CS8692 - Comprehensive Studies in Selected Topics in Computer Science

City University of Hong Kong

Department of Computer Science

When Deep Learning Meets Differential Privacy: Privacy, Security, and More
Final Report

Author:
Li Xinyan (ID: 55670594)

Supervisor:
Prof. Cong Wang

Date: November 8, 2021

Abstract

Over the past ten years, the research community has witnessed the unprecedented developments of deep learning (DL), along with the growing concerns in its privacy, security, and intrinsic vulnerability risks. As a canonical privacy-preserving criterion, differential privacy (DP) [23] are quickly adopted by researchers to mitigate the privacy risks of deploying DL models in real-life applications [1, 52].

Recently, we observed that some studies successfully extended the notion of DP to strengthen the security or reduce intrinsic vulnerabilities of DL models [43, 17, 19]. Their pioneering attempts prove the feasibility of constructing DP-based methods to solve deep learning problems.

To comprehensively deliver the latest research progress and facilitate further evaluations, we survey the most representative and novel breakthroughs of DP in the DL context. We first introduce the notion of DP from its essence to extensions, laying the groundwork for the following discussion on its feasibility of solving DL problems. Then, we categorize the DL problems into privacy issues, security issues, and intrinsic vulnerabilities. DP-based countermeasures and mitigation are presented subsequently with in-depth discussion.

More importantly, we reflect on the potential side effects of integrating DP into the training process of DL models, which were overlooked by the previous surveys. The summarized re-evaluations, remaining challenges, and future directions will offer a broader view for later scholars interested in this topic.

Contents

1 Introduction

Machine Learning (ML) is a branch of artificial intelligence (AI) where the learning model can mimic human learning behaviors and extract knowledge from data. Deep learning (DL) is a technique for implementing ML. The DL models can automatically learn the underlying patterns that explain the input training dataset and accurately output predictions for unseen data based on the patterns it learned during the training process [23].

Compared to the traditional machine learning models, the DL model adopts a more complex architecture called neural networks [34]. The neural networks have a large number of parameters and are based on layer structure, so they can remember higher-dimensional features and learn the complicated rules. DL models have been widely adopted in real-world sophisticated decision-making tasks, such as self-driving cars, automated medical diagnosis [69], intelligent recommender systems [48], and audio recognition [29].

While benefiting from the substantial computing power of the DL, our researchers do not fully understand the very details of neural networks. Due to the inexplicable high-dimensional feature spaces and the iterative training process, the neural network structure has low interpretability, making it difficult to explain certain model behaviors and vulnerabilities. A variety of DL privacy issues, security issues, and intrinsic vulnerabilities are observed and reported by the research community. The privacy and security attacks reduce the reliability of replacing the human decision-making process with DL models, while the intrinsic vulnerabilities reflect our insufficient understanding of DL models. These problems will surely threaten the healthy DL ecosystem and hinder the further development of DL techniques if left unaddressed.

At this point, some studies prove that introducing calibrated randomness into the training or predicting phase of neural networks can help to stabilize their performance [1, 23, 43].

Differential privacy (DP) is a canonical privacy definition in privacy-preserving data analysis [23]. Considering its time-proving achievements in the traditional privacy-preserving domain, the DL community has tried and successfully managed to formalize DP-based defenses for DL privacy issues [1, 52]. Moreover, we observe some recent studies start to apply DP in mitigating DL security problems or interpreting the models' intrinsic vulnerabilities.

We notice that the previous surveys might overlook the inherent connections between DL and DP and merely introduce the possibility of such combinations. To distinguish our contributions, we further reflect on the essences, interpretability, limitations, challenges, and advantages of utilizing DP definition to strengthen DL performance.

This work aims to comprehensively deliver the latest research progress and facilitate further evaluations of the researchers from both DL and DP domains. For DL researchers, we deliver the recent successful combinations between DP and DL. More importantly, we stress the inherent linkages of such combinations, encouraging them to discover other novel usages of DP in other DL problems. For DP researchers, we hope to provide them with a comprehensive view of the recent DL achievements and problems. Because DL still lacks interpretability even today, maybe some DP researchers with abundant mathematical knowledge can find possible answers from a unique angle. This survey is also friendly to beginners who are interested in DL and DP. They can explore the latest discussion on DP privacy, security, and intrinsic vulnerabilities.

