, and (ii) powerful computational resources to build the models. The use of such resources raises several technical, social, and ultimately legal demands: besides other technical requirements (*e.g.*, robustness), ML systems are demanded to be transparent and fair (*e.g.*, free from bias in decision-making) and capable of protecting the data of the various parties involved, mainly data owners and model users.

This latter requirement, *i.e.*, data protection, links to properties such as data confidentiality and privacy which are particularly relevant to achieve lawful data processing whenever current laws require that "appropriate technical and organisational measures"[1] be in place to mitigate the risk of leaking confidential or private information.

Such legally inspired requirements are no longer outside the scope of security experts. Existing regulations – for instance, the Health Insurance Portability and Accountability Act (HIPAA), the Cybersecurity Law of China, the California Consumer Privacy Act (CCPA), or, quite

* Principal corresponding author.
*E-mail address:* soumia.elmestari@uni.lu (S.Z. El Mestari).

[1] GDPR, *id. at* 2, Art 32.

https://doi.org/10.1016/j.cose.2023.103605
Received 3 March 2023; Received in revised form 31 August 2023; Accepted 15 November 2023

Available online 29 November 2023

relevant in European legal ecosystems, the General Data Protection Regulation (GDPR),[2] the Data Governance Act,[3] and the Artificial Intelligence Act[4]— will change the way one operates with ML systems mainly because violating the requirements may lead to high fines.

However, it is also not yet clear which privacy-preserving solution is best suited for a given scenario and how to apply it correctly. A deeper and more mature understanding of threats, risks, and mitigation that certain tools can offer become a necessary know-how for those operating in the ML sector.

This work systematises current knowledge about the preservation of privacy in ML workflows taking the perspective of those aware of the obligations imposed by legal requirements who wish to understand the risks that threaten data protection in the context of ML systems.

Although we organise our approach in light of the current legal landscape —and herein we refer primarily to the European legal framework whose provisions are centred on data subjects (*i.e.*, the identifiable natural persons to whom the data relates)[5] and on data processors and controllers (*i.e.*, the first being the entity and is legally bound to protect the personal data of the data subject)[6]— this work's discussion is from the point of view of the *data owners*, while it refers to other parties, such as the ML-based service providers, as *computational parties*. In Section 3 we clarify why we adopted a different terminology when talking about the roles involved in a ML process. We will also explain how data owners and computational parties map onto the legal roles of data subjects, data processors, and data controllers. Here, we just anticipate that by offering a perspective centred on data owners (and data owners can be, in certain circumstances, data controllers), this work can offer better and novel insights to those who are obliged by law to guarantee data protection during processing about the risks and about the technologies, like Privacy Enhancing Technologies (PETs), offered to mitigate them. This work reviews existing privacy and confidentiality issues and discusses current Privacy Enhancing Technology (PET) solutions to mitigate them and under which conditions. Ultimately, this work helps ML service providers reach a higher level of awareness about data protection issues and achieve a better presumption of compliance with current data protection, governance, and trustworthy AI regulations.

One consideration is due concerning the need for a systematisation of knowledge work like ours on PETs for ML which takes the perspective of data owners. One could think that the privacy-preserving problem can be solved by resorting to commercial ML infrastructure and service providers, such as Amazon and Microsoft, who are obliged to comply while offering a variety of cloud-based solutions to build ML models at first and then provide prediction services through cloud-deployed models. However, such a setting is often unsatisfactory, as it requires data owners to trust big tech service providers, while, on the other end, it opens further privacy risks with copies of potentially private data being used and in third-party servers.

Even if one resorted to the PETs offered today by the technical community, the issue remains that such current technologies may be insufficient to provide the required guarantees if applied disregarding several key factors that define how and where they are going to operate. For example, privacy preservation as a principle, according to the EU perspective, is one of the key elements of trustworthy ML (*e.g.*, see (Content European Commission, 2019)) and thus PETs may not be enough unless an argument is made about the trustworthiness of their implementations and reasons for their wide adoption.

## 2. Position and organisation of work

This work aims to gather existing knowledge on data protection for ML systems, and specifically on *confidentiality* and *privacy* as we will explain in Sections 4 and 5. Since different particular aspects of ML such as the nature of the pipeline, the architectural choices, and the different actors heavily affect how we see data protection risks, we start our work by highlighting them in Section 3. Our work offers a systematic review of the data protection safeguards applicable to ML systems which are collectively called PETs; it discusses their capabilities as defence mechanisms against existing confidentiality and privacy threats, and mainly from the perspective of *data owners*. As we anticipated in Section 1, this perspective is also that of the various computational parties (who can be data controllers) who wish to provide lawful data protection guarantees to the individuals who trust them with their data. Sections 8 and 9 conclude our work by providing an overview of open research directions in the field.

*Contribution and originality of this work* The protection of data privacy throughout the ML pipeline faces many challenges. First, the implementation of solutions against privacy threats in ML is still in its infancy and is expected to advance quickly in the near future. Thus, our discussion refers to the current state-of-the-art but also enriches the discussion by carefully pointing out certain aspects which are more fluid, for instance, the availability of stable libraries (Section 7 and Section A).

The applications of PETs span the entire ML workflow.

To this end, our considerations on what solutions are capable of protecting the data should be contextualised depending on each stage of the ML pipeline. Existing defence mechanisms must be discussed contextually: for each particular threat, their guarantees depend on whether in the ML workflow they apply along with the trust assumptions and the inherited performance degradation that may result from applying them (Section 6).

The current work is not the first state-of-the-art review of security and privacy in machine learning. In (De Cristofaro, 2020) Cristofaro presents a broad review of privacy in machine learning, adversarial models and attack methods, and prevention techniques at a very high level. The survey (Xue et al., 2020) explores the security issues of machine learning in a comprehensive way, analysing existing attacks, defence techniques, and security evaluation methods. It covers various aspects of machine learning security, including training set poisoning, backdoors, adversarial examples, model theft, and sensitive training data recovery. Real-world attacks are reviewed to highlight practical implications, and suggestions for security evaluations and future directions in machine learning security are provided. In (Liu et al., 2021a), the authors provide a state-of-the-art review of privacy issues and solutions for machine learning, highlighting the unique challenges of privacy preservation in the context of machine learning. Their work covers the categories of private machine learning, machine learning-aided privacy protection, and machine learning-based privacy attacks. Song et al. presents a benchmark for membership inference attacks by including non-neural network-based attacks. They also propose a privacy analysis metric called the privacy risk score, which helps to identify samples with high privacy risks and investigate factors that contribute to these risks (Song and Mittal, 2021). In this regard, another authoritative contribution is that of Papernot et al. (2018a); it discusses issues of security and privacy for machine learning, mainly referring to the famous Confidentiality Integrity Availability (CIA triad) security model, but also presents them with insights into emerging properties in fairness, accountability, and transparency. Papernot et al. clarifies the attack surfaces of a ML data processing pipeline, the trust assumptions placed onto the pipeline's relevant actors, the capabilities that an adversary puts in place to subvert the data processing's security

[2] Regulation (EU 2016/679 of the European Parliament and Council of the 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).

[3] Regulation of the EU Parliament and of the Council on European data governance (Data Governance Act), COM/2020/767 final, 2020/0340 (COD).

[4] Regulation of the EU Parliament and of the Council on European on Artificial Intelligence (Artificial Intelligence Act) and amending certain union legislative acts, COM/2021/206 final, 2021/0106 (COD).

[5] *id. at* 2 Art. 4(1).

[6] *id. at* 2 Art. 4(8).

and privacy during the training and inference phases, and finally the goals of the adversary in how it can violate instances of security and privacy properties, herein paired as confidentiality and privacy, and integrity and availability. Our organisational structure inevitably shares elements due to a common subject and, precisely, the ML pipeline, the threats and risks to security and privacy, and a discussion of current defence mechanisms. However, this work, compared to those previous works, is focused exclusively on confidentiality and privacy, diving deeper into threats, risks, and mitigation tools than (Papernot et al., 2018a). By taking the viewpoint of data owners wishing to protect their data according to current data protection regulations, our work offers a knowledge that can be directly translated into practise to those who operate in the ML sector.

Due to this focus and this perspective, our discussion on confidentiality and privacy reveals that they are qualities whose protection is more complex than it appears in other works such as (Papernot et al., 2018a; Al-Rubaie and Chang, 2019). By spelling out the different phases more clearly and distinguishing the various actors that sit on those phases, a more detailed overview of potential threats concerning the data flow and the malicious parties involved emerges. The different phases are also not independent, and a threat to one phase can increase the likelihood or severity of a threat to another. Furthermore, the attack surface against privacy and confidentiality is larger than one may initially think, and understanding what risk of attacks exists requires understanding the complex distributed architecture that today supports the training and inference phases of a machine learning system. In addition, it requires understanding the role played by the various distinguished actors —those we call computational parties— who can sit at different nodes in such an architecture and who can or cannot behave honestly. Finally, the pools of technologies that are available today, mainly differential privacy, homomorphic encryption, and trusted hardware environments, have evolved quickly, and there is more to say that can be of interest to data subjects and controllers. The article of Papernot et al., citing a work that has been influential on the subject, suggests using differential privacy without giving much detail. Instead, we discuss this measure, but more extensively, while taking a broader focus on other techniques as well. And since in this manuscript, our aim is to generate a reference guide for those who plan to implement practical privacy-preserving ML workflows, we also mention how combined environments such as OpenMined toolkits[7] offer a portfolio of tools and libraries.

As an additional, although subsidiary, contribution, this work also discusses how the regulation on data privacy links to the wider concern about how to achieve a trustworthy ML. In Section 10 we provide an outline of the legal and ethical framework by discussing the trustworthiness requirements and the challenges of applying them in an ML setting, along with the importance of the privacy angle as an unlocking factor for the data.

Section 8, concludes the paper and discusses the current research landscape in this domain, offering scientists from both the legal and computational sciences an overview of future directions in this field.

## 3. Preliminaries

In ML, the threats to data privacy change depending on several elements: the ML pipeline; the threat actors; the threatened parties (mainly the data owners); the architectural choices for the computations; and the phase of the pipeline that we are studying.

The workflow of a typical ML system includes three phases: data pre-processing; model building; and model serving. The typical data stakeholders are data owners, *computation parties* and *communication parties* (the latter will remain out of the scope of this work). Depending on the role of each of those stakeholders and the interactions between them, they have different protection expectations.

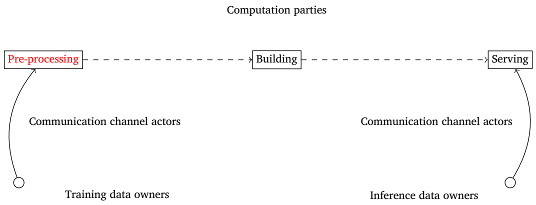

**Fig. 1.** The ML pipeline, and the actors involved. Data owners send their data, possibly via communications channels. Data are processed along the ML workflow, operated by the computation parties usually responsible for processing the data.

A machine learning system and its phases can be deployed locally, in the cloud, or in a combination of the two. Such choices will change how one looks at privacy and confidentiality requirements and relevant properties. Let us see those elements in more detail.

### 3.1. Machine learning pipeline

A typical ML system's workflow has three main phases:

**Data pre-processing:** In this step the data is cleaned and pre-processed to be used by the models. The various cleaning and preprocessing operations are task- and data-dependent.[8]

**Model building:** The model, during this step is trained to fit the data. The output of the machine learning model can be described as a function $h_\theta(x)$ where $x \in X$ is the input and $\theta \in \Theta$, the set of model parameters that will be specified during training. The type of optimisation depends on the task and the learning strategy: supervised, unsupervised, or reinforcement learning.

**Model Serving:** It includes deployment and inference operations. This phase focusses on serving a fully trained model as a service to external users for inference.

### 3.2. Data owners and other parties

The typical machine learning workflow involves the participation of different parties whose contributions are defined mainly by the context of deployment, the goal to achieve, and the architectural design of the workflow.

We categorise them into two main types: the *data owners* and the *computation parties*. There are also other parties which are out of scope and thus play a minor role in our discussion, such as the *communication channel actors*.[9] Fig. 1 shows the ML workflow and where the actors operate there.

Data owners are either *training data owners* or *inference data owners* (also called model/service customers, since they are the parties to which the model is served, and since this consumption is in fact an inference operation that requires them to send their data to get inferences as a service, they are inference data owners). In this work, these are considered the threatened parties which face the risk of their data being exposed or inferred by threat actors.

The computational parties are those responsible for performing the various operations of data preprocessing, model building, and serving. From the perspective of data owners, these parties are generally

[7] https://www.openmined.org/.

[8] As we will see, this phase may include already the usage of some privacy-enhancing techniques to protect the privacy and/or the confidentiality of the training data.

[9] The potential malicious actions that can be carried out by these actors are not specific to machine learning design choices but rather common for any secure network exchange of data.

considered as potential threat actors, especially if the entity doing the computation is a third-party cloud server.

Not necessarily, all such parties are distinct, and so the deployment setting is a key component to deciding whether and how the data should be protected. For example, if all computational roles are performed by the same entity that holds ownership of both training and inference data, then privacy and confidentiality concerns become minimal.

### 3.3. Workflow actors between the technical roles and the legal terminology

Interpreting privacy issues in the machine learning workflow under the EU legal instruments requires a careful usage of terminology when describing the actors. In most data protection regulations in the EU such as the GDPR, prominently, actors are the data subject, data controller, and data processor. The term "*data controller*" in GDPR[10] refers to "the entity that determines the purposes for which and the means by which personal data are processed. The controller must determine who is responsible for compliance with the data protection rules and how data subjects can exercise the rights in the rules".[11] The "data processor"[12] is "the party that processes personal data only on behalf of the controller".

The basic conditions for qualifying as a processor are, on the one hand, being a separate legal entity with respect to the controller and, on the other hand, processing personal data on his behalf. This processing activity may be limited to a very specific task or context, or may be more general and extended. The definition of processor envisages a wide range of actors that can play the role of processor ("… a natural or legal person, public authority, agency, or any other body...").[13]

Thus, for the terminology used in this paper, the *computational party* is the party with computational capabilities that performs computations / calculations on the data, whether those calculations are pre-processing, training, or inference computations. From a legal perspective, this computational party can be either a processor or both a processor and a controller at the same time. It qualifies as a processor if the legal requirements of the processing are dictated by another separate party known as the controller. The computational party takes both roles (processor and data controller) if it is the party that processes the data under a set of terms determined by itself.

Similarly, the term "data subject" merely refers to the person to whom the identification data relates. We quote "the 'data subject' is an identifiable natural person who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier, or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural, or social identity of that natural person".[14]

In the same context, "personal data" refer to "the data that can identify a given data subject". In a similar legal role, the Data Governance Act (DGA) uses another terminology, namely, "*data holder*" to refer to data owners who may own personal and non-personal data, we quote " 'data holder' means a legal person, including public bodies and international organisations, or a natural person, who is not a data subject in relation to the data in question, who is entitled under applicable Union or applicable national law to provide access to certain personal data or non-personal data or to disclose such data".[15]

The DGA definition of data holder aims to be more inclusive than the previous GDPR term of data subject; it opens the discussion of whether a 'data holder' is a 'data controller.' Therefore, although our work may facilitate the discussion about legal compliance by understanding the different privacy guarantees that each PET offers, we chose to use the term "data owner" to refer to the party that holds (or has) the data prior to any processing; it can legally refer to a data subject, a controller, or a data holder depending on the nature of the data to process and the legal instrument used to interpret the case. In addition to the legal instrument used to study each case, it is important to clarify that the legal roles depend on the legal nature of the data (*e.g.*, sensitive,[16] personal data,[17] etc.) and while this categorisation is important in the study of compliance, it has little relevance in the study of the technical feasibility, performance, and the guarantees offered by PETs. The privacy guarantees offered by PETs and even the implementation of PETs can be technically built agnostically of these legal types. To explain, when adopting one of the privacy enhancing techniques listed in this work in a machine learning pipeline, the legal nature of the data does not affect the possibility of adopting those PETs, nor does it affect the technical privacy guarantees offered by these PETs. Thus, we adopt a legally neutral terminology to refer to the actors.

### 3.4. Architectural choices

The architectural choices made when designing the ML pipeline will determine the positions and number of different parties and define whether they are data owners or computation parties; therefore, they will help define the threat model and spot the threat points.

These architectural choices are about the ML pipeline phases of reference; on whether the computations of those phases are performed locally or outsourced fully (resp. partially) to other servers; and on whether the architecture is centralised or distributed.

In a fully outsourced computation, all the steps in an operation are executed by an external computation party, for example, when a training operation is performed entirely in a cloud service. Respectively, a partially outsourced computation is where a number of steps of an operation are performed locally and the rest is performed in the cloud. For instance, in horizontal federated learning, some training operations are performed locally by the data owners, while the task of updating the global model is performed by a central aggregating party.

These architectural choices are influenced by a number of factors such as the availability of the computation power, the financial resources, the organisational and legal requirements, *etc.* We list them in order according to the ML phase involved:

**During the data pre-processing phase:**

- Data pre-processing can be performed fully locally by the data owner.
- Data preprocessing can be partially outsourced, where heavy algorithm preprocessing is performed by a cloud service.
- Data preprocessing is completely outsourced.

**During the model-building phase:**

- The training data owner party trains the complete model locally.
- The training data owner party trains only a partial model locally. This partial model can be used to refine a global ML model collaboratively or distributedly.
- The training data owner party sends its data to third-party entities that have the computational resources required to train the model.

**During the model-serving phase:**

- The model customer receives the trained model directly and performs the inference computation locally if such a computational resource is available.

[10] GDPR Art. 4.

[11] Article 29 Working Party Opinion 1/2010 on the concepts of 'controller' and 'processor' (WP 169).

[12] GDPR Art. 4.

[13] *id. at* 11.

[14] According to Article 3 (1) of Regulation (EU) 2018/1725.

[15] Art.2 DGA-Definitions.

[16] data Article 4(13)(14) and (15), and Article 9 and Recitals (51) to (56) of the GDPR.

[17] Art. 4 GDPR Definitions and Art. 9 GDPR Processing of special categories of personal data.

- The model customer uses a third-party facility where the trained model is already deployed to query it to receive the prediction service.

## 4. Privacy and confidentiality

Although in a legal context there is not a crisp distinction between privacy and confidentiality when referring to data protection and to appropriate measures to ensure it —for instance the GDPR mention confidentiality, while the 2002 Directive on privacy and electronic communications refer to privacy— in this work, more technical and orientated to ML technologies, we need to slightly redefine the terms.

Even if preserving confidentiality has been seen often as a sufficient condition to preserve privacy (*i.e.*, preserving confidentiality implies data privacy) (Gürses, 2010), or although the general interpretation suggests that confidentiality is about data to be protected while privacy concerns identifying people, when discussing the terms in reference to ML learning, we need to discern what we can learn from the sheer data (confidentiality) and what can be learnt by accessing statistics about the data (privacy). In fact, a ML model can be used to learn about a certain person, whose data have been used to build it, although there is no explicit reference in the model that points out the person and its data.