Considering the tremendous amount of literature in DP and DL, we narrow down the reviewing scope to the novel studies published on the top publications within the last ten years and reserve more space to discuss the most representative breakthroughs of DP in the DL context. The collected literature will next be presented based on our taxonomy with comprehensive descriptions in a clearly and concisely. On this basis, we will try to re-evaluate the literature starting from the canonical DP theory, ensuring the rigorousness of our conclusions.

To clarify the following discussion, we use ***privacy issues*** to describe the malicious activities where the adversary tries to gain unauthorized and private information from the DL models. These malicious activities are separate from ***security issues***, which describe the malicious activities that the adversary tries to mislead or interfere with the decision of the DL models. Then, the ***intrinsic vulnerabilities*** refer to the inherent vulnerable characteristics of DL models, which might be further exploited by the adversary to conduct privacy or security attacks without careful evaluations.

This survey is structured as follows: We start by introducing the essence and extensions of DP in Section 2, laying the groundwork for the following discussion on its feasibility of solving DL problems. Then, we categorize the DL problems into privacy issues (Section 3), security issues (Section 4), and intrinsic vulnerabilities (Section 4). DP-based countermeasures and mitigation are presented subsequently with in-depth discussion. In Section 5, we reflect on the satisfying consequence or potential side effects of integrating DP into the training or predicting process of DL models, which were overlooked by the previous surveys. The summarized re-evaluations, remaining challenges, and future directions will offer a broader view for later scholars interested in this topic.

2 Differential Privacy: From the Essence to Extensions

This section will introduce some concepts, properties, and applications of differential privacy (DP). The use cases are organized from DP's original query-release usage to subsequent extensions in different application scenarios. The following contents will provide the readers with a solid background before diving into the discussion of combining DP and DL.

2.1 Concept and Property

2.1.1 Concept

Differential Privacy [23] is a canonical privacy-preserving criterion offering the provable privacy guarantee on every individual's information in the dataset. In general, there are three components in the DP framework: input datasets, DP mechanism, and outputs. Any DP mechanism (aka. algorithm) satisfying the (ϵ, δ)-DP definition (See Eq. (1)) is guaranteed to provide indistinguishability for the input dataset, which means the adversary cannot re-identify whether an individual's record was included in the input dataset through observing the outputs.

$$\Pr[A(D) = S] \leq e^{\epsilon} \Pr[A(D') = S] + \delta \tag{1}$$

Specifically, D and D' are any two neighboring datasets with and without a specific participant's record (e.g., with or without Alice's record in Fig. 1). A (ϵ, δ)-DP mechanism A takes these two neighboring datasets as input and produces output S. The (ϵ, δ)-DP definition (1) shows that the DP mechanism A guarantees the adversary cannot distinguish the output S is produced by applying A on dataset D or D' due to the close probabilities.

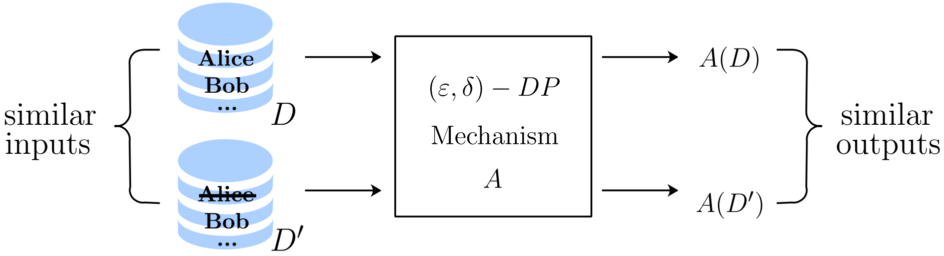


Figure 1: A example of DP framework.

Note that DP mechanism A is a probabilistic algorithm, which means the indistinguishability guarantee of the mechanism's outputs comes from the randomness in mechanism A. There are two parameters in Eq. (1) that control the level of indistinguishability privacy guarantee.