The way in which we distinguish confidentiality and privacy is not uncommon (*e.g.*, see (Choquette-Choo et al., 2021a)).

*Confidentiality* of the data ensures that there is no explicit disclosure of the data or of certain parts of the data. In other words, the confidentiality of the data is preserved if the data is never accessed in its raw form. Thus, the need for a party to keep their data secret from other parties is a confidentiality need. In this work, when the entire data point is kept secret from other parties (generally, when using cryptographic tools such as homomorphic encryption or functional encryption, etc.), it is referred to as a strong confidentiality guarantee. Whereas, when only certain features of the data are hidden, for *e.g.*, when anonymising personal identifiers such as names or age or race, we refer to it as a limited confidentiality guarantee since only the secrecy of those features is guaranteed, not the entire data point.

*Privacy* of the data is protected when it is ensured that the adversaries cannot leak sensitive pieces of information about the data through an intended interaction with the threatened party (such as the model deployment party), such as inferring the participation of a certain data point in the training of a given model, also known as membership inference attacks (Shokri et al., 2017). In other words, privacy is what can be revealed from sharing statistics about the data (Choquette-Choo et al., 2021a).

## 5. Threat model

Our threat model clarifies who are the threat actors depending on the elements in Section 3. It also defines the goals of these threat actors.

A threat is defined from the point of view of data owners. Since we are interested in examining risks from this standpoint, we consider out-of-scope threats regarding security aspects such as robustness and service availability, as well as privacy matters regarding the model and its parameters, which are about the protection of intellectual property.

From Section 3, we remind that there are two types of data owners: (a) training data owners whose data are used to train and build the model; and (b) inference data owners that interact with the already deployed model as service customers. From the perspective of both, the threats come from the rest of the stakeholders within the ML workflow —and therefore, are seen as threat actors—, with whom they interact directly or indirectly.

A direct interaction between a data owner and a threat actor is when the data are shared directly and explicitly between the two entities, for *e.g.*, when an inference data owner (a model customer) sends his data to the cloud inference facility to obtain predictions as a service. Instead, an indirect interaction occurs between a data owner and a threat actor when the data owner shares the data implicitly through the sharing of statistics, aggregations, or knowledge extracted from the data generally as a result of an inference computation, for *e.g.*, the probability vectors obtained when using an ML classifier.

Thus, a direct interaction between data owners and threat actors yields confidentiality risks because the data are shared directly, while an indirect interaction between them exposes privacy risks.

From the viewpoint of data owners, the threats against privacy and confidentiality change depending on the interaction points between them and threat actors, and whether the interaction is direct or indirect.

Table 1 summarises the existing threats that occur during the training and inference phases when those are operated on the cloud and during their interaction with model customers.

### 5.1. Confidentiality risks

Having the data owners as separate entities from the computation facilities requires the data to be uploaded to the servers of the computing facilities, hopefully through a secure channel. However, even if the transmission channel is secured and the data remain encrypted during transmission, the computation facility in most cases will only process them in plaintext format after being decrypted. Hence, the data will reside in third party servers in its original pain-text form.

This is the biggest type of threats, since the user loses any governance over his private information once decrypted in third-party servers which exposes the data to all possible threats and attacks performed by any malicious actors from within or outside this third-party server.

### 5.2. Privacy risks

In privacy attacks, the attacker aims to gain knowledge beyond what the machine learning service offers as inference results. The attacker's visibility of the model ranges from a black-box access such as the work of Shokri et al. (2017), where the attacker has only access to the inference results produced by the model, to a white-box access (Yeom et al., 2017; Szegedy et al., 2013; Nasr et al., 2019) where the attacker has partial or full access to the models' parameters or additional information about the model, such as the adversary having access to one of the partial local models used in a federated learning setting, or knowledge about explanation vectors about the model decisions, or even the architecture of the model in the case the model was a deep neural network, etc.

There are various privacy attacks against machine learning systems such as membership inference attacks (Shokri et al., 2017; Bernau et al., 2019; Jia et al., 2019a; Li et al., 2020) and model inversion attacks (Fredrikson et al., 2015; He et al., 2019; Wu et al., 2016).

#### 5.2.1. Membership inference attacks

In a membership inference (Shokri et al., 2017; Bernau et al., 2019; Jia et al., 2019a; Li et al., 2020) attack (see Fig. 2) the adversary tries to identify whether a given data point $(Dc_i, P_i)$ or a sample of data points were part of the training data set $D_{train}$ used to train a given model $h_\theta$. Revealing that a certain record was used to train a specific machine learning model is a strong indication of private information leakage about the individual data points in the training set. *e.g.*, knowing that a medical record was used to train a machine learning model deployed for diabetes detection can reveal that the person concerned has diabetes.

These attacks exist in the black-box and the white-box modes. In the black box mode, the attacker has only a query access to the model without any inner information about it, hence only the query results are used to infer the membership of data points within the original training set. In the white-box setting, the attacker has either access to the inner details (description) of the model or can download it locally.

**Table 1**

Points of interaction between the data owners and the other threat actors and potential threats against privacy and confidentiality from their perspective. Herein a ✓ indicates the presence of a threat, an ✗ the absence. When there is no interaction between the owner of the data and the threat actor, it is marked as not applicable (N/A).

| Threat actors / Data owners | Cloud data pre-processing/ training facility | | Cloud inference facility | | Other training Data owners (federated learning setting) | | Model customers | |
|---|---|---|---|---|---|---|---|---|
| | privacy risks | confidentiality risks | privacy risks | confidentiality risks | privacy risks | confidentiality risks | privacy risks | confidentiality risks |
| Training data owners | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | ✗ |
| Inference data owners | N/A | N/A | ✗ | ✓ | N/A | N/A | N/A | N/A |

The first membership inference attack was designed by Shokri et al. (2017) and implemented based on the concept of shadow models, which are models trained on some attacker dataset that is similar to the target model training set. The attack is modelled as a binary classification task trained on the prediction of shadow models in the adversary data set. Multiple works have then followed extending the attack to various settings such as in federated learning (Melis et al., 2018), transfer learning (Zou et al., 2020), generative models (Hayes et al., 2017), language models (Carlini et al., 2019; Song and Shmatikov, 2019a) and speech recognition models (Shah et al., 2021).

The membership inference attack can be combined with other privacy attacks such as model stealing or model reconstruction attacks (Wu et al., 2020a; Tramèr et al., 2016; Milli et al., 2019), where the adversary attempts to design a model $h_{\tilde{\theta}}(x)$ using black-box access to a target model $f$ with $\tilde{f}$ being an approximation or even a perfect match to $f$. The reconstructed model then is a perfect candidate for a performant shadow model that mimics the targeted model, and hence results in a more efficient membership inference attack.

This type of attacks relies on the fact that the models behave differently when seeing new data points compared to the training data. Models tend to be more confident about the training data, hence the prediction loss is significantly lower than the prediction loss of an unseen data point.

The poor generalisation of models is one of the main factors that improves the accuracy of membership inference attacks. An over-fitted model tends to hard memorise the training data points rather than learning the underlying distribution. Yeom et al. (2017); Song and Shmatikov (2019b) proved that overfitting is a sufficient condition to perform a membership inference attack, but not a necessary one. Furthermore, Long et al. (2018) demonstrated that even well-generalised models are prone to membership inference attacks due to the unintended memorisation problem that occurs when training machine learning models (Thakkar et al., 2021), where models memorise rare or unique sequences of training data exposing minorities in datasets to the risk of such attacks.

The architecture of the model, the type of model, and the characteristics of the data set, such as the dimensionality of the output and the uniformity within each class, are also factors that can affect the accuracy of the attack (Truex et al., 2018), (Shokri et al., 2017). Complex models exhibit higher precision in membership attack (Nasr et al., 2019), in addition to the fact that the higher the number of classes in the data set, the higher the level of membership leakage will be (Truex et al., 2018).

### 5.2.2. Model poisoning attacks to extract training data

In model poisoning attacks (see Fig. 2), the attacker introduces adversarial examples into the training set $D_{train}$ in order to manipulate the behaviour of the model during inference time or to manipulate the training of the model.

Poisoning attacks are not restricted to training data points, but also to model weights. In a federated setting, for example, it is possible to poison the global model by influencing the weights or regularising them, inserting hidden back doors, or even injecting poisoning neurones (Muñoz-González et al., 2017; Jagielski et al., 2018; Chen et al., 2018). These threats can be used as a first-step strategy to improve the success rate of other privacy attacks, such as membership inference attacks (Tramèr et al., 2022) or data reconstruction attacks such as the attack of Sun et al. (2021) where a malicious client falsifies its local training mode by injecting malicious model parameters that were shared with the victim through the aggregator. These malicious parameters require more effort from the victim's local model training in order to counteract the defect, which exposes more details to the adversary.

### 5.2.3. Model inversion attacks

Although membership inference attacks can reveal the existence of certain data points in the training of the model, model inversion attacks (Fredrikson et al., 2015; He et al., 2019; Wu et al., 2016) go beyond that (see Fig. 2) by trying to create similar realistic samples of features that accurately describe each of the classes of the data set that were used to train the targeted model itself. Furthermore, in the case where the features match the raw data (image data, videos, etc.), the inversion attack includes a reconstruction of a full approximation of the training data used. For example, in the face recognition model, an attacker can recover an individual face image whose photo has been used as part of the training set.

These attacks use confidence vector values as probability vectors returned by machine learning models exposed as APIs, then those vectors are used to compute an average that represents a certain class (Yang et al., 2019c). The biggest risk exists when a class represents a single individual data point, which is the case with face recognition tasks.

### 5.2.4. Attribute inference attacks

In attribute inference attacks (also called "feature reconstruction attack") (Jayaraman and Evans, 2022), the adversary knows some attributes about given data and by attacking a model that was trained on these data, the attacker aims to extract other attributes about those data (*e.g.*, an attacker knows the name and age attributes and aims to infer the gender). This kind of attack targets particularly vertical federated learning settings, where the attacker is either the active party[18] or the passive party.

When the attacker is the active party, this kind of attack is called a feature inference attack (Weng et al., 2020), (Luo et al., 2021), (Jiang et al., 2022). Weng et al. (2020) approach proposed two attacks, and both approaches focus on investigating the potential data leakage caused by numerical computations. The first attack targets XGBoost by encoding magic numbers in the gradients and then trying to recover the features via a proposed reverse sum attack method, while the second attack targets logistic regression using a reverse multiplication method. While Weng et al. work gave strong indications about the possibilities of data leakage through the sharing of intermediate computation results, their attacks targeted simple models, namely XGBoost and logistic regression models, and cannot be extended to more complex models. Later Luo et al. (2021) overcame this limitation by proposing an equality solving attack for linear regression models, a path restriction attack for decision

[18] In Vertical Federated Learning (VFL) the active party is the party that holds the labels.

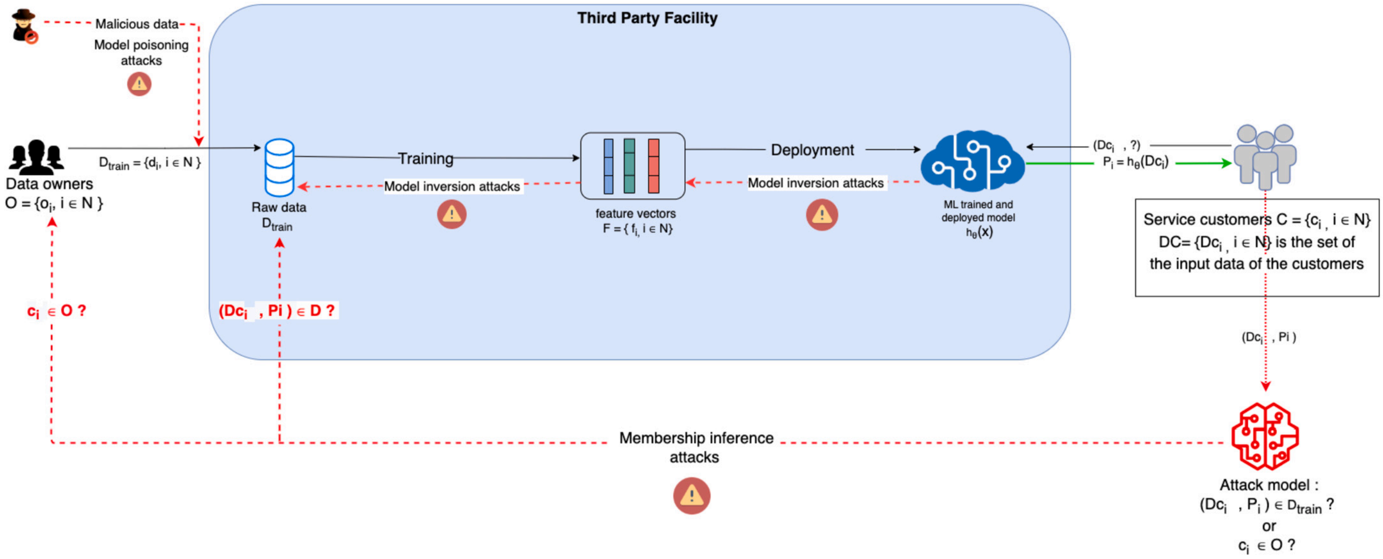

**Fig. 2.** Summary of some data privacy threats in a machine learning workflow, namely: model poisoning, model inversion, membership inference attacks. The scenario considered is when both data owners and customers use a third-party computation facility to perform model building operations, as well as inference.

tree models, and a generative regression network to attack more complex models. Although their work covered attacks for a wider range of models, their attack setting assumes that the attacker, that is, the active party, knows the entire model weights, including the local model of the passive party. In addition to attacks that exploit the intermediate computation results, Jiang et al. (2022) proposed another method from feature inference attacks that exploits gradients. Jiang's attack targeted simple models such as simple logistic regression models as well as more complex models like multilayer neural networks in both a white-box setting and black-box setting; however, they rely on the assumption that the active party has access to a small set of the passive party data.

On the other hand, when the attacker is the passive party, the attribute inference attack is called the “label inference attack” (Fu et al., 2022). Fu et al. (2022) proposed three different label inference attacks. In their first attack “passive label inference attack through model completion” the adversary fine-tunes the bottom model with an additional classification layer for label inference using a small set of auxiliary labelled data. The first attack exploits the ability of the passive attacker to turn his owned features into an indicative representation that can be used to predict the labels. For their second attack, the adversary accelerates the gradient descent on his local model using an adaptive malicious local optimiser to get access to a trained bottom model that encodes more hidden information about labels. The third attack, called the “direct label inference attack”, relies on the gradients received to infer the labels; however, the attack works only in a federated learning setting without model splitting.[19] Liu et al. (2021b) proposed another way to recover labels using batch-averaged gradients that target classification models, where the top model uses a Softmax function on the sum of intermediate results and cross entropy as the loss function. While Li et al. (2021) proposed two attacks to retrieve labels from the norm and the direction of intermediate shared gradients.

### 5.2.5. *Data reconstruction attacks*

Data reconstruction attacks or membership reconstruction attacks aim to reconstruct samples from the training set of the target model which results in severe consequences especially when the model is trained on highly regulated data such as health data.

This kind of attacks targets models trained in federated learning settings (both horizontal and vertical federated learning), as well as online learning settings.

During a horizontal federated learning setting,[20] reconstruction attacks exploit the gradients shared as intermediate results between the central aggregator and local clients (Zhu et al., 2019; Yin et al., 2021; Hitaj et al., 2017; Yang et al., 2023). Among the first works on these attacks, we find the attack introduced by Phong et al. (2018), where the authors demonstrated that training data can be mathematically derived from the weight gradients of the first layer and bias in fully connected models. Zhu et al. (2019) proposed a reconstruction attack on Convolutional Neural Networks (CNNs) where they initially created dummy data points sampled randomly with their respective random dummy labels; then those dummy data are fed to the models to get dummy gradients, by obtaining the dummy gradients, they try to optimise them to match the original gradients of a targeted original training data such that the closer the dummy gradients become to the original gradients, the more the dummy data will look like the original training data. Their experimental setup recovered both images and text data; however, it suffers from stability problems. Later, Zhao et al. (2020) addressed the stability limitations of Zhu's reconstruction attack (Zhu et al., 2019) by exploiting the relationship between the labels and the gradient signs. Another approach to address the problems is in the work of Zhu et al. (2019) was introduced by Ren et al. (2022) where a Generative Adversarial Neural Network (GAN) was used as an attack model, their approach was more stable and scales well on large-resolution images.

The extension of these attacks into a vertical federated learning setting[21] was first introduced by Jin et al. (2021) whose approach focusses on recovering the gradients with respect to the outputs of the first fully connected layer; then, using chain rule, they recover the inputs of the respective first fully connected layer; after that, they randomly gener-

[19] In vertical federated learning constructed with model splitting the participants can not access to the last layer of the neural network and thus the label in the server are more secure (Wang et al., 2023).

[20] “Horizontal federated learning, or sample-based federated learning, is introduced in the scenarios that data sets share the same feature space but different in sample” (Yang et al., 2019b).

[21] Vertical federated learning refers to the setting where “different parties hold different feature data belonging to the same set of samples “ (Li et al., 2023).

ate fake data and labels and try to optimise this generation process so that the inputs of the first fully connected layer of the true training data match the inputs of the first fully connected layer of the fake data. Data reconstruction attacks in vertical federated learning achieve high success rates because of the capability of Deep Neural Networks (DNNs) in modelling the correlation between the intermediate calculations and the inputs.

Data reconstruction attacks can also target online learning settings where models are curated by training on newly collected data (Salem et al., 2020), they exploit the posterior difference between the curated model and the old one.

#### 5.2.6. Training data extraction attacks from language models

Large generative language models are vulnerable to a training data leaking attack, where the generated outputs of the model can reveal text sequences from the model's training data. Carlini et al. (2021) designed a simple attack to study the extent of training data memorisation in GPT2. The attack relies on querying the model to generate large amounts of data; then they use membership inference attack to classify the generated outputs into either members or non-members of the training data. The attack showed that the larger the language models are, the more they memorise verbatim sequences of the training data. This kind of attacks feeds on the potential memorisation of sequences of training data by large neural networks. Carlini et al. (2019) showed that large sequence models tend to memorise rare and unique sequences even if those are not directly related to the intended task, regardless of whether the model is overfitted. They called this kind of memorisation "unintended memorisation", which occurs when models reveal out-of-distribution training data that are irrelevant to the learning task and do not help in improving model accuracy. Zhang et al. (2021) studied another type of memorisation called "Counterfactual Memorisation", which inspired notions from human psychology and studied the changes in model's predictions when a particular document is omitted during training. Their work revealed that more training epochs and the presence of foreign language text and structured text like car sales listings increase counterfactual memorisation. Furthermore, their study shows that the model predictions can be affected when particular training examples with high memorisation are omitted.

Moreover, the risk of memorisation by language models can be increased when there are duplicate sequences within the training data (Carlini et al., 2023).

## 6. Mitigation techniques

### 6.1. Trust model

Each and every privacy enhancing technology within the ML data flow is characterised by the guarantees it gives under a set of assumptions. In other words, confidentiality or privacy guarantees for each PET are granted under certain constraints. These constraints are known as *trust assumptions*. The trust assumptions bound the capabilities of the malicious entities involved, thus the guarantee holds only within those *limitations*: they do not hold anymore if the limitations are removed. Because of this duality, in all the tables where we comment on privacy-enhancing techniques' guarantees, we keep both columns "Trust Assumptions" and "Limitations". Some of our readers, usually security analysts, may prefer the trust assumptions to be stated explicitly; others, usually data owners, may find it more helpful to reflect on the limitations of a privacy-enhancing technique.

In this section, we present the existing privacy preserving techniques throughout the machine learning pipeline from data preparation to model inference along with their trust assumptions and the guarantees they offer. We also discuss their limitations and the costs that result from adopting them.

### 6.2. Mitigation techniques during the data preparation phase

Data prepossessing/preparation phase is one of the first steps in the machine learning pipeline where data are generally cleaned, labelled if necessary, and prepared to be fed to models for training.

When adopting a privacy-preserving technology in this phase, the training data owner aims to minimise the exposure of the data to the other threat actors either by concealing the sensitive attributes or by adding noise to the data before sending them to the training computation party.

Privacy-preserving approaches for data preparation focus on three directions: (i) Identifying sensitive attributes and concealing them partially or fully; (ii) Applying perturbation techniques that add a certain amount of noise to prevent reverse engineering statistical conclusions to retrieve original data points; (iii) Resorting to surrogate dataset techniques. All these techniques are summarised in Table 2 where the different privacy and confidentiality guarantees offered by these PETs are discussed together with the set of trust assumptions to consider. We also point out the utility loss that can occur due to adopting these techniques.

#### 6.2.1. Private attributes concealing and/or elimination

Sensitive attributes concealing mechanisms such as k-anonymity (Sweeney, 2002), l-diversity (Machanavajjhala et al., 2006), and t-closeness (Li et al., 2007) have been adopted to remove and/or replace any identification information, i.e. any attributes that are considered sensitive from the data used to train models. These approaches aim to ensure that the subjects of the data[22] or the entities whose data are considered private or sensitive cannot be re-identified based on their sensitive attributes while the data remain useful.

These mechanisms have been used for a long time in data mining workflows. Friedman et al. (2008) were among the first to propose an extension of k-anonymity to be applied in various data mining algorithms such as decision trees, association rules and clustering algorithms.

The k-anonymity mechanism ensures that private attributes about an individual are indistinguishable from at least k-1 other individuals. k-anonymity starts by spotting the identifiers and quasi-identifiers for each data attribute, after which the identifiers are removed, and the quasi-identifiers are partially obscured. Extending the k-anonymisation, the l-diversity mechanism reduces the granularity of the data representation; and by using suppression and generalisation techniques, a data point can be mapped to at least $l-1$ other records in the dataset, ensuring additional levels of diversity across sensitive fields. The t-closeness technique improves the t-diversity concept.

The main critique that anonymisation techniques often receive is the impact of anonymisation on the accuracy of models due to information loss caused by removing the identifiers and the quasi-identifiers from the data (Ni et al., 2022). To overcome this drawback, Kifer and Gehrke (2006) designed a utility metric for several anonymisation algorithms to produce an anonymous version of ML workflows.

Recent works focus on building utility-preserving anonymisation strategies. Yao et al. (2023) proposed a utility-aware anonymisation model for data that contain multiple sensitive attributes by deassociating the relationship between quasi-identifiers and sensitive attributes by partitioning a set of data records into groups. Alternatively, Goldsteen et al. (2022) builds accuracy-guided anonymisation. Their approach is based on training an anonymiser model on the training data used to build the target model whose accuracy is to be preserved; then they use the predictions of this target model as labels for the anonymiser.

Although not considered an anonymisation technique, we should also mention the various, and still commonly applied in certain do-

[22] We use the term 'subject of the data' or 'data subject' in the case where the data is personal.

**Table 2**
Privacy enhancing technologies applied during the data preparation phase.

| | Trust assumptions | Privacy guarantees | Confidentiality guarantees | Utility loss | Limitations |
|---|---|---|---|---|---|
| Anonymization | Threat actors do not have access to the anonymisation mechanism. | Limited privacy guarantees prone to re-identification attacks (Wondracek et al., 2010). | Confidentiality guarantees limited to anonymised features. | Depends on the relevance of the correlations between the anonymised features and the target variables. | Possibility to deanonymise the data via linkage attacks using other datasets. |
| Pseudo-anonymisation | Threat actors do not have access to the lookup tables that contain pseudonyms. | Limited privacy guarantees prone to re-identification attacks. | Confidentiality guarantees limited to pseudoanonymised features. | Depends on the relevance of the correlations between the pseudonymised features and the target variables. | Possibility to deanonymise the data when performing linkage attacks using other datasets. |
| Differential privacy | Training parties are not trusted to apply calibrated noise to produce differential privacy. | Strong privacy guarantees: The contribution of each individual data point cannot be inferred. | No confidentiality guarantees. | Depends on the privacy budget $\epsilon$ the smaller is $\epsilon$ the less accurate is the result, generally LDP is known to suffer from a high utility loss. | The more an $\epsilon$-DP algorithm is run, the weaker the privacy guarantee for a database becomes. |
| Surrogate datasets | Threat actors do not have access to the synthetic data generators | Privacy guarantees depend on how much can the synthetic data reveal about the real data. | Strong confidentiality guarantees. | Depends on how much the distributional characteristics of the original datasets are preserved in the synthetic dataset. | When generative models are used, there is an inherent risk that the surrogate data set may contain outlier data reproduced by the neural network generator. |

mains, techniques used to pseudo-anonymise the data. Generally speaking, their purpose is to replace the true entries with synthetically generated data (Neubauer and Heurix, 2011) and to maintain the lookup tables to keep them safe, to deanonymise the data in case of need. This may happen when, in the case of unexpected finding *e.g.*, finding out that a group has cancer or that in the analysis of message, one finds out signs of abuse, the true entities must be revealed.

Despite the simplicity, pseudo-anonymization remains weakly adopted in machine learning systems due to its many shortcomings. Pseudo-anonymisation poses technical challenges about safeguarding the lookup tables.

Furthermore, sensitive attributes concealing techniques are susceptible to deanonymisation attacks (Narayanan and Shmatikov, 2019; De Montjoye et al., 2013). If we take the example of natural language processing tasks, sensitive attributes concealing techniques start by identifying sensitive information in the text such as names, gender, or address using a named entity recogniser. However, the best named entity recogniser has an accuracy of 90% and here we end up with a 10% error rate. The tricky fact is that we evaluate the error based on a human annotation with no guarantees that an inter-annotator agreement was done.[23] Another source of noise is the fact that the named entity recogniser (NER) is heavily dependent on the structure of the text. Grammatical structure and typos play a significant role in the accuracy of NERs. In short, private attribute concealing techniques, despite the simplicity of implementing them, remain weak in the fort of the reidentification attacks.

### 6.2.2. Perturbation techniques

Perturbation techniques are statistical tools that add a certain amount of noise to prevent reverse engineering of statistical conclusions to infer the membership of individual data points. Local differential privacy (Dwork and Roth, 2014; Evfimievski et al., 2003) (LDP) has strong privacy guarantees because it maintains plausible deniability at the user level. Works such as (Chen et al., 2011; Jiang et al., 2013; Erlingsson et al., 2014) used LDP as a strong privacy guarantee for sharing sensitive location data and user data collection. One potential drawback when working with local differential privacy is the aggregated noise budget. To reduce the effect of the aggregated noise on the final result, LDP requires a large number of data points. Furthermore, since for each data point an amount of noise is added, the total noise tends to be large. An alternative approach is using global differential privacy by making the pre-processing algorithms differentially private. Mo et al. (2019) propose a differential private based preprocessing technique for distance-based clustering in big data mining; their proposed adaptive mechanism proves to provide a good trade-off between privacy and utility in distance-based clustering algorithms. Amin et al. (2019) proposed an $\epsilon$ differentially private algorithm to calculate the covariance matrix, which is a specific preprocessed representation of a dataset that can be used for regression and PCA. However, most differential private preprocessing techniques are generally designed for specific targeted algorithms, *e.g.*, the approach of Mo et al. (2019) is designed for distance-based clustering algorithms, while the technique of Amin et al. (2019) is designed for algorithms that compute on covariance matrices, and thus both approaches are hard to extend for other algorithms. To alleviate this limitation Stoddard et al. (2014) proposed a differentially private preprocessing algorithm for feature selection that is agnostic to the classifier used during the training phase.

### 6.2.3. Surrogate dataset techniques

Machine learning models aim to learn the underlying distributions of data sets to achieve the target. In the case of highly sensitive data, the usage of these data to train a model becomes problematic due to the very strict regulations that govern the processing and use of such datasets. Therefore, one way to achieve the goal is the usage of surrogate datasets, which are formed by grouping anonymised datasets and abstracting the dataset using sketching techniques (Sabay et al., 2018; Yang et al., 2019a) or even using generative models to generate synthetic datasets (Nik Aznan et al., 2019) (Assefa et al., 2021). Recent advances focus on the use of deep generative models to generate synthetic datasets; Dash et al. (2020) used GANs (generative adversarial networks) to generate hospital time series based on the MIMIC-III dataset. One of the shortcomings of generative models used to generate sensitive data is the fact that these models require sufficiently large and heterogeneous training data to build good quality generators; however, such a requirement is problematic to satisfy since access to sensitive real data, such as brain MRI data (Alrashedy et al., 2022), is restricted in the first place. To overcome this challenge, researchers used data

[23] Inter-annotator agreement IAA is a measure of the quality of the annotations done by multiple annotators by accounting for how many annotators can make the same annotation decision for a certain label category or class.

augmentation techniques such as the work of Huo et al. (2018) and Qasim et al. (2020) however, their approaches tend to limit the user's control over the generated data. A more recent work by Fernandez et al. (2022) proposes a model that generates both the MRI brain images and their respective labels where the generation of the MRI images is conditioned by the generated labels. Generating good-quality synthetic data faces many challenges. Generators should implement a statistically accurate generation model that introduces little to no bias compared to real data (Mannino and Abouzied, 2019). Furthermore, the process of generating sensitive data should be audited to allow an acceptable level of privacy, robustness and quality (Belgodere et al., 2023) (Alaa et al., 2022).

### 6.3. Mitigation techniques during the model building phase

Privacy enhancing techniques added during model training can be divided into four categories: (i) perturbation techniques added to the training algorithm itself to turn it into a differentially private algorithm; (ii) training on encrypted data; (iii) privacy-preserving architectural choices like PATE and federated learning; and (iv) training on privacy-preserving hardware solutions such as trusted execution environments. While the former serve to offer privacy guarantees by mitigating model reverse engineering attacks such as membership inference attacks, the latter offers confidentiality guarantees. This is due to the fact that cryptographic tools, TEEs, and vanilla federated learning guarantee the secrecy of the data and thus prevent attacks that try to directly access the data in its original form, while DP and PATE guarantee that adversaries cannot reveal sensitive information about the data through seeing statistics and/or knowledge extracted from the data. These techniques are summarised in Table 3 along with the inherited privacy and confidentiality guarantees that result from adopting them under a set of trust assumptions. In addition to that, we point out the potential utility losses caused by these PETs.

#### 6.3.1. Differential private training

Differential privacy can be implemented within the training algorithm to make optimisation algorithms differentially private, as illustrated in (Abadi et al., 2016; Li et al., 2019). Abadi et al. (2016) propose a differentially private stochastic gradient descent approach to train a privacy-preserving deep neural network model. Among more recent work, McMahan et al. (2017b) demonstrate that it is possible to train large recurrent language models with user-level differential privacy guarantees with only a negligible cost in predictive accuracy. In (Li et al., 2019), Li et al. proposed a differentially private algorithm for gradient-based parameter transfer to enable a differentially private setting for transfer learning tasks.

In differentially private stochastic gradient descent proposed by Abadi et al. (2016) and also (Shokri and Shmatikov, 2015) privacy comes at the cost of utility. Furthermore, the amount of noise added to gradients does not take into account the importance of the learnt patterns, which exposes fairness issues towards minorities in the dataset and also poor accuracy. To address this problem, Phan et al. (2017) proposed an adaptive mechanism to inject noise into features based on the contribution of each feature to the output, adding Laplace noise to the affine transformations of neurones and loss functions. Nasr et al. (2020) suggested that the loss in model accuracy is due to the fact that the Gaussian mechanism is not utility-preserving. They suggested randomising gradients with a t-student distribution instead.

Later works also aimed to design differentially private version of the non-gradient-based optimisers. In (Kusner et al., 2015) proposed a differentially private Bayesian optimisation to fine-tune the hyperparameters of a wide variety of machine learning models.

#### 6.3.2. Encrypted machine learning training

Cryptographic tools offer a strong confidentiality guarantee, which is also known in the literature as "confidential-level privacy", the adoption of cryptosystems in the training process is a promising step. However, the computation involved in model training is more complex. Traditional cryptosystems, such as the Advanced Encryption Standard (AES), preserve the secrecy of data during transmission or storage. However, when trying to compute on these ciphers, the results of the computations become meaningless; this yields the obligation to decrypt the ciphers prior to any computation.

Crypto-based training uses recently proposed advanced cryptographic schemes that mainly include homomorphic encryption (HE) (Gentry, 2009; van Dijk et al., 2010; Brakerski et al., 2014; Martins et al., 2017; Cheon et al., 2017) and functional encryption (FE) (Boneh et al., 2011; Goldwasser et al., 2014; Abdalla et al., 2015, 2019) schemes. These schemes offer the possibility of computing over encrypted data.

Homomorphic encryption (HE) is a form of public key encryption with the particularity that it allows computations on encrypted data due to the fact that it is build upon a homomorphism that by definition preserves the structure of each algebraic group. In short, the result of the computation remains encrypted and is the encrypted version of the result of the same computation performed on the original data. Likewise, functional encryption(FE) is a generalisation of public-key encryption that allows the evaluation of a function on a ciphertext and outputs the result in a plain-text form.

In both techniques, the computation party operates over encrypted inputs. However, the key difference between HE and FE is that in functional encryption, the computation party that evaluates a function $f(Enc_k(x))$ can learn the value of $f(x)$, while in homomorphic encryption, the computation party learns $Enc_k(f(x))$. Thus, theoretically speaking, if functional encryption is used, then the computation party is not trusted with the input data but is trusted enough with the outputs of the computation.

Compared to training on nonencrypted data, training on encrypted data may require an additional step: data encoding. This is because most cryptosystems, such as BGV (Brakerski, Gentry, Vaikuntanathan) (Yagisawa, 2015) compute on polynomials. Moreover, training machine learning algorithms require computations in floating-point. A later work in homomorphic encryption proposed a scheme called CKKS (Cheon, Kim, Kim, Song) (Cheon et al., 2017) that can operate on floating-point values.

Most works on training on encrypted data face many shortcomings directly related to the computation burden and also the loss of precision. Taking the case of homomorphic encryption, for example, when training a deep neural network model, a large chain of multiplications and non-linear function evaluations is performed. The latter one is not supported by homomorphic encryption schemes, so the solution is to either use lookup tables, polynomial substitution, or even approximations using low degree polynomials (Obla et al., 2020), while the former introduces the problems of exploding noise budgets, which can partially be addressed using bootstrapping techniques (Cheon et al., 2018).

For homomorphically encrypted training, we recall the works of Nikolaenko et al. (2013) to train a logistic regression model on a homomorphically encrypted dataset and the recent work of (Nandakumar et al., 2019) to train a handwritten digit classifier on the encrypted version of the MNIST dataset (Deng, 2012). More recently, Park et al. (2022) proposed a framework for training fair Support Vector Machine (SVM) algorithms using the CKKS scheme; their training approach includes adding a regularisation parameter that controls the magnitude of the disparate impact. To address the problem of computing nonlinear functions, Baruch et al. (2022) proposed a novel framework for training neural networks by replacing activation functions with trainable polynomial functions where the coefficients are learnt during the training process, they also used knowledge distillation to transfer knowledge from a stronger pre-trained teacher model to a weaker student model. Knowledge distillation is used to avoid a long training process. Their work shows promising results, especially that it allows for training large models. Yoo and Yoon (2021) also addressed the computation of nonlinear functions using the TFHE scheme, where they explained how

**Table 3**
Privacy enhancing technologies applied during the model training phase.

| | Trust Assumptions | Privacy guarantees | Confidentiality guarantees | Utility loss | Limitations |
|---|---|---|---|---|---|
| Differential private training | Training parties are trusted to apply calibrated noise to produce differential privacy. | strong privacy guarantees: The contribution of each individual data point cannot be inferred. | No confidentiality guarantees. | Depends on the privacy budget $\epsilon$ the smaller is $\epsilon$ the less accurate is the result, however, GDP is more utility preserving than LDP | The more an $\epsilon$-DP algorithm is run, the weaker the privacy guarantee for a database becomes. |
| Federated learning | The aggregator is trusted to be honest in performing the aggregation operation. | Limited to None privacy guarantees when used in vanilla mode, since shared gradients can leak additional information. | Strong confidentiality guarantees. | Depends on the federated learning architecture, the quality of the local datasets and the strategy of the local updates (Charles and Konečnỳ, 2021; Kang et al., 2022; Zhang et al., 2022). | The other data owners may perform poising attacks.<br>The training facility may carry out privacy attacks using shared gradients.<br>Model customers can perform privacy attacks by using query-based access to the models. |
| Homomorphic encryption For training data | Training parties do not have access to decryption keys. | No privacy guarantees. | Strong confidentiality guarantees. | The approximations of the non-linear functions can cause a utility/ accuracy loss.<br>The usage of approximate number schemes such as CKKS results in additional noise $Dec(ct) = \tilde{m} = m + f$, where f is a small error (Kim et al., 2022) | Model customers can perform privacy attacks using query-based access to the models. |
| Functional encryption for training data | Training parties do not have access to decryption keys. | No privacy guarantees. | Strong confidentiality guarantees. | Depending on the underlying used scheme generally, the existing schemes do not support comparison operations such as min or max, and FE-based works have shown their demonstration only up to 5 layers (Xu et al., 2019b) neural network. | The training party can perform privacy attacks since the gradients will be in plaintext.<br>Model customers can perform privacy attacks by using query-based access to the models. |
| Trusted execution environment | | No privacy guarantees. | Moderate confidentiality guarantees based on the hardware security. | No utility loss caused by training on TEEs since the training operations and the data will not be modified. | Prone to side-channel attacks.<br>Model customers can perform privacy attacks by using query-based access to the models. |
| PATE | Teacher models are trusted to be trained using disjoint subsets from the private dataset. | Strong privacy guarantees. | No confidentiality guarantees. | Depends on the privacy budget $\epsilon$ the smaller is $\epsilon$ the less accurate is the result. | The students must share their data with all the teachers, and therefore, no privacy is guaranteed in this step. |

to define the four main operations needed in the encrypted domain, namely: addition, two's complement, exponential function, and division to allow building the sigmoid function using primitive gates. Although their approach proved good accuracy, it suffers from low time performance.