First, ϵ is known as the privacy budget. It decides how much randomness (calibrated noise) will be interpolated to the outputs. The smaller ϵ suggests the closer relationship of two probabilities ($\Pr[A(D)=S]$ and $\Pr[A(D')=S]$), yielding a stronger privacy guarantee at the cost of less accurate outputs (since it introduces the larger amount of noise to the outputs).

Second, δ is a relaxation parameter that allows the privacy guarantee to fail with a negligible probability. The existence of δ can reduce the amount of noise (i.e., obtain more accurate outputs) needed to reach a higher level of privacy guarantee, but the mechanism might get no guarantee (the DP inequality (1) does not hold) with the negligible probability δ.

2.1.2 Property

Several essential properties create the rich diversity of DP mechanisms and provide the theoretical basis for further DP extensions.

Firstly, the composition theories guarantee that DP mechanisms X and Y can be composed sequentially or parallelly to form a new DP mechanism Z. The resulting DP mechanism Z will still satisfy the DP definition with proper design. We will not expand the mathematical details in this survey. Interested readers can find the exhaustive explanation and deduction in this book [23].

The second property of the DP mechanism is called post-processing. Any deterministic functions F (or their convex combinations) that apply on top of the outputs of any DP mechanisms will preserve the privacy guarantee. In other words, it is impossible to reverse the privacy protection provided by DP mechanisms, which means performing arbitrary subsequent computations on the output of a DP mechanism will not harm the original privacy guarantee. As a result, knowing the outputs of DP mechanisms will not increase the adversary's advantage of de-privatizing the original inputs (this is why DP mechanisms can achieve private data publishing).

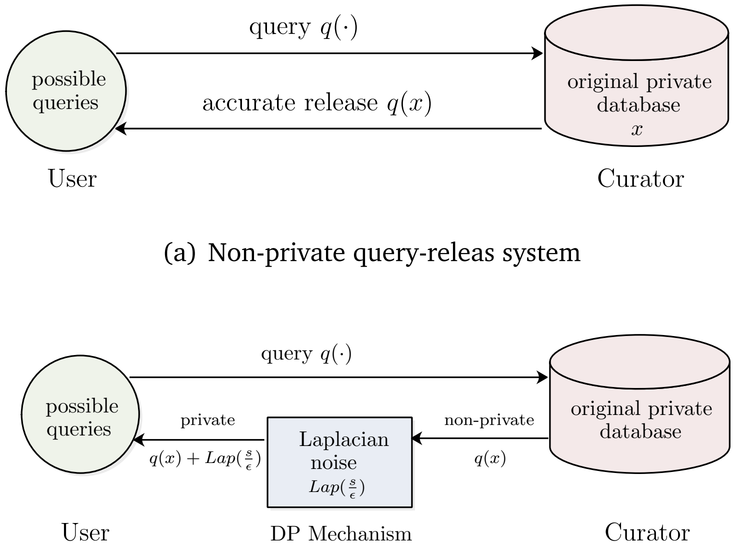
The third property of the DP mechanism is called group privacy. In Fig. 1, the DP mechanism provides the privacy guarantee for neighboring datasets that differ only in one record (Bob's record). Thanks to the exponential relationship between the two probabilities ($\Pr[A(D)=S]$ and $\Pr[A(D')=S]$), the guarantee can be extended to a group of k records. As shown in Eq. (2), the DP guarantee still holds for any neighboring datasets D and D' with and without a group of k participants' records (or differs in k records).

$$\Pr[A(D)=S] \leq e^{k\epsilon} \Pr[A(D')=S] + ke^{(k-1)\epsilon}\delta \tag{2}$$

The cost of group privacy is revealed in the degrade of privacy level: the (ϵ, δ)-DP becomes $(k\epsilon, ke^{(k-1)\epsilon}\delta)$-DP with the larger privacy budget ϵ (less private) and high failure probability δ.

2.2 Privacy-Preserving Query-Release Problem

A typical non-private query-release system is modeled in Fig. 2(a), where the trusted curator holds the private dataset, and the untrusted user can query the curator to obtain statistical information of the private dataset.



(a) Non-private query-releas system

(b) DP-based private query-releas system

Figure 2: The typical query-release systems.

In the non-private setting, the query q is a deterministic function over the original dataset. The untrusted user can easily exploit the auxiliary information and query function to extract individual information. Consider the following two non-private queries:

a) *What is the number of infected people in Dataset DISEASE?*
>>> 1000.0

b) *What is the number of infected people whose ID is not ‘001’ in Dataset DISEASE?*
>>> 999.0

If the adversary knows Mr. X’s ID is ‘001’, then he can conclude that Mr. X is infected by observing the changes in the accurate outputs of two counting queries. Actually, this is a typical *differencing attack*. It gives us a hint that accurately responding to queries can damage individual privacy. Randomness is a must.

$$F(x) = q(x) + Lap(\frac{s}{\epsilon}) \tag{3}$$

The *Laplace mechanism* F is a basic DP mechanism that samples noise from the Laplace distribution and adds them to the non-private deterministic query results $q(x)$ (See Fig. 2(b)). As shown in Eq. 3, the noise comes from the Laplace distribution $Lap(\frac{s}{\epsilon})$, where s stands for the sensitivity of query function q.

Because the query function q is pre-defined and deterministic, we can measure the amount of changes in $q(x)$ when its input x changes. The maximum of all the possible changes of query function q is fixed and represented by sensitivity s. The sensitivity s and the privacy budget ϵ jointly decide the amount of noise added by the mechanism F. Owing to the random noise drawn from the Laplace distribution $Lap(\frac{s}{\epsilon})$, the outputs of the mechanism F is now probabilistic and satisfy the DP definition in Eq. (1).

In the above example, the accurate outputs of query a) and b) are 1000 and 999 respectively. When applying the Laplace mechanism, every query output will be combined with the noise sampled from a Laplace distribution before releasing to the untrusted user. Image that if the noise sampled for query a) and b) are -0.5 and 0.1 respectively, then the user will obtain:

a) *What is the number of infected people in Dataset DISEASE?*
>>> 999.5

b) *What is the number of infected people whose ID is not '001' in Dataset DISEASE?*
>>> 999.1

In the privacy-preserving query-release scenario, the probabilistic DP mechanism F warps around the deterministic query function q to yield the noisy outputs. We comment on the intuition and effects of randomness in the probabilistic DP mechanisms. On the one hand, a certain amount of randomness allows the mechanism to hide the differences in the outputs when querying neighboring datasets. Note that the differences between outputs are not eliminated but only partially covered by the randomness in the mechanisms. Thus, DP mechanisms provide a deniability of being involved in the query result, and this is where privacy comes from.

On the other hand, the amount of randomness is calibrated and bounded by special distributions according to the privacy budget and the sensitivity (the maximum changes in $q(x)$ when the input x changes) of the query function. The statistical properties of these special distributions (e.g., the Exponential, Laplace, and Gaussian distribution) further bring the DP mechanisms theoretical proofs on the indistinguishable probabilities, which results in the provable privacy guarantee.

2.3 Differential Privacy Extensions

Zhu et al. [78] present a detailed survey of the extended usage of DP notion, including the private online social network (extended DP mechanisms for graph structure data), privacy-preserving recommender system (extended DP mechanisms for stream data and continual release), location privacy (extended DP mechanisms for sparse location data), and others.

We mention these extensions to illustrate DP's potential in solving different problems. In the private online social network problem [13], the DP mechanisms ensure the query results over two neighboring graphs (graphs that differ in one node or one edge) are indistinguishable.

For the privacy-preserving recommender system [47], the user's shopping history is represented by a continual bitstream recording of whether they had purchased an item or not during a time period. The goal of DP mechanisms is to release the statistical information of the bitstream without publishing the user's decision on every item.

The location privacy aims to release the individual's 2D location in a private manner. The DP mechanisms can help the user protect their actual location while disclosing adequate geometric information to maintain the service usability [11].

2.4 Differential Privacy in Machine Learning

Dwork et al. [23] describe the machine learning task as Valiant's Probably Approximately Correct (PAC) problem [64], where the learning problem is defined as an unknown distribution D.