The efficiency of computations is another issue that concerns training on homomorphically encrypted data (Lee et al., 2022). Mihara et al. (2020) improved the efficiency of training ML models on encrypted data using the CKKS scheme by proposing a new packing method for the weight matrix that allows significantly reducing the total number of heavy homomorphic operations. Other works such as the work of HELayers (Aharoni et al., 2020) proposed a new way of computing 2D convolutions in addition to a new packing method to reduce computation overhead.

The computation of non-linear functions in the encrypted domain is not the only challenge facing the training of ML models using homomorphic encryption; selecting the scheme parameters is based on determining the multiplicative depth prior to computations.[24] However, when training models, it is hard to determine the number of epochs needed to achieve the desired performance.

In functional encryption training, we recall the work of Ryffel et al. (2019) for a partially encrypted training process in a polynomial neural network model, and the work of Xu et al. (2019b) who proposed a framework that supports training a neural network model over encrypted data.

### 6.3.3. *Federated learning*

Federated learning (FL) (McMahan et al., 2017a) is rather an architectural solution for a privacy-preserving training where multiple data owners can contribute in training a large model without the need for their datasets to be moved to a third-party computation facility. In this decentralised setting the computation party, also called the model aggregator, sends copies of the model to the datasets owners to run a local training for a number of epochs on their data and then sends back the locally trained version to a model aggregator that computes the model parameters' updates and sends the new version to the participants again in an iterative process. The decentralised training opens many possibilities in real-world applications in which regulations and data sharing policies do not allow the transfer of such sensitive data. Despite the fact that federated learning offers confidentiality level of guarantees for the training data, Thakkar et al. (2021) found that FL settings have an effect

[24] "The multiplicative depth is the maximal number of sequential homomorphic multiplications which can be performed on fresh ciphertexts such that once decrypted we retrieve the result of these multiplications." (Aubry et al., 2020).

in reducing unintended memorisation, which is a key factor in training data extraction attacks explored in section 5.2.6.

Although the participating parties do not share their training datasets directly, the local models are shared with the aggregator; which exposes the risks of information leakage and disclosure of privacy. Many proposals have been made to mitigate these risks, such as privacy-preserving meta-learning (Xu et al., 2019a) and client-sided (local) differential privacy (Geyer et al., 2017). By design, federated learning has a heavy communication burden and therefore multiple works focus on alleviating this issue (Diao et al., 2020; Ye et al., 2020).

### 6.3.4. Training in trusted execution environments

The trusted execution environment (TEE) creates an isolated execution environment on a separate kernel that provides guarantees on code authentication, the integrity of the runtime state, and the confidentiality of its code, data, and runtime states kept in permanent memory. Thus, in can be used to process data on an untrusted third-party computation facility.

There exists work that uses these environments especially for global aggregators in a decentralised training setting (Mo et al., 2021; Chen et al., 2020).

Trusted environments are prone to side channel attacks that infer leaked informations obtained from exploiting the hardware implementation leakages such as: power consumption, electromagnetic leaks, or even sound that can provide an additional information about the processes. To mitigate these risks, in general, TEEs are employed with additional oblivious techniques (Stefanov et al., 2018). In a recent work (Law et al., 2020), the authors proposed a collaborative secure XGBoost system that runs in trusted environments by additionally modifying XGBoost's algorithms to be data-agnostic to avoid potential side-channel risks. A similar work also addressed this issue by partially sampling the order when the active party or all parties have access to the TEE (Chamani and Papadopoulos, 2020).

### 6.3.5. PATE

Private aggregation of teacher ensembles (PATE) (Papernot et al., 2018b) is an architecture that transfers the knowledge of an ensemble model called "teachers" to a "student" model to offer a model inference service for a student model. Teacher models are trained using disjoint subsets with no overlaps from the private dataset with no constraints on teacher training. Then, the knowledge learnt by the teachers is transferred to a public student model; the teachers label a public unlabelled dataset which will be used to train the student. The privacy of sensitive data that teachers have been trained on is guaranteed by adding a DP noise during the labelling process.

PATE guarantees privacy by limiting student training to a limited number of teacher votes and revealing only the topmost vote after carefully adding random noise.

Several extensions have been introduced to PATE. Jordon et al. (2018) proposed PATE-GAN, a modified version of GANs by modifying the discriminator training procedure using PATE. Their proposed method is useful to generate synthetic data that satisfy differential privacy guarantees with respect to the original data. The key limitation of PATE-GAN is that it relies on the assumption of the necessity of training the discriminator with PATE to ensure that the generator will satisfy DP guarantees; however, when the synthetic records are labelled as fake by the teacher discriminators, the student discriminator would be trained on a biased dataset and the generator will generate low quality synthetic data. G-PATE (Long et al., 2021) addresses this limitation by proving that it is not necessary to ensure that the discriminator is differentially private to train a differentially private generator. Teacher discriminators are directly connected to the student generator, and the student model does not have its own discriminator. To ensure DP the gradient aggregator adds noise to the information from teacher discriminators, and the output of this aggregator is a gradient vector that guides the student generator to boost the quality of its synthetic samples. The PATE framework was also used in speech classification with the work of Yang et al. who proposed PATE-AAE (Yang et al., 2021) an adversarial autoencoder generator with a PATE-based classifier (PATE-AAE).

PATE uses $\epsilon$ differential privacy, where $\epsilon$ is the privacy parameter that assigns a protection level to the entire data set. This protection may be unnecessary for unsensitive data points in the datasets, and thus the utility loss becomes unjustifiable for datasets where only a subset is considered sensitive. To solve this issue Boenisch et al. (2023) proposed the "Individualised PATE" framework for an individual assignment of privacy budgets among only the sensitive training data. Their approach proves to be more utility preserving.

## 6.4. Mitigation techniques during the model serving phase:

Model inference or model serving is the last step in the machine learning pipeline. At this stage, the model is trained and ready to be used as a service. Nowadays, machine learning as a service is guaranteed via an emerging set of cloud platforms which are able to deploy a wide set of models and provide predictions to customers via end points. Most PETs applied in this step aim to protect the confidentiality of the inference data during the inference task. This is due to the deployment design choices of the service, where the inference is fully outsourced, and thus implies that the customer needs to send his data in order to receive the predictions.

Privacy-enhancing strategies during model inference can be grouped into two main categories: (i) cryptographic tools, and (ii) privacy preserving hardware solutions such as trusted execution environments. These techniques are summarised in Table 4. The guarantees provided when adopting these tools are stated in a set of *trust assumptions*. We also discuss the various utility losses that may be caused when adopting these techniques.

### 6.4.1. Encrypted inference

From a computational point of view, the inference task is relatively simpler than the training task. This is because one inference operation can be seen as one epoch of training for one data sample without calculating the error and propagating the loss to update the parameters. Thus, given the same model, the inference is computationally simpler and less consuming than the training. For this reason we generally find that advanced cryptosystems (primarily homomorphic encryption) are more applied for inference than for training. This is due to the computational inefficiency of these cryptosystems.

Encrypted inference is a secured computation that primarily aims to enable two or more parties to arbitrarily evaluate a function for both their inputs without revealing anything except the output of the computation. Under this set of tools we find secure multi-party computation protocols (Cramer et al., 2015), Garbled circuit evaluation protocols (Yao, 1982), and advanced cryptosystems like homomorphic encryption (Gentry, 2009; van Dijk et al., 2010; Brakerski et al., 2014; Martins et al., 2017; Cheon et al., 2017) and functional encryption (Boneh et al., 2011; Goldwasser et al., 2014; Abdalla et al., 2015, 2019).

Secure multiparty computation (MPC) protocols are used when multiple parties want to privately evaluate a function over their inputs. On the other hand, homomorphic encryption and functional encryption are cryptographic primitives that allow the computation on encrypted data. Thus as part of the MPC strategy choices, homomorphic encryption can be used as part of the protocol (Ghanem and Moursy, 2019). Earlier works that introduced HE in machine learning inference include regression analysis models (Nikolaenko et al., 2013; de Cock et al., 2015). CryptoNets (Gilad-Bachrach et al., 2016) was among the first propositions to deploy neural networks on encrypted data using a levelled homomorphic encryption scheme. CryptoNets supports the addition and multiplication of encrypted data, but requires prior knowledge of the complexity of the arithmetic circuit. Multiple works emerged after CryptoNets suggesting enhancements, for *e.g.*, the Delphi framework (Mishra et al., 2020) that integrates a hybrid cryptographic protocol to reduce

**Table 4**
Privacy enhancing technologies applied during the model inference / service phase.

| | Trust Assumptions | Privacy guarantees | Confidentiality guarantees | Utility loss | Limitations |
|---|---|---|---|---|---|
| Trusted execution environment | | **No** privacy guarantees. | **moderate** confidentiality guarantees based on the hardware security. | No utility loss caused by performing inference on TEEs since this deployment choice won't change anything in the model operations and the inference data won't be modified. | Prone to side channel attacks.<br>Model customers can perform privacy attacks by exploiting the inference results. |
| Oblivious transformation for inference | The encryption keys are not accessed by the inference party. | **No** privacy guarantees. | **strong** confidentiality guarantees. | Depends on the cryptographic primitives used *e.g.*, (Liu et al., 2017; Huang et al., 2021). | Model customers can conduct privacy attacks using query-based access to the models. |
| Homomorphic encryption for inference data | The inference party does not have access to the decryption keys. | **No** privacy guarantees. | **Strong** confidentiality guarantees. | - The approximations of the non-linear functions can cause a utility/ accuracy loss.<br>- The usage of approximate number schemes such as CKKS result in an additional noise $Dec(ct) = \tilde{m} = m + f$, where f is a small error (Kim et al., 2022). | Model customers can perform privacy attacks using query-based access to the models. |
| Functional encryption for inference data | The inference party does not have access to the decryption keys | **No** privacy guarantees. | **Strong** confidentiality guarantees. | Depends on the underlying scheme generally used so far: The existing schemes do not support comparison operations such as: min or max and FE based works have shown their demonstration only up to 5 layers (Xu et al., 2019b) neural network. | The inference party can perform privacy attacks, since the predictions will be in plaintext.<br>Model customers can perform privacy attacks using query-based access to the models. |
| Confidence masking of model outputs | The prediction party performs the confidence masking correctly. | Privacy guarantees depend on the confidence masking technique. | **No** confidentiality guarantees. | Noise based masking techniques may not ensure the accuracy of the labels. | |

both the computation and communication costs, in addition, the authors also included a planner that generates neural network architecture configurations automatically. Helen (Zheng et al., 2019) is another work that developed a secure coopetitive learning of a linear model without disclosing their data during the process of a distributed convex optimisation technique called alternating direction method of multipliers (ADMM), in which a generic maliciously secure multiparty computation is based on the SPDZ protocol (Damgård et al., 2012) derived from Somewhat Homomorphic Encryption (SHE) schemes. In addition to that and recently, the authors of Helen proposed Cerebro (Zheng et al., 2021) to study the trade-off between transparency and privacy in collaborative cryptographic learning.

On the same track, functional encryption-based approaches such as (Ryffel et al., 2019; Marc et al., 2019; Dufour-Sans et al., 2018) are also emerging, the authors proposed methods to perform partially encrypted and privacy-preserving predictions using adversarial training and functional encryption.

Though FE schemes offer the possibility to compute on encrypted data, the result will be in plaintext. Thus, when choosing the proper technique to preserve the confidentiality of the data between homomorphic encryption or functional encryption, one should consider the sensitivity of the predictions and the trust level of the inference computation party.

Garbled circuits and secure multiparty computation protocols are not emerging techniques: they have been studied for a long time. However, the community continues to work to improve their efficiency (Katz et al., 2018; Mohassel et al., 2015). Garbled circuits are generally an important component in secure multiparty protocols. The breakthrough that introduced these techniques within the Privacy Preserving Machine Learning (PPML) community was the Chameleon (Riazi et al., 2018) framework, which combines the usage of additive secret techniques for linear evaluation and garbled circuit protocols for non-linear evaluation. Similarly to Chamelon, the $ABY^3$ framework (Mohassel and Rindal, 2018) proposes a three-server paradigm based on the mixed 2PC protocol wherein data owners secretly share their data among three servers that train and evaluate models on the joint data using three-party computation.

We also find the deepSecure framework (Rouhani et al., 2018), which proposes a more general framework that supports various types of neural network evaluations based on garbled circuits and efficient design and optimisation methodologies. The efficiency of secure multi-party computation protocols depends on the underlying cryptographic primitives used, however when evaluating ML models under secure multi-party computation protocols, the computations will be more time-consuming compared to plain text training or inference (Sayyad, 2020). Garbled circuits are also used to evaluate nonlinear functions in homomorphic encryption settings because the HE schemes do not support non-linear evaluation.

### 6.4.2. *Oblivious transformation*

It is a knowledge transformation that changes an existing model into an *oblivious model* (Liu et al., 2017; Rathee et al., 2021). They are generally used in combination with advanced encryption schemes and trusted environments. Oblivious transformations ensure that predictions are calculated so that the deployment server learns nothing about the input of the clients and the clients learn nothing about the model, except the prediction results. The first attempt to construct an oblivious representation of neural networks for secure inference was done by Barni et al. (2006) where Pallier's scheme (Paillier, 1999) was used in addition to some oblivious transformations. However, this method is considered inefficient. Also it may leak information about the input data. Among recent works in this direction, we recall MiniONN (Liu et al., 2017) in which the authors used lightweight cryptographic primitives such as secret sharing and garbled circuits to evaluate neural networks, making it less computationally expensive. However, since it used an approximation for computing nonlinear functions, the method has some shortcomings in model's utility and accuracy, especially for large models.

Lately, to address the scalability and efficiency problems in the previously mentioned works, Huang et al. propose TONIC (Huang et al., 2021) a compiler to convert traditionally trained neural networks into oblivious format using two of the two secure two-party computation languages, i.e., ObliVM and ABY to scale faster for real-world applications.

### 6.4.3. *Inference on trusted environments*

Trusted Execution Environments (TEE) can also be used for private inference, where the model customer can send the input data to a trusted environment for inference with less privacy concerns, since the inference will be executed in an isolated environment. Generally, TEEs are combined with oblivious transformations to mitigate the risks of side-channel attacks (Stefanov et al., 2018). To enhance the efficiency of such an approach, there are multiple frameworks to optimise both the privacy guarantees and the computation burden for machine learning models. For *e.g.*, Slalom (Tramer and Boneh, 2018), a framework for efficient deep learning inference in any trusted execution environment. The framework partitions execution between trusted and untrusted environments. In particular, all linear layers are securely delegated from a TEE (*e.g.*, Intel SGX, or Sanctum) to a faster, yet untrusted, co-located processor.

### 6.4.4. *Confidence masking of model outputs*

Confidence masking techniques aim to minimise privacy leakage in model outputs by obfuscating the confidence scores without changing the predicted label. Jia et al. (2019b) proposed MemGuard, a confidence masking strategy for model outputs to defend against membership inference attacks. MemGuard adds a carefully crafted noise level to the confidence vector of the targeted classifiers in order to fool a membership inference by misclassifying them. This is possible because a membership inference attack is a classifier that is able to classify confidence vectors into either members or non-members of the training data, and classifiers are vulnerable to adversarial examples. MemGuard feeds on this vulnerability because the added noise vector aims to turn the confidence vector into an adversarial one to fool the membership inference attack classifier. Later, Yang et al. (2020b) proposed a confidence vector purification framework. The framework is based on training a purifier model that takes confidence vector outputed by the target model and computes a purified version of this vector that reduces information content in the confidence vector by minimising the dispersion. Their approach can mitigate both membership inference and inversion attacks, but it cannot ensure that the labels will not be changed. Mazzone et al. (2022) developed an enhanced strategy that used a repeated knowledge distillation strategy before applying confidence masking. The success of this approach is based on the availability of an adequate surrogate dataset to perform the distillation process.

Other confidence masking approaches limit the exposure of the confidence vector to only the top k scores (Li and Zhang, 2021) or even the label-only score (Shokri et al., 2017). Despite their promising results, confidence masking techniques remain weak against Label-Only attacks (Choquette-Choo et al., 2021b).

## 7. Technical tools for privacy preserving machine learning

The adoption of PETs in production or even in research depends heavily on the available libraries and implementations. In this section, we will discuss some of the libraries that implement, namely: homomorphic encryption schemes, differential privacy, functional encryption, secure multi-party computation, garbled circuits, and oblivious transformations, some anonymisation libraries along with the hybrid toolkit HaGrid, which implements a wide set of PETs to allow remote privacy preserving data analysis. Table 5 summarises the tools and libraries for privacy-enhancing technologies discussed in this work along with their basic properties. We emphasise that the list is not an exhaustive one but rather should be a sufficient guide to practitioners.

From an EU GDPR legal perspective, anonymisation and pseudoanonymisation techniques provide solutions when the processing of certain types of data is highly regulated. ARX (Prasser et al., 2020) toolbox offers a wide set of algorithms along this goal that can be helpful, even for non-developers since it has a graphical user interface. The weak point of ARX is the absence of binders for Python since it is the most used language in the machine learning community or even among data analysts.

Encryption tools offer a strong level of confidentiality and are generally useful when the computational party is not trusted. In Table 5 we find a wide range of libraries that offer various implementations of homomorphic encryption schemes, functional encryption schemes, secure multi-party computation protocols, Garbled circuits, and Oblivious transformation toolkits. Supporting approximate floating-point operations in these implementation schemes is the most important feature in these libraries when used for machine learning tasks since the various mathematical operations in machine learning training or inference are executed on floating-point numbers; *e.g.*, for homomorphic encryption; the CKKS scheme (Cheon et al., 2017) is widely used in PPML; for this, we note that the library PALISADE (2020) is the only library that implements a countermeasure against secret key recovery attacks on CKKS (Li and Micciancio, 2021) by adding some Gaussian noise during decryption. The amount of noise that should be added during decryption is parameterized in PALISADE, leaving the choice to the user to decide the amount of noise to mitigate key recovery attacks. This design choice has many shortcomings since both the overestimation and the underestimation of the noise distribution parameters are harmful. An underestimation is useless from a privacy perspective, whereas the overestimation of noise can cause a huge utility loss, especially since CKKS is a lossy encryption scheme where the decryption outputs a noisy result. A more strict noise flooding mechanism must be implemented such as the one described by Li et al. (2022). HE libraries can also be subject to side channel attacks such as the one described by Aydin et al. (2022). The lack of Python binders and tensor-based implementation such as (Benaissa et al., 2021) of these schemes makes the adoption of cryptographic tools in machine learning restricted to data scientists with sufficient cryptographic knowledge.

Unlike cryptographic tools, differential privacy tools are more data-scientist-friendly with a wide set of toolkits such as IBM Diffprivlib (Holohan et al., 2019) and Google DP library (Google, 2021). The adoption of DP in the already existing ML frameworks like Tensorflow and Pytorch by creating alternatives of the existing ML frameworks such as Pytorch Opacus (Yousefpour et al., 2021) and Tensorflow Privacy (Tensorflow team, 2019) made differential privacy more widely adopted in PPML.

PPML tools aim to solve the lack of access and usage of data that are highly sensitive and private by providing alternative privacy-preserving tools that motivated the creation of HaGrid (Hall et al., 2021). The Hagrid toolkit provides a stack of secure and private data science software including a differential privacy library, secure multi-party computation support, and a federated learning setting in a Numpy-like interface in addition to a command line interface to deploy data nodes and to make them accessible to data scientists to perform analysis on data in a secure and private way without even seeing the data.

There exist other tools that can be considered under the umbrella of ML privacy preservation tools, such as TEE orchestration tools and hardware and frameworks of federated learning.

**Takeaway:** For a wider adoption, PPML tools should have a Python tensor-like support to make it data scientist friendly, and a floating point computation support (especially for cryptographic tools).

## 8. Challenges and research directions

Finally, we would like to point out some of the open questions and research directions in this field:

- **Measurement and evaluation of privacy:** there is a need for formal evaluation tools and frameworks to measure the privacy guarantees that privacy preserving machine learning (PPML) tools provide. Measurements are needed for auditing and data protection risk assessment for accountability purposes.
- **Communication Efficiency:** some PPML tools suffer from a high communication cost and therefore are inefficient to deploy for large complex machine learning models, such as secure multiparty (SPMC) protocols and federated learning. Thus, there is a large room for optimising computational complexity, such as optimising SMPC protocols through the use of MPC compilers.
- **Computation Efficiency:** computation efficiency is one of the drawbacks of the cryptographic tools used in PPML. This established the need to reduce the computational complexity either by making ML models designed for cryptographic evaluation or by making the cryptographic schemes more computationally efficient.
- **Privacy budget versus utility and/or fairness**: When adopting perturbation techniques such as differential privacy, privacy comes at the cost of utility (thus reliability), as well as the fairness of models (since minorities in datasets are the most harmed by utility loss). Approaches that balance between privacy, fairness, and utility are needed to establish trustworthy machine learning systems.
- **The relationship between privacy and other trustworthy machine learning elements:** trustworthiness in machine learning includes many components such as preservation of privacy, fairness, transparency, robustness, etc. The enforcement of privacy can come at the cost of these other components, which makes it challenging to identify the impact of privacy enhancing technologies on transparency, for example. Understanding how each tool may impact the other elements of trustworthiness and how to set a clear trade-off is an active area of research.

## 9. Conclusion

Data protection concerns are the main blocker to the adoption of machine learning in life-impacting applications where sensitive and personal data should be processed. We have collected, reviewed, and presented a body of knowledge about data protection (and its technical interpretation as data privacy and confidentiality) in machine learning systems.

By taking the perspective of the data owners being the threatened parties, we showed that the threats against the process data can be fully understood only in reference to the machine learning pipeline, the role of parties, and the deployment architecture for the machine learning system. Such a perspective, which is in line with the discussion on data protection and trustworthy AI according within current EU regulations, points out the different vulnerabilities in machine learning systems that multiply the risks of data leakage throughout the pipeline by taking into consideration the design choices, the different actors, and how the threat actors can exploit these weaknesses to maintain efficient yet hard to detect privacy attacks.

This work thoroughly discusses the defence mechanisms for each phase within the ML pipeline, so highlighting the guarantees that these tools offer under specific trust assumptions. Since the defence mechanisms have not reached the full maturity level yet, we focus on discussing problems in adopting these PETs in order to highlight possible rooms for improvement.

Finally, we outline some challenges and research directions that call for a collaborative interdisciplinary effort to address them. This work serves as a guide for ML practitioners and researchers in this field.

## 10. The next horizon: from privacy-preserving ML to trustworthy AI

In order to offer a larger context, we discuss our work in the broader context of the European initiative on trustworthy machine learning. We review a few key principles and concepts in comparison to the technical discussion presented in the preceding sections.

According to the report of the High-Level Expert Group on Artificial Intelligence in the European Commission (HLEG) (High-Level Expert Group on AI, 2019), a trustworthy machine learning model fulfils many criteria, namely: lawful and ethical use, robustness and reliability, fairness and transparency, and preservation of privacy.

The real challenge in implementing these requirements of trustworthiness lies in the continuous evolution, update, and retraining operations that touch the systems throughout their life cycle; as well as the non-deterministic nature of some models (e.g., usage of dropout, variational auto-encoders, etc.).

Hence, the traditional tools that are used to measure the certainty of software systems working correctly such as unit tests, end-user testing, code reviews, and design documentation; are insufficient to ensure the quality and trustworthiness of machine learning systems.

In the rest of this section, we will discuss the requirements to consider for trustworthy machine-learning pipelines.

### 10.1. Lawfulness and ethicalness

Receiving data inputs and functioning as services places machine learning applications within the scope of data privacy and data processing regulations, which encompass frameworks like the EU GDPR, UK GDPR, as well as more recent legislations such as the AI Act and the Data Governance Act. However, these regulations present limitations to machine learning use cases. If we take the example of the principle of data minimisation in the GDPR, it aims to ensure that the collected data fit the purpose of the project. This suggests that developers should determine the quantity of data and features required for each model, this is technically a challenge since it is not always possible to predict what a model will learn from data and how this knowledge will be shaped to get the predictions. Furthermore, limiting features from a training dataset can lead to a decrease in the model generalisation, and thus the overall performance of the model will be affected.

Data minimisation is not the only challenging principle; fairness and transparency have also been widely discussed (Felzmann et al., 2019). Another particularity of machine learning systems is the ability to repurpose them (technically known as “transfer learning”). This common strategy to reuse models allows the reduction of need to collect large volumes of data when building large models. The particularity of transfer learning is the possibility to reuse models without the reuse of the original data. Therefore, listing all possible uses for which the data will or may be used can be challenging.

Multiple efforts contributed to the implementation of ethical guidelines for machine learning and artificial intelligence systems, such as the EU Commission’s ethical guidelines for trustworthy AI (European Commission, 2019). These guidelines aim at prescribing how machine learning models should be built and exploited for the best benefit of the environment and the large public.

### 10.2. Reliability and robustness

The Reliability and Robustness of the machine learning tools are vital requirements to establish trust in machine learning products. They primarily deal with fault tolerance, the recoverability of the system, and the quality of the system’s output.

To elaborate, machine learning models are trained using static datasets; however, once deployed, these models may receive inference data whose distribution is different from the baseline distribution of the training data; hence resulting in low-quality results. The ability of a model to generalise depends on the model's hyper parameters and also on the variety and the size of the training datasets; and is far from being guaranteed (Chung et al., 2018; Barbedo, 2018).

In addition to that, a reliable machine learning system should be able to handle uncertainties caused by degrading equipment or sensors in manufacturing environments. The detection of uncertainties and drops in production performance are among the consequences of data set drifts that can be responsible for the inference bias (Müller and Salathé, 2020).

The robustness of machine learning models addresses the vulnerability to a set of attacks such as adversarial attacks (Michels et al., 2019), *e.g.*, Wu et al. (2020b) could design T-shirts with adversarial prints that can make individuals wearing them undetectable by pedestrian detection models, thus decreasing the reliability of security systems that use these models. Robustness can be challenged by neural-level trojans (Zou et al., 2018), hardware attacks (Bhunia et al., 2014; Clements and Lao, 2019), and privacy threats (Section 5.2).

To address these issues, continuous monitoring is required for the inference of the deployed model, the reliability of the hardware used for deployment, and the immunity against network cyberattacks.

### *10.3. Transparency*

Transparency promotes understandability; the term can be used interchangeably with explainability and interpretability. The concept emerged as a requirement in data protection laws. Both the UK and the EU GDPR set transparency and fairness as the first principles of data protection regulations (Fischer-Hübner et al., 2016).

As a technical property in machine learning models, transparency addresses the understandability of how the model functions internally: at the model level, the individual components (model parameters), and the training algorithm (Lepri et al., 2018).

Approaches to address transparency by using explainability tools focus on explaining the already used black box models; while the paradigm of transparency by interpretability mainly focusses on building easy-to-trace and less opaque models (Rudin, 2019). Building interpretable models requires a significant amount of expertise and computational resources. The task becomes more challenging when the interpretable models; which are by design simpler models, have to match the performance of the complex black-box models. Furthermore, existing explanations of black-box models do not provide details about what the model is doing. For example, saliency maps, which are considered to be explanatory for convolutional neural networks, only provide information about the image parts that are omitted by the network; however, this does not give any details about what the model does with the parts that it considers relevant for the task.

### *10.4. Fairness*

Fairness in machine learning refers to issues related to a discriminating behaviour of models towards certain groups, especially the minorities, in the data sample used for training and/or testing.

The formulation of fairness in quantitative fields (e.g.: maths, computer science, etc.) tends to be narrow and neglects the nuances and various conceptions of fairness (Mulligan et al., 2019). For example, approaching fairness by equalising the accuracy metrics across the population groups ends up inducing residual unfairness within fair models (Kallus and Zhou, 2018). In addition to that and specifically in the case of machine learning systems, placing constraints on fairness may come at the cost of accuracy since it restricts the learning algorithm (Menon and Williamson, 2018). The black-box nature of some machine learning models, such as neural networks, makes it challenging to ensure fairness. The discussion of this trade-off of fairness versus accuracy is still widely studied. In a recent work (Dutta et al., 2020), authors demonstrated that this trade-off can be alleviated when the used datasets are actively collected. In other words, gathering more features for the unprivileged groups. While this sounds promising, it enlarges the dataset horizontally, which introduces an extra computation and storage burden.

Multiple advances in this direction include mainly defining criteria that a machine learning system must meet in order to be considered fair. Among those works, we find the Microsoft fairness checklist (Madaio et al., 2020) that incorporates different stages of the AI system development and deployment life cycle.

### *10.5. Privacy*

The data processed during training or inference is subject to different leakage and privacy threats that are a result of either the internal functioning of the machine learning models, the deployment architectures, or both.

Preserving privacy in machine learning systems goes beyond access control to the appropriateness of data flows and was described in many frameworks, namely the contextual integrity framework (Nissim and Wood, 2018; Nissenbaum, 2004). The structured transparency framework proposed by Trask et al. (2020) highlighted the idea of considering the appropriateness of the flow of data throughout the machine learning pipeline by considering five components, namely input privacy (referred to as confidentiality in our work), output privacy, input verification, output verification, and flow governance.

When we consider the mathematical definitions of privacy in machine learning, Differential Privacy (DP) (Dwork and Roth, 2014) has been widely accepted in multiple domains because of its provable privacy guarantee. However, the DP approaches applied in machine learning are still computationally inefficient.

Privacy in the machine learning context should guarantee governance over the input data and the algorithms, integrity of the processing, and its results to offer mechanisms for transparent and transparently auditable technical implementations.

> **Takeaway:** A trustworthy ML system can be achieved by establishing the requirements of an acceptable trade-off between the technical properties, mainly: fairness, transparency, privacy, and reliability; in accordance with suitable ethical and legal frameworks.

## CRediT authorship contribution statement

**Soumia Zohra El Mestari:** Conceptualization, Data curation, Investigation, Methodology, Resources, Visualization, Writing – original draft, Writing – review & editing. **Gabriele Lenzini:** Funding acquisition, Project administration, Supervision, Validation, Writing – review & editing. **Huseyin Demirci:** Resources, Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

No data was used for the research described in the article.

## Acknowledgements

This work has been supported by the EU 956562, MSCA-ITN-2020 - Innovative Training Networks, "Legality Attentive Data Scientists" (LeADS) project.

## Appendix A. Summary of technical tools used in ML

**Table 5**
PETs libraries and toolkits.

| Technique | Library Name | Supported algorithms/ schemes or protocols | Programming languages |
|---|---|---|---|
| Differential privacy | Google's DP library (Google, 2021) | Laplace mechanism<br>Gaussian Mechanism<br>Various statistical aggregations | C++, GO, JAVA |
|  | Pytorch Opacus (Yousefpour et al., 2021) | Various optimisation of DP SGD | Python |
|  | RAPPOR (Fanti et al., 2015) | Local Differential Privacy Mechanisms | Python, R |
|  | OpenDP (OpenDP community, 2021) | Core for DP algorithms<br>sdk for DP for tabular and relational data | Python, Rust |
|  | IBM Diffprivlib (Holohan et al., 2019) | Geometric mechanism<br>Gaussian Mechanism<br>Uniform Mechanism<br>Laplace Mechanism<br>Bingham Mechanism<br>Exponential Mechanism<br>Vector Mechanism<br>Models for Classifications and Clustering<br>Tools for DP histograms and quantiles | Python |
|  | TensorFlow Privacy (Tensorflow team, 2019) | Gaussian mechanism<br>Skellam Mechanism<br>DP SGD Optimiser<br>DP estimators (Binary and Multi Label) | Python |
|  | PyDP (OpenMined Community, 2020) | Geometric mechanism<br>Laplace mechanism<br>Aggregate statistics algorithms | C++, Python |
| Homomorphic encryption | TFHE (Chillotti et al., 2016) | TFHE (Chillotti et al., 2020) | Rust |
|  | CuFHE (Cetin et al., 2018) | TFHE (Chillotti et al., 2020) on CUDA-enabled GPUs | C++ |
|  | HEAAN (Han et al., 2016) | CKKS (Cheon et al., 2017) | C++, Python |
|  | SEAL (SEAL, 2022) | BFV (Brakerski, 2012)<br>CKKs (Cheon et al., 2017)<br>BGV (Brakerski et al., 2014) | Python, C++ |
|  | HElib (Hunt et al., 2020) | BGV (Brakerski et al., 2014)<br>CKKS (Cheon et al., 2017) | C++ |
|  | PALISADE (PALISADE, 2020) | BGV (Brakerski et al., 2014)<br>BFV (Brakerski, 2012)<br>FHEW (Ducas and Micciancio, 2015)<br>CKKS (Cheon et al., 2017) | C++ |
| Functional encryption | CiFEr/ GoFE (Marc et al., 2018) | Abdalla et al. scheme (Abdalla et al., 2015)<br>Abdalla et al. scheme (Abdalla et al., 2018)<br>Agrawal et al. scheme (Agrawal et al., 2016)<br>DMCFE (Chotard et al., 2018)<br>Datta et al. scheme (Datta et al., 2018)<br>FAME (Agrawal and Chase, 2017)<br>KP-ABE (Goyal et al., 2006)<br>Michalevsky et al. scheme (Michalevsky and Joye, 2018)<br>Dufour et al. scheme (Dufour-Sans et al., 2018) | CiFEr in C<br>GoFE in Go |
| Secure Multi-Party Computation | Crypten toolkit (Knott et al., 2021) | Additive secret sharing (Damgård et al., 2012; Evans et al., 2018) | Python |
|  | ABY /ABY3 (Demmler et al., 2015) | SPDZ (Damgård et al., 2012) and various MPC protocols<br>Linear regression<br>Logistic regression<br>Database Inner, Left, and Full Joins<br>Database Union<br>Set Cardinality | C++ |
|  | MP-SPDZ (Keller, 2020) | TinyOT (Nielsen et al., 2012)<br>SPDZ (Damgård et al., 2012)<br>MASCOT (Keller et al., 2016)<br>SPDZ, Overdrive (Keller et al., 2018)<br>Yao grained circuits (Yao, 1982)<br>Generalised secret sharing (Benaloh and Leichter, 1988)<br>Shamir (Shamir, 1979)<br>other MPC protocols; | C + +<br>Python |
| Garbled circuits and oblivious transformations (OT) | EMP-toolkit (Wang et al., 2016) | IKNP OT extension (Ishai et al., 2003)<br>Ferret OT (Yang et al., 2020a)<br>Wang et al. protocol (Wang et al., 2017) | C++ |
|  | TinyGarble (Hussain et al., 2020) | Yao's Garbled Circuit (GC) (Yao, 1982) | C++ |
| Anonymization | ARX (Prasser et al., 2020) | k-anonymity (Sweeney, 2002)<br>t-closeness (Li et al., 2007)<br>$\delta$-disclosure privacy (Brickell and Shmatikov, 2008)<br>$\beta$-likeness (Cao and Karras, 2012)<br>$\delta$-presence (Nergiz et al., 2007) | JAVA |
|  | AnonyPy (Fujita, 2021) | k-anonymity (Sweeney, 2002)<br>l-diversity (Machanavajjhala et al., 2006)<br>t-closeness (Li et al., 2007) | Python |
| Hybrid toolkits | HaGrid/PySyft (Hall et al., 2021) | SMPC protocols<br>Differential privacy<br>Federated learning | Python |

## Appendix B. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cose.2023.103605.

## References

Abadi, Martin, Chu, Andy, Goodfellow, Ian, McMahan, Brendan, Mironov, Ilya, Talwar, Kunal, Zhang, Li, 2016. Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 308–318.

Abdalla, Michel, Benhamouda, Fabrice, Kohlweiss, Markulf, Waldner, Hendrik, 2019. Decentralizing inner-product functional encryption. In: IACR International Workshop on Public Key Cryptography. Springer, pp. 128–157.

Abdalla, Michel, Bourse, Florian, De Caro, Angelo, Pointcheval, David, 2015. Simple functional encryption schemes for inner products. In: Katz, Jonathan (Ed.), Public-Key Cryptography – PKC 2015. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 733–751.

Abdalla, Michel, Catalano, Dario, Fiore, Dario, Gay, Romain, Ursu, Bogdan, 2018. Multi-input functional encryption for inner products: function-hiding realizations and constructions without pairings. In: Shacham, Hovav, Boldyreva, Alexandra (Eds.), Advances in Cryptology – CRYPTO 2018. Springer International Publishing, pp. 597–627.

Agrawal, Shashank, Chase, Melissa, 2017. Fame: fast attribute-based message encryption. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. CCS '17. Association for Computing Machinery, New York, NY, USA, pp. 665–682.

Agrawal, Shweta, Libert, Benoît, Stehlé, Damien, 2016. Fully secure functional encryption for inner products, from standard assumptions. In: Robshaw, Matthew, Katz, Jonathan (Eds.), Advances in Cryptology – CRYPTO 2016. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 333–362.