Firstly, the machine learning algorithm is given a number of labeled samples (X, Y) (i.e., the training dataset) drawn from an unknown distribution D. The learning goal is to approximate the target distribution D by adjusting the learning model F. The learning model F is a functional mapping that maps the input X to specific output Y based on the mapping rules, so it can be seen as a function F with plenty of parameters, including weights, bias, and hyperparameters.

During the training process, the learning model automatically adapts its parameters to minimize the distance between D and F (i.e., the model learns the best patterns by solving optimization problems and stores the learned patterns in the model parameters). Mathematically, the distance is measured as the loss (prediction error) between the model's current prediction and the ground truth. The learning algorithm solves the optimization problem on F with the objective of minimizing the loss.

After the successful learning process, the well-trained model F should accurately predict the labels of new samples drawn from the same distribution D.

Kasiviswanathan et al. [41] raise the question of private learning: when the training dataset contains sensitive individual information, how much privacy is preserved if we publish the well-trained model f that approximates the distribution of private dataset? Moreover, can we formulate a provable privacy guarantee for individual records in the training dataset?

Recall that DP has a special property called post-processing, which means if we can integrate DP guarantee into any stage along the machine learning pipeline (training or predicting progress), we will obtain the model predictions that satisfy the DP notion for neighboring inputs. Furthermore, the privacy level of the model is quantified and provable under the mathematical proofs under the DP framework.

There have been a lot of studies that privatize the model learning pipeline using the DP notion, including input perturbation [32], objective perturbation [67], gradient perturbation [1], and output perturbation [12, 52]. Further discussion on the learning pipeline will be provided in Section 3.

When privatizing the model learning pipeline using the DP notion, the researchers observe an interesting phenomenon: DP strongly connects to the model's stability and generalization ability. Stability is a measure of the extent to which a mechanism's output changes when its input changes [22]. The DP definition naturally offers the stability guarantee: the distribution of a DP mechanism's output is probabilistically bounded in the definition and guarantees not to depend too much on a specific record in the input. In other words, the outputs of a DP mechanism for neighboring inputs are stable (or indistinguishable). Dworl et al. [22] further link the stability guarantee with DP's generalization ability in adaptive data analysis (e.g., machine learning). Further discussion on the model's generalization ability will be provided in Section 4.2.1.

3 Differential Privacy in Deep Learning: Privacy

3.1 Deep Learning Privacy Issues

Compared to the traditional model in machine learning, the DL model adopts a more complex model architecture called neural networks [34]. A classic neural network model has multiple layers (See Fig. 3): one input layer (the first layer), several hidden layers, and one output layer (the last layer). Each layer consists of a certain number of neurons that store parameters.

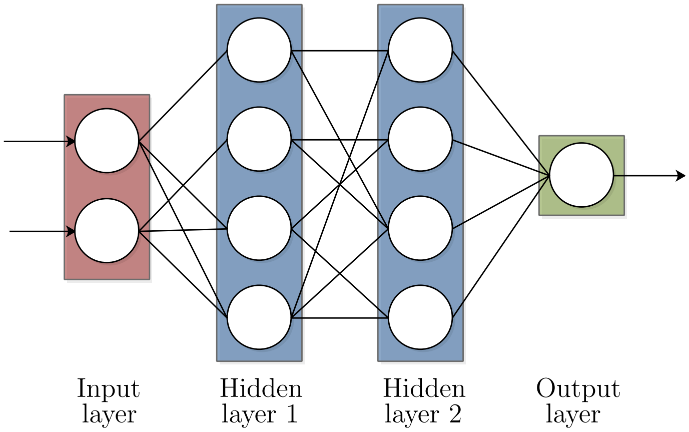


Figure 3: A classic neural network model.

The neural networks have more parameters and are based on layer structure to remember higher-dimensional features and complicated rules. The price for the outstanding expressive ability is the poor interpretability due to the neural networks' non-convexity, making it difficult to explain certain model behaviors and vulnerabilities (Section 4 discusses the meaning of inexplicable behaviors and why they are hard to interpret).

As such, the not-yet-fully-studied behaviors and vulnerabilities can be exploited by adversaries to conduct privacy and security attacks. In this section, we categorize the DL privacy issues based on what private information the adversary is trying to conquer: a) the training Data-oriented attacks; b) the model-oriented attacks.