Aharoni, Ehud, Adir, Allon, Baruch, Moran, Drucker, Nir, Ezov, Gilad, Farkash, Ariel, Greenberg, Lev, Masalha, Ramy, Moshkowich, Guy, Murik, Dov, et al., 2020. Helayers: a tile tensors framework for large neural networks on encrypted data. arXiv preprint arXiv:2011.01805.

Al-Rubaie, Mohammad, Chang, J. Morris, 2019. Privacy-preserving machine learning: threats and solutions. IEEE Secur. Priv. 17 (2), 49–58.

Alaa, Ahmed, Van Breugel, Boris, Saveliev, Evgeny S., van der Schaar, Mihaela, 2022. How faithful is your synthetic data? Sample-level metrics for evaluating and auditing generative models. In: International Conference on Machine Learning. PMLR, pp. 290–306.

Md Ali, Nawab Yousuf, Md Rahman, Lizur, Chaki, Jyotismita, Dey, Nilanjan, Santosh, K.C., et al., 2021. Machine translation using deep learning for universal networking language based on their structure. Int. J. Mach. Learn. Cybern. 12 (8), 2365–2376.

Alrashedy, Halima Hamid N., Almansour, Atheer Fahad, Ibrahim, Dina M., Hammoudeh, Mohammad Ali A., 2022. Braingan: brain mri image generation and classification framework using gan architectures and cnn models. Sensors 22 (11), 4297.

Amin, Kareem, Dick, Travis, Kulesza, Alex, Munoz, Andres, Vassilvitskii, Sergei, 2019. Differentially private covariance estimation. Adv. Neural Inf. Process. Syst. 32.

Assefa, Samuel A., Dervovic, Danial, Mahfouz, Mahmoud, Tillman, Robert E., Reddy, Prashant, Veloso, Manuela, 2021. Generating synthetic data in finance: opportunities, challenges and pitfalls. In: Proceedings of the First ACM International Conference on AI in Finance. ICAIF '20. Association for Computing Machinery, New York, NY, USA.

Aubry, Pascal, Carpov, Sergiu, Sirdey, Renaud, 2020. Faster homomorphic encryption is not enough: improved heuristic for multiplicative depth minimization of Boolean circuits. In: Topics in Cryptology–CT-RSA 2020: The Cryptographers' Track at the RSA Conference 2020, San Francisco, CA, USA, February 24–28, 2020, Proceedings. Springer, pp. 345–363.

Aydin, Furkan, Karabulut, Emre, Potluri, Seetal, Alkim, Erdem, Aysu, Aydin, 2022. RevEAL: single-trace side-channel leakage of the seal homomorphic encryption library. In: 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, pp. 1527–1532.

Barbedo, Jayme Garcia Arnal, 2018. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. Comput. Electron. Agric. 153, 46–53.

Barni, Mauro, Orlandi, Claudio, Piva, Alessandro, 2006. A privacy-preserving protocol for neural-network-based computation. In: Proceedings of the 8th Workshop on Multimedia and Security. MM&Sec '06. Association for Computing Machinery, New York, NY, USA, pp. 146–151.

Baruch, Moran, Drucker, Nir, Greenberg, Lev, Moshkowich, Guy, 2022. A methodology for training homomorphic encryption friendly neural networks. In: International Conference on Applied Cryptography and Network Security. Springer, pp. 536–553.

Belgodere, Brian, Dognin, Pierre, Ivankay, Adam, Melnyk, Igor, Mroueh, Youssef, Mojsilovic, Aleksandra, Navartil, Jiri, Nitsure, Apoorva, Padhi, Inkit, Rigotti, Mattia, et al., 2023. Auditing and generating synthetic data with controllable trust trade-offs. arXiv preprint arXiv:2304.10819.

Benaissa, Ayoub, Retiat, Bilal, Cebere, Bogdan, Belfedhal, Alaa Eddine, 2021. Tenseal: a library for encrypted tensor operations using homomorphic encryption. arXiv preprint arXiv:2104.03152.

Benaloh, Josh, Leichter, Jerry, 1988. Generalized secret sharing and monotone functions. In: Conference on the Theory and Application of Cryptography. Springer, pp. 27–35.

Bernau, Daniel, Grassal, Philip-William, Robl, Jonas, Kerschbaum, Florian, 2019. Assessing differentially private deep learning with membership inference. CoRR. arXiv:1912.11328 [abs].

Bhunia, Swarup, Hsiao, Michael S., Banga, Mainak, Narasimhan, Seetharam, 2014. Hardware trojan attacks: threat analysis and countermeasures. Proc. IEEE 102 (8), 1229–1247.

Boenisch, Franziska, Mühl, Christopher, Rinberg, Roy, Ihrig, Jannis, Dziedzic, Adam, 2023. Individualized pate: differentially private machine learning with individual privacy guarantees. Proc. Priv. Enh. Technol. 1, 158–176.

Boneh, Dan, Sahai, Amit, Waters, Brent, 2011. Functional encryption: definitions and challenges. In: Proceedings of the 8th Conference on Theory of Cryptography. TCC'11. Springer-Verlag, Berlin, Heidelberg, pp. 253–273.

Brakerski, Zvika, 2012. Fully homomorphic encryption without modulus switching from classical gapsvp. In: Annual Cryptology Conference. Springer, pp. 868–886.

Brakerski, Zvika, Gentry, Craig, Vaikuntanathan, Vinod, 2014. (Leveled) fully homomorphic encryption without bootstrapping. ACM Trans. Comput. Theory 6 (3), 1–36.

Brickell, Justin, Shmatikov, Vitaly, 2008. The cost of privacy: destruction of data-mining utility in anonymized data publishing. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 70–78.

Cao, Jianneng, Karras, Panagiotis, 2012. Publishing microdata with a robust privacy guarantee. arXiv preprint arXiv:1208.0220.

Carlini, Nicholas, Ippolito, Daphne, Jagielski, Matthew, Lee, Katherine, Tramer, Florian, Zhang, Chiyuan, 2023. Quantifying memorization across neural language models. In: Conference on Learning Representations, vol. 11.

Carlini, Nicholas, Liu, Chang, Erlingsson, Úlfar, Kos, Jernej, Song, Dawn, 2019. The secret sharer: evaluating and testing unintended memorization in neural networks. In: Proceedings of the 28th USENIX Conference on Security Symposium. SEC'19. USENIX Association, USA, pp. 267–284.

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

Cetin, Gizem Selcan, Dai, Wei, Opanchuk, Bogdan, Minibaev, Eugene, 2018. CuFHE: cuda-accelerated fully homomorphic encryption library. https://github.com/vernamlab/cuFHE.

Chai, Junyi, Zeng, Hao, Li, Anming, Ngai, Eric W.T., 2021. Deep learning in computer vision: a critical review of emerging techniques and application scenarios. Mach. Learn. Appl. 6, 100134.

Chamani, Javad Ghareh, Papadopoulos, Dimitrios, 2020. Mitigating leakage in federated learning with trusted hardware. arXiv preprint arXiv:2011.04948.

Charles, Zachary, Konečnỳ, Jakub, 2021. Convergence and accuracy trade-offs in federated learning and meta-learning. In: International Conference on Artificial Intelligence and Statistics. PMLR, pp. 2575–2583.

Chen, Rui, Mohammed, Noman, Fung, Benjamin C.M., Desai, Bipin C., Xiong, Li, 2011. Publishing set-valued data via differential privacy. Proc. VLDB Endow. 4 (11), 1087–1098.

Chen, Yu, Luo, Fang, Li, Tong, Xiang, Tao, Liu, Zheli, Li, Jin, 2020. A training-integrity privacy-preserving federated learning scheme with trusted execution environment. Inf. Sci. 522, 69–79.

Chen, Yudong, Su, Lili, Xu, Jiaming, 2018. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. In: Abstracts of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems, p. 96.

Cheon, Jung Hee, Han, Kyoohyung, Kim, Andrey, Kim, Miran, Song, Yongsoo, 2018. Bootstrapping for approximate homomorphic encryption. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, pp. 360–384.

Cheon, Jung Hee, Kim, Andrey, Kim, Miran, Song, Yongsoo, 2017. Homomorphic encryption for arithmetic of approximate numbers. In: International Conference on the Theory and Application of Cryptology and Information Security. Springer, pp. 409–437.

Chillotti, Ilaria, Gama, Nicolas, Georgieva, Mariya, Izabachène, Malika, 2020. Tfhe: fast fully homomorphic encryption over the torus. J. Cryptol. 33 (1), 34–91.

Chillotti, Ilaria, Gama, Nicolas, Georgieva, Mariya, Izabachène, Malika, 2016. TFHE: fast fully homomorphic encryption library. https://tfhe.github.io/tfhe/.

Choquette-Choo, Christopher A., Dullerud, Natalie, Dziedzic, Adam, Zhang, Yunxiang, Jha, Somesh, Papernot, Nicolas, Wang, Xiao, 2021a. Capc learning: confidential and private collaborative learning. arXiv preprint arXiv:2102.05188.

Choquette-Choo, Christopher A., Tramer, Florian, Carlini, Nicholas, Papernot, Nicolas, 2021b. Label-only membership inference attacks. In: Meila, Marina, Zhang, Tong (Eds.), Proceedings of the 38th International Conference on Machine Learning. 18–24 Jul. In: Proceedings of Machine Learning Research, vol. 139. PMLR, pp. 1964–1974.

Chotard, Jérémy, Dufour Sans, Edouard, Gay, Romain, Phan, Duong Hieu, Pointcheval, David, 2018. Decentralized multi-client functional encryption for inner product. In: Peyrin, Thomas, Galbraith, Steven (Eds.), Advances in Cryptology – ASIACRYPT 2018. Springer International Publishing, pp. 703–732.

Chung, Yeounoh, Haas, Peter J., Upfal, Eli, Kraska, Tim, 2018. Unknown examples & machine learning model generalization. CoRR. arXiv:1808.08294 [abs].

Clements, Joseph, Lao, Yingjie, 2019. Hardware trojan design on neural networks. In: 2019 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5.

de Cock, Martine, Dowsley, Rafael, Nascimento, Anderson C.A., Newman, Stacey C., 2015. Fast, privacy preserving linear regression over distributed datasets based on pre-distributed data. In: Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security, pp. 3–14.

European Commission, 2019. Content Directorate-General for Communications Networks, and Technology. Ethics Guidelines for Trustworthy AI. Publications Office.

OpenDP community, 2021. OpenDP: the opendp library is a modular collection of statistical algorithms that adhere to the definition of differential privacy. https://github.com/opendp/opendp.

OpenMined Community, 2020. PyDP: Python wrapper for google's differential privacy. https://github.com/OpenMined/PyDP.

PALISADE community, 2020. PALISADE: palisade lattice cryptography library. https://gitlab.com/palisade.

Cramer, Ronald, Damgård, Ivan Bjerre, et al., 2015. Secure Multiparty Computation. Cambridge University Press.

De Cristofaro, Emiliano, 2020. An overview of privacy in machine learning.

Damgård, Ivan, Pastro, Valerio, Smart, Nigel, Zakarias, Sarah, 2012. Multiparty computation from somewhat homomorphic encryption. In: Annual Cryptology Conference. Springer, pp. 643–662.

Dash, Saloni, Yale, Andrew, Guyon, Isabelle, Bennett, Kristin P., 2020. Medical time-series data generation using generative adversarial networks. In: Artificial Intelligence in Medicine: 18th International Conference on Artificial Intelligence in Medicine, AIME 2020, Minneapolis, MN, USA, August 25–28, 2020, Proceedings 18. Springer, pp. 382–391.

Datta, Pratish, Okamoto, Tatsuaki, Tomida, Junichi, 2018. Full-hiding (unbounded) multi-input inner product functional encryption from the k-linear assumption. In: Abdalla, Michel, Dahab, Ricardo (Eds.), Public-Key Cryptography – PKC 2018. Springer International Publishing, pp. 245–277.

De Montjoye, Yves-Alexandre, Hidalgo, César A., Verleysen, Michel, Blondel, Vincent D., 2013. Unique in the crowd: the privacy bounds of human mobility. Sci. Rep. 3 (1), 1–5.

Demmler, Daniel, Schneider, Thomas, Zohner, Michael, 2015. Aby-a framework for efficient mixed-protocol secure two-party computation. In: NDSS.

Deng, Li, 2012. The mnist database of handwritten digit images for machine learning research. IEEE Signal Process. Mag. 29 (6), 141–142.

Diao, Enmao, Ding, Jie, Tarokh, Vahid, 2020. Heterofl: computation and communication efficient federated learning for heterogeneous clients. arXiv preprint arXiv:2010.01264.

van Dijk, Marten, Gentry, Craig, Halevi, Shai, Vaikuntanathan, Vinod, 2010. Fully homomorphic encryption over the integers. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, pp. 24–43.

Ducas, Léo, Micciancio, Daniele, 2015. Fhew: bootstrapping homomorphic encryption in less than a second. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, pp. 617–640.

Dufour-Sans, Edouard, Gay, Romain, Pointcheval, David, 2018. Reading in the dark: Classifying encrypted digits with functional encryption. Cryptology ePrint Archive.

Dutta, Sanghamitra, Wei, Dennis, Yueksel, Hazar, Chen, Pin-Yu, Liu, Sijia, Varshney, Kush, 2020. Is there a trade-off between fairness and accuracy? A perspective using mismatched hypothesis testing. In: International Conference on Machine Learning. PMLR, pp. 2803–2813.

Dwork, Cynthia, Roth, Aaron, 2014. The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. 9 (3–4), 211–407.

Content European Commission, 2019. Directorate-General for Communications Networks and Technology. Ethics Guidelines for Trustworthy AI.

Erlingsson, Úlfar, Pihur, Vasyl, Korolova, Aleksandra, 2014. Rappor: randomized aggregatable privacy-preserving ordinal response. In: Proceedings of the 21st ACM Conference on Computer and Communications Security. Scottsdale, Arizona.

Evans, David, Kolesnikov, Vladimir, Rosulek, Mike, et al., 2018. A pragmatic introduction to secure multi-party computation. Found. Trends® Priv. Secur. 2 (2–3), 70–246.

Evfimievski, Alexandre, Gehrke, Johannes, Srikant, Ramakrishnan, 2003. Limiting privacy breaches in privacy preserving data mining. In: Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. PODS '03. Association for Computing Machinery, New York, NY, USA, pp. 211–222.

Fanti, Giulia, Pihur, Vasyl, Erlingsson, Úlfar, 2015. Building a rappor with the unknown: privacy-preserving learning of associations and data dictionaries. arXiv preprint arXiv:1503.01214.

Felzmann, Heike, Villaronga, Eduard Fosch, Lutz, Christoph, Tamò-Larrieux, Aurelia, 2019. Transparency you can trust: transparency requirements for artificial intelligence between legal norms and contextual concerns. Big Data Soc. 6 (1), 2053951719860542.

Fernandez, Virginia, Pinaya, Walter Hugo Lopez, Borges, Pedro, Tudosiu, Petru-Daniel, Graham, Mark S., Vercauteren, Tom, Cardoso, M. Jorge, 2022. Can segmentation models be trained with fully synthetically generated data? In: International Workshop on Simulation and Synthesis in Medical Imaging. Springer, pp. 79–90.

Fischer-Hübner, Simone, Angulo, Julio, Karegar, Farzaneh, Pulls, Tobias, 2016. Transparency, privacy and trust–technology for tracking and controlling my data disclosures: does this work? In: IFIP International Conference on Trust Management. Springer, pp. 3–14.

Fredrikson, Matt, Jha, Somesh, Ristenpart, Thomas, 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. CCS '15. Association for Computing Machinery, New York, NY, USA, pp. 1322–1333.

Friedman, Arik, Wolff, Ran, Schuster, Assaf, 2008. Providing k-anonymity in data mining. VLDB J. 17, 07.

Fu, Chong, Zhang, Xuhong, Ji, Shouling, Chen, Jinyin, Wu, Jingzheng, Guo, Shanqing, Zhou, Jun, Liu, Alex X., Wang, Ting, 2022. Label inference attacks against vertical federated learning. In: 31st USENIX Security Symposium (USENIX Security 22), pp. 1397–1414.

Fujita, Taisuke, 2021. AnonyPy: anonymization library for python. https://github.com/glassonion1/anonypy/.

Gentry, Craig, 2009. Fully homomorphic encryption using ideal lattices. In: Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing. STOC '09. Association for Computing Machinery, New York, NY, USA, pp. 169–178.

Geyer, Robin C., Klein, Tassilo, Nabi, Moin, 2017. Differentially private federated learning: a client level perspective. arXiv preprint. arXiv:1712.07557.

Ghanem, Sahar M., Moursy, Islam A., 2019. Secure multiparty computation via homomorphic encryption library. In: 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS), pp. 227–232.

Ghassemi, Marzyeh, Naumann, Tristan, Schulam, Peter, Beam, Andrew L., Chen, Irene Y., Ranganath, Rajesh, 2020. A review of challenges and opportunities in machine learning for health. AMIA Summits Transl. Sci. Proc. 2020, 191.

Gilad-Bachrach, Ran, Dowlin, Nathan, Laine, Kim, Lauter, Kristin, Naehrig, Michael, Wernsing, John, 2016. Cryptonets: applying neural networks to encrypted data with high throughput and accuracy. In: International Conference on Machine Learning. PMLR, pp. 201–210.

Goldsteen, Abigail, Ezov, Gilad, Shmelkin, Ron, Moffie, Micha, Farkash, Ariel, 2022. Anonymizing machine learning models. In: Garcia-Alfaro, Joaquin, Muñoz-Tapia, Jose Luis, Navarro-Arribas, Guillermo, Soriano, Miguel (Eds.), Data Privacy Management, Cryptocurrencies and Blockchain Technology. Springer International Publishing, Cham, pp. 121–136.

Goldwasser, Shafi, Gordon, S. Dov, Goyal, Vipul, Jain, Abhishek, Katz, Jonathan, Liu, Feng-Hao, Sahai, Amit, Shi, Elaine, Zhou, Hong-Sheng, 2014. Multi-input functional encryption. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, pp. 578–602.

Google, 2021. Google DP repository: libraries to generate differentially private statistics over datasets. https://github.com/google/differential-privacy.

Goyal, Vipul, Pandey, Omkant, Sahai, Amit, Waters, Brent, 2006. Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security. CCS '06. Association for Computing Machinery, New York, NY, USA, pp. 89–98.

Gürses, Seda, 2010. Pets and their users: a critical review of the potentials and limitations of the privacy as confidentiality paradigm. Identity Inf. Soc. 3 (3), 539–563.

Hall, Adam James, Jay, Madhava, Cebere, Tudor, Cebere, Bogdan, van der Veen, Koen Lennart, Muraru, George, Xu, Tongye, Cason, Patrick, Abramson, William, Benaissa, Ayoub, et al., 2021. Syft 0.5: a platform for universally deployable structured transparency. arXiv preprint arXiv:2104.12385.