Next, DP-based countermeasures are classified based on where they are implemented throughout the DL model's learning process, followed by the reflections and conclusions of applying DP to mitigate DL security issues.

3.1.1 Training Data-Oriented Attacks

Before introducing the privacy attacks on training data, we need to set up a plausible notion of training data privacy.

In private statistical database publishing, Dalenius et al. [15] claim that privacy is achieved if individual information cannot be learned from the database when one does not have access to this database. This assertion is later proved unachievable by Dwork et al. [21], and they propose a new standard for the privacy-preserving database with meaningful statistical usability: the statistical database should be able to release statistical information of the dataset as a whole while protecting the privacy of every individual record.

For example, we allow the lung cancer dataset to release the statistical fact that people who smoke have a higher risk of developing lung cancer. But the dataset should not reveal whether the participant Bob has lung cancer or Bob is a heavy smoker.

To a certain extent, the new standard for privacy-preserving aligns with the goal of machine learning tasks: the learning model should capture the statistical information of the population (e.g., smoking causes lung cancer) rather than remembering the specific record in the training dataset (e.g., Bob's record). Unfortunately, the DL models do not stop at the underlying patterns in the training dataset. They actually remember too many details about the training dataset.

Membership Inference Attack: Zhang et al. [76] point out that modern DL models have the sufficient capacity to memorize the training dataset. The modern DL models have a large number of parameters, which is enough to appropriate a very complex distribution in high-dimensional space. Recall that during the model's training process, the learning objective is to minimize the loss (distance) by adjusting the network parameters (weights, bias, hyperparameters). Thus, the DL model will try to fit every record in the training dataset as close as possible.

Song et al. [58] describe this phenomenon using the term 'over memorized' where the DL models will have higher confidence on the data record that it was trained on. Since records in the training dataset will be fed to the model several times during the training process in order to minimize the loss, records that belong to the training dataset will have a much smaller prediction loss than those that are not.

Shokri et al. [57] show that an adversary can infer whether a specific image was included in the training dataset by observation the well-trained DL model's prediction (confidence scores). The confidence score is a probability vector where each row suggests the probability that the input image belongs to a label.

A similar attack was later proposed by Carlini et al. [9] but targeting the nature language processing model. They craft some 'canary' (some sensitive phrase with special format) and insert them into the training dataset. For example, a canary can be a sentence with sensitive information: '*My social security number is 012-345-6789*'. The well-trained DL model needs to compute the perplexity [1] for the given sentences.

They prove that the DL model remembers the canary, and the adversary can even extract the sensitive information in the canary from the final model. In this example, the perplexity of the sentence '*My social security number is 000-000-0000*' will be higher than the inserted canary. The adversary can infer the canary with dominant advantages by trying different combinations of the sensitive information in the tested sentence.

Besides observing the confidence scores and the perplexity, another study achieves meaningful membership inference with label-only outputs. In other words, the presence or absence of an individual's record is no longer private, which is undesirable for participants who contribute their private information to train some disease diagnostic models (e.g., HIV/ADIS). This violation is called a *membership inference attack* [57].

Attribute Inference Attack: On the basis of membership inference, some studies [24] disclosed another violation called *attribute inference attack*. [2] The attribute inference attack assumes that an adversary holds the incomplete records in the training dataset (e.g., the adversary only knows some insensitive attributes of records) and tries to infer the missing attributes of records (which are private and sensitive). Fredrikson et al. [24] show how an adversary uses incomplete medical records and a well-trained model to recover the sensitive genotype attribute successfully. Their attack algorithm maximizes the posterior probability of observing the private attributes given the nonsensitive attributes and the query access to a well-trained model.

3.1.2 Model-Oriented Attacks

Model Stealing: The model stealing attack [62] and the hyperparameter stealing attack [66] are representative examples of model-oriented attacks.

In model-oriented attacks, the adversary aims to steal the well-trained DL models or their parameters instead of the private information in the training dataset. The training and fine-tuning process of DL models is costly and time-consuming. Some technology companies keep the well-trained DL models as their trade secrets, making profits by offering query APIs or charging for the DL model's prediction (predictive analytics) rather than providing public access to the well-trained DL model. This business mode is known as 'Machine Learning as a Service' (MLaaS).