Han, Kyoohyung (Kay), Hong, Seungwan, Kim, Andrey, 2016. HEAAN: ckks scheme library. https://github.com/snucrypto/HEAAN.

Hayes, Jamie, Melis, Luca, Danezis, George, De Cristofaro, Emiliano, 2017. LOGAN: evaluating privacy leakage of generative models using generative adversarial networks. CoRR. arXiv:1705.07663 [abs].

He, Zecheng, Zhang, Tianwei, Lee, Ruby B., 2019. Model inversion attacks against collaborative inference. In: Proceedings of the 35th Annual Computer Security Applications Conference. ACSAC '19. Association for Computing Machinery, New York, NY, USA, pp. 148–162.

High-Level Expert Group on AI, 2019. Ethics Guidelines for Trustworthy AI. Technical report. European Commission.

Hitaj, Briland, Ateniese, Giuseppe, Perez-Cruz, Fernando, 2017. Deep models under the gan: information leakage from collaborative deep learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. CCS '17. Association for Computing Machinery, New York, NY, USA, pp. 603–618.

Holohan, Naoise, Braghin, Stefano, Aonghusa, Pól Mac, Levacher, Killian, 2019. Diffprivlib: the IBM differential privacy library. ArXiv e-prints. arXiv:1907.02444 [cs.CR].

Huang, Po-Hsuan, Tu, Chia-Heng, Chung, Shen-Ming, 2021. Tonic: towards oblivious neural inference compiler. In: Proceedings of the 36th Annual ACM Symposium on Applied Computing, pp. 491–500.

Hunt, Hamish, Crawford, Jack L., Steffinlongo, Enrico, Shoup, Victor J., 2020. HElib: open-source software library that implements homomorphic encryption. https://github.com/homenc/HElib/.

Huo, Yuankai, Xu, Zhoubing, Moon, Hyeonsoo, Bao, Shunxing, Assad, Albert, Moyo, Tamara K., Savona, Michael R., Abramson, Richard G., Landman , Bennett A., 2018. Synseg-net: synthetic segmentation without target modality ground truth. IEEE Trans. Med. Imaging 38 (4), 1016–1025.

Hussain, Siam, Li, Baiyu, Koushanfar, Farinaz, Cammarota, Rosario, 2020. Tinygarble2: smart, efficient, and scalable Yao's Garble Circuit. In: Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice, pp. 65–67.

Ishai, Yuval, Kilian, Joe, Nissim, Kobbi, Petrank, Erez, 2003. Extending oblivious transfers efficiently. In: Annual International Cryptology Conference. Springer, pp. 145–161.

Jagielski, Matthew, Oprea, Alina, Biggio, Battista, Liu, Chang, Nita-Rotaru, Cristina, Li, Bo, 2018. Manipulating machine learning: poisoning attacks and countermeasures for regression learning. In: 2018 IEEE Symposium on Security and Privacy (SP). IEEE, pp. 19–35.

Jayaraman, Bargav, Evans, David, 2022. Are attribute inference attacks just imputation? In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. CCS '22. Association for Computing Machinery, New York, NY, USA, pp. 1569–1582.

Jia, Jinyuan, Salem, Ahmed, Backes, Michael, Zhang, Yang, Gong, Neil Zhenqiang, 2019a. Memguard: defending against black-box membership inference attacks via adversarial examples. CoRR. arXiv:1909.10594 [abs].

Jia, Jinyuan, Salem, Ahmed, Backes, Michael, Zhang, Yang, Gong, Neil Zhenqiang, 2019b. Memguard: defending against black-box membership inference attacks via adversarial examples. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. CCS '19. Association for Computing Machinery, New York, NY, USA, pp. 259–274.

Jiang, Kaifeng, Shao, Dongxu, Bressan, Stéphane, Kister, Thomas, Tan, Kian-Lee, 2013. Publishing trajectories with differential privacy guarantees. In: Proceedings of the 25th International Conference on Scientific and Statistical Database Management. SSDBM, New York, NY, USA. Association for Computing Machinery.

Jiang, Xue, Zhou, Xuebing, Grossklags, Jens, 2022. Comprehensive analysis of privacy leakage in vertical federated learning during prediction. Proc. Priv. Enh. Technol. 2022 (2), 263–281.

Jin, Xiao, Chen, Pin-Yu, Hsu, Chia-Yi, Yu, Chia-Mu, Chen, Tianyi, 2021. Cafe: catastrophic data leakage in vertical federated learning. Adv. Neural Inf. Process. Syst. 34, 994–1006.

Jordon, James, Yoon, Jinsung, Van Der Schaar, Mihaela, 2018. Pate-gan: generating synthetic data with differential privacy guarantees. In: International Conference on Learning Representations.

Kallus, Nathan, Zhou, Angela, 2018. Residual unfairness in fair machine learning from prejudiced data. In: Dy, Jennifer, Krause, Andreas (Eds.), Proceedings of the 35th International Conference on Machine Learning. 10–15 Jul. In: Proceedings of Machine Learning Research, vol. 80. PMLR, pp. 2439–2448.

Kang, Yan, Luo, Jiahuan, He, Yuanqin, Zhang, Xiaojin, Fan, Lixin, Yang, Qiang, 2022. A framework for evaluating privacy-utility trade-off in vertical federated learning. arXiv preprint arXiv:2209.03885.

Katz, Jonathan, Ranellucci, Samuel, Rosulek, Mike, Wang, Xiao, 2018. Optimizing authenticated garbling for faster secure two-party computation. In: Annual International Cryptology Conference. Springer, pp. 365–391.

Keller, Marcel, 2020. MP-SPDZ: a versatile framework for multi-party computation. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security.

Keller, Marcel, Orsini, Emmanuela, Scholl, Peter, 2016. Mascot: faster malicious arithmetic secure computation with oblivious transfer. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 830–842.

Keller, Marcel, Pastro, Valerio, Rotaru, Dragos, 2018. Overdrive: making spdz great again. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, pp. 158–189.

Kifer, Daniel, Gehrke, Johannes, 2006. Injecting utility into anonymized datasets. In: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data. SIGMOD '06. Association for Computing Machinery, New York, NY, USA, pp. 217–228.

Kim, Andrey, Papadimitriou, Antonis, Polyakov, Yuriy, 2022. Approximate homomorphic encryption with reduced approximation error. In: Cryptographers' Track at the RSA Conference. Springer, pp. 120–144.

Knott, Brian, Venkataraman, Shobha, Hannun, Awni, Sengupta, Shubho, Ibrahim, Mark, der Van Maaten, Laurens, 2021. Crypten: secure multi-party computation meets machine learning. arXiv:2109.00984.

Kusner, Matt, Gardner, Jacob, Garnett, Roman, Weinberger, Kilian, 2015. Differentially private Bayesian optimization. In: Bach, Francis, Blei, David (Eds.), Proceedings of the 32nd International Conference on Machine Learning. 07–09 Jul. In: Proceedings of Machine Learning Research, vol. 37. PMLR, Lille, France, pp. 918–927.

Law, Andrew, Leung, Chester, Poddar, Rishabh, Ada Popa, Raluca, Shi, Chenyu, Sima, Octavian, Yu, Chaofan, Zhang, Xingmeng, Zheng, Wenting, 2020. Secure collaborative training and inference for xgboost. In: Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice, pp. 21–26.

Lee, Joon-Woo, Kang, Hyungchul, Lee, Yongwoo, Choi, Woosuk, Eom, Jieun, Deryabin, Maxim, Lee, Eunsang, Lee, Junghyun, Yoo, Donghoon, Kim, Young-Sik, No, Jong-Seon, 2022. Privacy-preserving machine learning with fully homomorphic encryption for deep neural network. IEEE Access 10, 30039–30054.

Lepri, Bruno, Oliver, Nuria, Letouzé, Emmanuel, Pentland, Alex, Vinck , Patrick, 2018. Fair, transparent, and accountable algorithmic decision-making processes: the premise, the proposed solutions, and the open challenges. Philos. Technol. 31 (4), 611–627.

Li, Baiyu, Micciancio, Daniele, 2021. On the security of homomorphic encryption on approximate numbers. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, pp. 648–677.

Li, Baiyu, Micciancio, Daniele, Schultz, Mark, Sorrell, Jessica, 2022. Securing approximate homomorphic encryption using differential privacy. In: Annual International Cryptology Conference. Springer, pp. 560–589.

Li, Jeffrey, Khodak, Mikhail, Caldas, Sebastian, Talwalkar, Ameet, 2019. Differentially private meta-learning. CoRR. arXiv:1909.05830 [abs].

Li, Jiacheng, Li, Ninghui, Ribeiro, Bruno, 2020. Membership inference attacks and defenses in supervised learning via generalization gap. ArXiv. arXiv:2002.12062 [abs].

Li, Ninghui, Li, Tiancheng, Venkatasubramanian, Suresh, 2007. t-closeness: privacy beyond k-anonymity and l-diversity. In: 2007 IEEE 23rd International Conference on Data Engineering. IEEE, pp. 106–115.

Li, Oscar, Sun, Jiankai, Yang, Xin, Gao, Weihao, Zhang, Hongyi, Xie, Junyuan, Smith, Virginia, Wang, Chong, 2021. Label leakage and protection in two-party split learning. arXiv preprint arXiv:2102.08504.

Li, Qun, Thapa, Chandra, Ong, Lawrence, Zheng, Yifeng, Ma, Hua, Camtepe, Seyit A., Fu, Anmin, Gao, Yansong, 2023. Vertical federated learning: taxonomies, threats, and prospects. arXiv preprint arXiv:2302.01550.

Li, Zheng, Zhang, Yang, 2021. Membership leakage in label-only exposures. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. CCS '21. Association for Computing Machinery, New York, NY, USA, pp. 880–895.

Liu, Bo, Ding, Ming, Shaham, Sina, Rahayu, Wenny, Farokhi, Farhad, Lin, Zihuai, 2021a. When machine learning meets privacy: a survey and outlook. ACM Comput. Surv. 54 (2), 1–36.

Liu, Jian, Juuti, Mika, Lu, Yao, Asokan, Nadarajah, 2017. Oblivious neural network predictions via minionn transformations. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 619–631.

Liu, Yang, Zou, Tianyuan, Kang, Yan, Liu, Wenhan, He, Yuanqin, Yi, Zhihao, Yang, Qiang, 2021b. Batch label inference and replacement attacks in black-boxed vertical federated learning. arXiv preprint arXiv:2112.05409.

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

Long, Yunhui, Wang, Boxin, Yang, Zhuolin, Kailkhura, Bhavya, Zhang, Aston, Gunter, Carl, Li, Bo, 2021. G-pate: scalable differentially private data generator via private aggregation of teacher discriminators. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P.S., Vaughan, J. Wortman (Eds.), Advances in Neural Information Processing Systems, vol. 34. Curran Associates, Inc., pp. 2965–2977.

Luo, Xinjian, Wu, Yuncheng, Xiao, Xiaokui, Ooi, Beng Chin, 2021. Feature inference attack on model predictions in vertical federated learning. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE, pp. 181–192.

Machanavajjhala, Ashwin, Gehrke, Johannes, Kifer, Daniel, Venkitasubramaniam, Muthuramakrishna, 2006. L-diversity: privacy beyond k-anonymity. In: 22nd International Conference on Data Engineering (ICDE'06), pp. 24–36.

Madaio, Michael A., Stark, Luke, Wortman Vaughan, Jennifer, Wallach, Hanna, 2020. Co-designing checklists to understand organizational challenges and opportunities around fairness in ai. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–14.

Mamoshina, Polina, Vieira, Armando, Putin, Evgeny, Zhavoronkov, Alex, 2016. Applications of deep learning in biomedicine. Mol. Pharm. 13 (5), 1445–1454.

Mannino, Miro, Abouzied, Azza, 2019. Is this real? Generating synthetic data that looks real. In: Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology. UIST '19. Association for Computing Machinery, New York, NY, USA, pp. 549–561.

Marc, Tilen, Stopar, Miha, Benčina, Benjamin, Hartman, Jan, 2018. CiFEr/ GoFE: open-source software library that implements homomorphic encryption. https://github.com/fentec-project.

Marc, Tilen, Stopar, Miha, Hartman, Jan, Bizjak, Manca, Modic, Jolanda, 2019. Privacy-enhanced machine learning with functional encryption. In: European Symposium on Research in Computer Security. Springer, pp. 3–21.

Martins, Paulo, Sousa, Leonel, Mariano, Artur, 2017. A survey on fully homomorphic encryption: an engineering perspective. ACM Comput. Surv. 50 (6), 1–33.

Mazzone, Federico, van den Heuvel, Leander, Huber, Maximilian, Verdecchia, Cristian, Everts, Maarten, Hahn, Florian, Peter, Andreas, 2022. Repeated knowledge distillation with confidence masking to mitigate membership inference attacks. In: Proceedings of the 15th ACM Workshop on Artificial Intelligence and Security. AISec'22. Association for Computing Machinery, New York, NY, USA, pp. 13–24.

McMahan, H. Brendan, Moore, Eider, Ramage, Daniel, Hampson, Seth, y Arcas, Blaise Aguera, 2017a. Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics. PMLR, pp. 1273–1282.

McMahan, Brendan, Ramage, Daniel, Talwar, Kunal, Zhang, Li, 2017b. Learning differentially private recurrent language models. arXiv preprint arXiv:1710.06963.

Melis, Luca, Song, Congzheng, De Cristofaro, Emiliano, Shmatikov, Vitaly, 2018. Inference attacks against collaborative learning. CoRR. arXiv:1805.04049 [abs].

Menon, Aditya Krishna, Williamson, Robert C., 2018. The cost of fairness in binary classification. In: Conference on Fairness, Accountability and Transparency. PMLR, pp. 107–118.

Michalevsky, Yan, Joye, Marc, 2018. Decentralized policy-hiding abe with receiver privacy. In: European Symposium on Research in Computer Security. Springer, pp. 548–567.

Michels, Felix, Uelwer, Tobias, Upschulte, Eric, Harmeling, Stefan, 2019. On the vulnerability of capsule networks to adversarial attacks. CoRR. arXiv:1906.03612 [abs].

Mihara, Kentaro, Yamaguchi, Ryohei, Mitsuishi, Miguel, Maruyama, Yusuke, 2020. Neural network training with homomorphic encryption. arXiv preprint arXiv:2012.13552.

Milli, Smitha, Schmidt, Ludwig, Dragan, Anca D., Hardt, Moritz, 2019. Model reconstruction from model explanations. In: Proceedings of the Conference on Fairness, Accountability, and Transparency. FAT* '19. Association for Computing Machinery, New York, NY, USA, pp. 1–9.

Mishra, Pratyush, Lehmkuhl, Ryan, Srinivasan, Akshayaram, Zheng, Wenting, Popa, Raluca Ada, 2020. Delphi: a cryptographic inference service for neural networks. In: 29th USENIX Security Symposium (USENIX Security 20), pp. 2505–2522.

Mo, Fan, Haddadi, Hamed, Katevas, Kleomenis, Marin, Eduard, Perino, Diego, Kourtellis, Nicolas, 2021. Ppfl: privacy-preserving federated learning with trusted execution environments. In: Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services, pp. 94–108.

Mo, Ran, Liu, Jianfeng, Yu, Wentao, Jiang, Fu, Gu, Xin, Zhao, Xiaoshuai, Liu, Weirong, Peng, Jun, 2019. A differential privacy-based protecting data preprocessing method for big data mining. In: 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), pp. 693–699.

Mohassel, Payman, Rindal, Peter, 2018. Aby3: a mixed protocol framework for machine learning. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 35–52.

Mohassel, Payman, Rosulek, Mike, Zhang, Ye, 2015. Fast and secure three-party computation: the garbled circuit approach. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 591–602.

Müller, Martin, Salathé, Marcel, 2020. Addressing machine learning concept drift reveals declining vaccine sentiment during the COVID-19 pandemic. CoRR. arXiv:2012.02197 [abs].

Mulligan, Deirdre K., Kroll, Joshua A., Kohli, Nitin, Wong, Richmond Y., 2019. This thing called fairness: disciplinary confusion realizing a value in technology. Proc. ACM Hum.-Comput. Interact. 3 (CSCW).

Muñoz-González, Luis, Biggio, Battista, Demontis, Ambra, Paudice, Andrea, Wongrassamee, Vasin, Lupu, Emil C., Roli, Fabio, 2017. Towards poisoning of deep learning algorithms with back-gradient optimization. In: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pp. 27–38.

Nandakumar, Karthik, Ratha, Nalini, Pankanti, Sharath, Halevi, Shai, 2019. Towards deep neural network training on encrypted data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.

Narayanan, Arvind, Shmatikov, Vitaly, 2019. Robust de-anonymization of large sparse datasets: a decade later. May 21, 2019.

Nasr, Milad, Shokri, Reza, et al., 2020. Improving deep learning with differential privacy using gradient encoding and denoising. arXiv preprint arXiv:2007.11524.

Nasr, Milad, Shokri, Reza, Houmansadr, Amir, 2019. Comprehensive privacy analysis of deep learning: passive and active white-box inference attacks against centralized and federated learning. In: 2019 IEEE Symposium on Security and Privacy (SP), pp. 739–753.

Nergiz, Mehmet Ercan, Atzori, Maurizio, Clifton, Chris, 2007. Hiding the presence of individuals from shared databases. In: Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, pp. 665–676.

Neubauer, Thomas, Heurix, Johannes, 2011. A methodology for the pseudonymization of medical data. Int. J. Med. Inform. 80 (3), 190–204.

Ni, Chunchun, Cang, Li Shan, Gope, Prosanta, Min, Geyong, 2022. Data anonymization evaluation for big data and iot environment. Inf. Sci. 605, 381–392.

Nielsen, Jesper Buus, Nordholt, Peter Sebastian, Orlandi, Claudio, Burra, Sai Sheshank, 2012. A new approach to practical active-secure two-party computation. In: Annual Cryptology Conference. Springer, pp. 681–700.

Nik Aznan, Nik Khadijah, Atapour-Abarghouei, Amir, Bonner, Stephen, Connolly, Jason D., Al Moubayed, Noura, Breckon, Toby P., 2019. Simulating brain signals: creating synthetic eeg data via neural-based generative models for improved ssvep classification. In: 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–8.

Nikolaenko, Valeria, Weinsberg, Udi, Ioannidis, Stratis, Joye, Marc, Boneh, Dan, Taft, Nina, 2013. Privacy-preserving ridge regression on hundreds of millions of records. In: 2013 IEEE Symposium on Security and Privacy. IEEE, pp. 334–348.

Nissenbaum, Helen, 2004. Privacy as contextual integrity. Wash. L. Rev. 79, 119.

Nissim, Kobbi, Wood, Alexandra, 2018. Is privacy privacy? Philos. Trans. R. Soc. A, Math. Phys. Eng. Sci. 376 (2128), 20170358.

Obla, Srinath, Gong, Xinghan, Aloufi, Asma, Hu, Peizhao, Takabi, Daniel, 2020. Effective activation functions for homomorphic evaluation of deep neural networks. IEEE Access 8, 153098–153112.

Paillier, Pascal, 1999. Public-key cryptosystems based on composite degree residuosity classes. In: Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques. In: Lecture Notes in Computer Science, vol. 1592. Springer, pp. 223–238.

Papernot, Nicolas, McDaniel, Patrick, Sinha, Arunesh, Wellman, Michael P., 2018a. SoK: security and privacy in machine learning. In: Proc. of the 2018 IEEE European Symposium on Security and Privacy (EuroS&P).

Papernot, Nicolas, Song, Shuang, Mironov, Ilya, Raghunathan, Ananth, Talwar, Kunal, Erlingsson, Úlfar, 2018b. Scalable private learning with pate. arXiv preprint arXiv:1802.08908.

Park, Saerom, Byun, Junyoung, Lee, Joohee, 2022. Privacy-preserving fair learning of support vector machine with homomorphic encryption. In: Proceedings of the ACM Web Conference 2022. WWW '22. Association for Computing Machinery, New York, NY, USA, pp. 3572–3583.

Phan, Nhathai, Wu, Xintao, Hu, Han, Dou, Dejing, 2017. Adaptive Laplace mechanism: differential privacy preservation in deep learning. In: Proceedings - 17th IEEE International Conference on Data Mining. ICDM 2017, pp. 385–394.

Phong, Le Trieu, Aono, Yoshinori, Hayashi, Takuya, Wang, Lihua, Moriai, Shiho, 2018. Privacy-preserving deep learning via additively homomorphic encryption. IEEE Trans. Inf. Forensics Secur. 13 (5), 1333–1345.

Prasser, Fabian, Eicher, Johanna, Spengler, Helmut, Bild, Raffael, Kuhn, Klaus A., 2020. Flexible data anonymization using arx—current status and challenges ahead. Softw. Pract. Exp. 50 (7), 1277–1304.

Qasim, Ahmad B., Ezhov, Ivan, Shit, Suprosanna, Schoppe, Oliver, Paetzold, Johannes C., Sekuboyina, Anjany, Kofler, Florian, Lipkova, Jana, Li, Hongwei, Bjoern, Menze, 2020. Red-gan: attacking class imbalance via conditioned generation. Yet another medical imaging perspective. In: Arbel, Tal, Ben Ayed, Ismail, de Bruijne, Marleen, Descoteaux, Maxime, Lombaert, Herve, Pal, Christopher (Eds.), Proceedings of the Third Conference on Medical Imaging with Deep Learning. 06–08 Jul. In: Proceedings of Machine Learning Research, vol. 121. PMLR, pp. 655–668.

Rathee, Deevashwer, Rathee, Mayank, Goli, Rahul Kranti Kiran, Gupta, Divya, Sharma, Rahul, Chandran, Nishanth, Rastogi, Aseem, 2021. Sirnn: a math library for secure rnn inference. In: 2021 IEEE Symposium on Security and Privacy (SP). IEEE, pp. 1003–1020.

Ren, Hanchi, Deng, Jingjing, Xie, Xianghua, 2022. Grnn: generative regression neural network—a data leakage attack for federated learning. ACM Trans. Intell. Syst. Technol. 13 (4).

Riazi, Mohammad Sadegh, Weinert, Christian, Tkachenko, Oleksandr, Songhori, Ebrahim M., Schneider, Thomas, Koushanfar, Farinaz, 2018. Chameleon: a hybrid secure computation framework for machine learning applications. In: Proceedings of the 2018 on Asia Conference on Computer and Communications Security, pp. 707–721.

Rouhani, Bita Darvish, Riazi, M. Sadegh, Koushanfar, Farinaz, 2018. Deepsecure: scalable provably-secure deep learning. In: Proceedings of the 55th Annual Design Automation Conference, pp. 1–6.

Rudin, Cynthia, 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat. Mach. Intell. 1 (5), 206–215.

Ryffel, Théo, Pointcheval, David, Bach, Francis, Dufour-Sans, Edouard, Gay, Romain, 2019. Partially encrypted deep learning using functional encryption. Adv. Neural Inf. Process. Syst. 32.

Sabay, Alfeo, Harris, Laurie, Bejugama, Vivek, Jaceldo-Siegl, Karen, 2018. Overcoming small data limitations in heart disease prediction by using surrogate data. SMU Data Sci. Rev. 1 (3), 12.

Salem, Ahmed, Bhattacharya, Apratim, Backes, Michael, Fritz, Mario, Zhang, Yang, 2020. Updates-Leak: data set inference and reconstruction attacks in online learning. In: 29th USENIX Security Symposium (USENIX Security 20). USENIX Association, pp. 1291–1308.

Sayyad, Suhel, 2020. Privacy preserving deep learning using secure multiparty computation. In: 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 139–142.

Microsoft SEAL (release 4.0). https://github.com/Microsoft/SEAL. Microsoft Research, Redmond, WA, March 2022.

Shah, Muhammad A., Szurley, Joseph, Mueller, Markus, Mouchtaris, Athanasios, Droppo, Jasha, 2021. Evaluating the vulnerability of end-to-end automatic speech recognition models to membership inference attacks. In: Proc. Interspeech 2021, pp. 891–895.

Shamir, Adi, 1979. How to share a secret. Commun. ACM 22 (11), 612–613.

Shokri, Reza, Shmatikov, Vitaly, 2015. Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. CCS '15. Association for Computing Machinery, New York, NY, USA, pp. 1310–1321.

Shokri, Reza, Stronati, Marco, Song, Congzheng, Shmatikov, Vitaly, 2017. Membership inference attacks against machine learning models. In: 2017 IEEE Symposium on Security and Privacy (SP). IEEE, pp. 3–18.

Song, Congzheng, Shmatikov, Vitaly, 2019a. Auditing data provenance in text-generation models. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD '19. Association for Computing Machinery, New York, NY, USA, pp. 196–206.

Song, Congzheng, Shmatikov, Vitaly, 2019b. Overlearning reveals sensitive attributes. arXiv preprint arXiv:1905.11742.

Song, Liwei, Mittal, Prateek, 2021. Systematic evaluation of privacy risks of machine learning models. In: 30th USENIX Security Symposium (USENIX Security 21), pp. 2615–2632.

Stefanov, Emil, Van Dijk, Marten, Shi, Elaine, Chan, T-H. Hubert, Fletcher, Christopher, Ren, Ling, Yu, Xiangyao, Devadas, Srinivas, 2018. Path oram: an extremely simple oblivious ram protocol. J. ACM 65 (4), 1–26.

Stoddard, Ben, Chen, Yan, Machanavajjhala, Ashwin, 2014. Differentially private algorithms for empirical machine learning. arXiv preprint arXiv:1411.5428.

Sun, Yuwei, Chong, Ng S.T., Ochiai, Hideya, 2021. Information stealing in federated learning systems based on generative adversarial networks. In: 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 2749–2754.

Surden, Harry, 2014. Machine learning and law. Wash. L. Rev. 89, 87.

Sweeney, Latanya, 2002. k-anonymity: a model for protecting privacy. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 10 (05), 557–570.

Szegedy, Christian, Zaremba, Wojciech, Sutskever, Ilya, Bruna, Joan, Erhan, Dumitru, Goodfellow, Ian, Fergus, Rob, 2013. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.

Tensorflow team, 2019. Tensorflow privacy: a python library that includes implementations of tensorflow optimizers for training machine learning models with differential privacy. https://github.com/tensorflow/privacy.

Thakkar, Om Dipakbhai, Ramaswamy, Swaroop, Mathews, Rajiv, Beaufays, Francoise, 2021. Understanding unintended memorization in language models under federated learning. In: Proceedings of the Third Workshop on Privacy in Natural Language Processing. Association for Computational Linguistics, pp. 1–10. Online.

Tramer, Florian, Boneh, Dan, 2018. Slalom: fast, verifiable and private execution of neural networks in trusted hardware. arXiv preprint arXiv:1806.03287.

Tramèr, Florian, Shokri, Reza, San Joaquin, Ayrton, Le, Hoang, Jagielski, Matthew, Hong, Sanghyun, Carlini, Nicholas, 2022. Truth serum: poisoning machine learning models to reveal their secrets. arXiv preprint arXiv:2204.00032.

Tramèr, Florian, Zhang, Fan, Juels, Ari, Reiter, Michael K., Ristenpart, Thomas, 2016. Stealing machine learning models via prediction {APIs}. In: 25th USENIX Security Symposium (USENIX Security 16), pp. 601–618.

Trask, Andrew, Bluemke, Emma, Garfinkel, Ben, Cuervas-Mons, Claudia Ghezzou, Dafoe, Allan, 2020. Beyond privacy trade-offs with structured transparency. CoRR. arXiv:2012.08347 [abs].

Truex, Stacey, Liu, Ling, Gursoy, Mehmet Emre, Yu, Lei, Wei, Wenqi, 2018. Towards demystifying membership inference attacks. CoRR. arXiv:1807.09173 [abs].

Vila, Laura Cross, Escolano, Carlos, Fonollosa, José A.R., Costa-Jussa, Marta R., 2018. End-to-end speech translation with the transformer. In: IberSPEECH, pp. 60–63.

Wang, Xiao, Malozemoff, Alex J., Katz, Jonathan, 2016. EMP-toolkit: efficient MultiParty computation toolkit. https://github.com/emp-toolkit.

Wang, Xiao, Ranellucci, Samuel, Katz, Jonathan, 2017. Authenticated garbling and efficient maliciously secure two-party computation. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 21–37.

Wang, Yilei, Lv, Qingzhe, Zhang, Huang, Zhao, Minghao, Sun, Yuhong, Ran, Lingkai, Li, Tao, 2023. Beyond model splitting: preventing label inference attacks in vertical federated learning with dispersed training. World Wide Web, 1–17.

Weng, Haiqin, Zhang, Juntao, Ma, Xingjun, Xue, Feng, Wei, Tao, Ji, Shouling, Zong, Zhiyuan, 2020. Practical privacy attacks on vertical federated learning. arXiv preprint arXiv:2011.09290.

Wondracek, Gilbert, Holz, Thorsten, Kirda, Engin, Kruegel, Christopher, 2010. A practical attack to de-anonymize social network users. In: Proceedings of the 2010 IEEE Symposium on Security and Privacy. SP '10. IEEE Computer Society, USA, pp. 223–238.

Wu, Bang, Yang, Xiangwen, Pan, Shirui, Yuan, Xingliang, 2020a. Model extraction attacks on graph neural networks: taxonomy and realization. CoRR. arXiv:2010.12751 [abs].

Wu, Xi, Fredrikson, Matt, Jha, Somesh, Naughton, Jeffrey F., 2016. A methodology for formalizing model-inversion attacks. In: 2016 IEEE 29th Computer Security Foundations Symposium (CSF), pp. 355–370.

Wu, Zuxuan, Lim, Ser-Nam, Davis, Larry S., Goldstein, Tom, 2020b. Making an invisibility cloak: real world adversarial attacks on object detectors. In: European Conference on Computer Vision. Springer, pp. 1–17.

Xu, Runhua, Baracaldo, Nathalie, Zhou, Yi, Anwar, Ali, Ludwig, Heiko, 2019a. Hybridalpha: an efficient approach for privacy-preserving federated learning. In: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, pp. 13–23.

Xu, Runhua, Joshi, James B.D., Li, Chao, 2019b. Cryptonn: training neural networks over encrypted data. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). IEEE, pp. 1199–1209.

Xue, Mingfu, Yuan, Chengxiang, Wu, Heyi, Zhang, Yushu, Liu, Weiqiang, 2020. Machine learning security: threats, countermeasures, and evaluations. IEEE Access 8, 74720–74742.

Yagisawa, Masahiro, 2015. Fully homomorphic encryption without bootstrapping. Cryptology ePrint Archive.

Yang, Chao-Han Huck, Siniscalchi, Sabato Marco, Lee, Chin-Hui, 2021. Pate-aae: incorporating adversarial autoencoder into private aggregation of teacher ensembles for spoken command classification. In: Interspeech.

Yang, Haomiao, Ge, Mengyu, Xiang, Kunlan, Li, Jingwei, 2023. Using highly compressed gradients in federated learning for data reconstruction attacks. IEEE Trans. Inf. Forensics Secur. 18, 818–830.

Yang , Kang, Weng, Chenkai, Lan, Xiao, Zhang, Jiang, Wang, Xiao, 2020a. Ferret: fast extension for correlated ot with small communication. In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, pp. 1607–1626.

Yang, Mengwei, Song, Linqi, Xu, Jie, Li, Congduan, Tan, Guozhen, 2019a. The tradeoff between privacy and accuracy in anomaly detection using federated xgboost. arXiv preprint arXiv:1907.07157.

Yang, Qiang, Liu, Yang, Chen, Tianjian, Tong, Yongxin, 2019b. Federated machine learning: concept and applications. ACM Trans. Intell. Syst. Technol. 10 (2), 1–19.

Yang, Ziqi, Shao, Bin, Xuan, Bohan, Chang, Ee-Chien, Zhang, Fan, 2020b. Defending model inversion and membership inference attacks via prediction purification. CoRR. arXiv:2005.03915 [abs].

Yang, Ziqi, Zhang, Jiyi, Chang, Ee-Chien, Liang, Zhenkai, 2019c. Neural network inversion in adversarial setting via background knowledge alignment. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. CCS '19. Association for Computing Machinery, New York, NY, USA, pp. 225–240.

Yao, Andrew C., 1982. Protocols for secure computations. In: 23rd Annual Symposium on Foundations of Computer Science (sfcs 1982). IEEE, pp. 160–164.

Yao, Lin, Wang, Xue, Hu, Haibo, Wu, Guowei, 2023. A utility-aware anonymization model for multiple sensitive attributes based on association concealment. IEEE Trans. Dependable Secure Comput., 1–12.

Ye, Dongdong, Yu, Rong, Pan, Miao, Han, Zhu, 2020. Federated learning in vehicular edge computing: a selective model aggregation approach. IEEE Access 8, 23920–23935.

Yeom, Samuel, Fredrikson, Matt, Jha, Somesh, 2017. The unintended consequences of overfitting: training data inference attacks. CoRR. arXiv:1709.01604 [abs].

Yin, Hongxu, Mallya, Arun, Vahdat, Arash, Alvarez, Jose M., Kautz, Jan, Molchanov, Pavlo, 2021. See through gradients: image batch recovery via gradinversion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16337–16346.

Yoo, Joon Soo, Yoon, Ji Won, 2021. t-bmpnet: trainable bitwise multilayer perceptron neural network over fully homomorphic encryption scheme. Secur. Commun. Netw. 2021, 1–19.

Yousefpour, Ashkan, Shilov, Igor, Sablayrolles, Alexandre, Testuggine, Davide, Prasad, Karthik, Malek, Mani, Nguyen, John, Ghosh, Sayan, Bharadwaj, Akash, Zhao, Jessica, Cormode, Graham, Mironov, Ilya, 2021. Opacus: user-friendly differential privacy library in PyTorch. arXiv preprint arXiv:2109.12298.

Zhang, Chiyuan, Ippolito, Daphne, Lee, Katherine, Jagielski, Matthew, Tramèr, Florian, Carlini, Nicholas, 2021. Counterfactual memorization in neural language models. arXiv preprint arXiv:2112.12938.

Zhang, Xiaojin, Kang, Yan, Chen, Kai, Fan, Lixin, Yang, Qiang, 2022. Trading off privacy, utility and efficiency in federated learning. arXiv preprint arXiv:2209.00230.

Zhao, Bo, Mopuri, Konda Reddy, Bilen, Hakan, 2020. idlg: improved deep leakage from gradients. arXiv preprint arXiv:2001.02610.

Zheng, Wenting, Deng, Ryan, Chen, Weikeng, Popa, Raluca Ada, Panda, Aurojit, Stoica, Ion, 2021. Cerebro: a platform for {Multi-Party} cryptographic collaborative learning. In: 30th USENIX Security Symposium (USENIX Security 21), pp. 2723–2740.

Zheng, Wenting, Popa, Raluca Ada, Gonzalez, Joseph E., Stoica, Ion, 2019. Helen: maliciously secure coopetitive learning for linear models. In: 2019 IEEE Symposium on Security and Privacy (SP). IEEE, pp. 724–738.

Zhu, Ligeng, Liu, Zhijian, Han, Song, 2019. Deep leakage from gradients. Adv. Neural Inf. Process. Syst. 32.

Zou, Minhui, Shi, Yang, Wang, Chengliang, Li, Fangyu, Song, Wen-Zhan, Wang, Yu, 2018. Potrojan: powerful neural-level trojan designs in deep learning models. CoRR. arXiv:1802.03043 [abs].

Zou, Yang, Zhang, Zhikun, Backes, Michael, Zhang, Yang, 2020. Privacy analysis of deep learning in the wild: membership inference attacks against transfer learning. CoRR. arXiv:2009.04872 [abs].

**Soumia Zohra El Mestari:** Soumia received her Master's along with her engineering degrees from the Ecole Superieure d'informatique (Sidi Bel Abbes, Algeria), in 2020. Her research interests are in Machine Learning, Trust and Transparency in data-driven tools and Privacy-Preserving machine learning. Prior to joining the Sociotechnical Cybersecurity Interdisciplinary research group, IRiSC, headed by Prof. Gabriele Lenzini, Soumia worked as a machine learning engineer and data analyst. Currently she is pursuing her PhD in IRiSC funded by the interdisciplinary EU project Legality Attentive Data Scientists (LeADS).

**Prof. Dr. Gabriele Lenzini:** Gabriele Lenzini, Associate professor at the University of Luxembourg and Chief Scientist II at the University Center for Reliability, Security and Trust (SnT). Heads the Interdisciplinary Research group in Sociotechnical Security (IRiSC). Lenzini has more than 15 years of experience in the design and analysis of security and private systems, a topic addressed using a toolkit of different research methods, formal and experimental. His most recent interest also addresses compliance with current data protection regulation, as well as the design of solutions attentive to related European directives on cyber-security and cyber-defense.

**Dr Huseyin Demirci:** Huseyin Demirci received his Ph.D. degree from Marmara University (Turkey), in 2004.

He worked as a researcher on several areas, such as mathematical analysis of the security systems, genome sequencing, and bioinformatics. His main research interests include cryptology with a specific focus on design and analysis of symmetric encryption systems, lightweight and wireless security, bioinformatics, NGS data analysis and genomic privacy and authentication systems.

He participated in the construction of the first next generation genome sequencing center of Turkey and the identification of many gene-disease associations.

Huseyin works at the Sociotechnical Cybersecurity Interdisciplinary research group, IriSC, University of Luxembourg.