Tramer et al. [62] assume the adversary only has the black-box query access to some DL models deployed on the remote cloud server. By querying the model APIs with abundant inputs and collecting the corresponding outputs (confidence scores or labels), the

[1] The perplexity of a given sentence captures the degree of uncertainty a model has in predicting. The lower perplexity suggests the model's higher confidence in predicting.

[2] The attribute inference attack is also known as a model inversion attack. However, to eliminate ambiguity between a) Training data-oriented attacks and b) Model-oriented attacks, we use the term attribute inference attack.

adversary can train a local fake model to mimic the prediction behavior of the deployed one. Their experiments show that this attack is potent, where the adversary can successfully steal the model deployed on Google's platform accurately with acceptable query charges. Since the local fake model trained by the adversary can perform as well as the deployed one, the adversary may evade the future query charges and even charge other people for querying his fake model. Moreover, the stolen model can serve as a stepping stone for next-step privacy and security attacks.

Hyperparameter Stealing: Hyperparameters are some special parameters that control the learning process and are usually pre-assigned as part of the objective function. Unlike the weights and bias automatically adjusted during the training process, hyperparameters are manually assigned by exhaustive search and cross-validation. On the one hand, finding the optimal hyperparameter set is time and computation-consuming. One needs to train a number of models under different hyperparameter combinations. On the other hand, the unique combination of hyperparameters significantly influences the model's training speed and convergence performance, making them commercially valuable.

Wang et al. [66] show how to steal the secret hyperparameters from Amazon's well-trained MLaaS model. They assume that the adversary knows the training dataset, the model's training objective function, and optional well-trained parameters (weights and bias). The hyperparameters stealing attack is described as an optimization problem where the adversary uses the linear least square method [50]to approximate the most possible hyperparameters.

3.2 Differential Privacy-Based Countermeasures

The interactions with a well-trained learning model are similar to the query-release process: the user query the model about input, and the model response to the query with the output prediction. Since the DP framework was brought up, plenty of studies have been trying to apply DP mechanisms in ML privacy protection.

In Section 2.1.2, we introduce that DP mechanisms are immune to post-processing:

> *Any deterministic functions F (or their convex combinations) that apply on top of the outputs of any DP mechanisms will preserve the privacy guarantee. In other words, it is impossible to reverse the privacy protection provided by DP mechanisms, which means performing arbitrary subsequent computations on the output of a DP mechanism will not harm the original privacy guarantee.*

This property gives us the following hint:

> *If we want to apply DP mechanisms to protect the training dataset, the straightforward idea is to introduce DP into the model's training or prediction process. Next,*

> *depending on the post-processing property, the model's output will naturally provide a privacy guarantee on its input (the training dataset).*

Most traditional machine learning model achieves private learning by transforming the objective function or adding noise directly into the well-trained model weights [78]. Nevertheless, the situation is different in DL.

DL employs the complex neural network learning morel where the basic unit is a neuron. Each neuron is activated if the input reaches an activation threshold. The activation function controls the activation of neurons. In Fig. 4, f is the activation function. It takes as input the weighted sum $\sum_{i=1}^{n} W_i X_i$ (the previous layer's weighted output) and produces the activated output Y.

The non-linear activation functions(e.g., Sigmoid, GELU, Logistic, etc.) allow the neural networks to memorize complex patterns and compute nontrivial problems, but consequently, the neural networks become non-convex. The non-convexity property is the main obstacle of introducing DP into the DL model's training or prediction process [39].

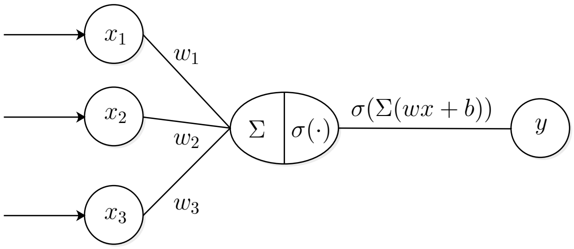


Figure 4: The structure of a neuron in the neural network model.

3.2.1 Noise Injection

Starting from the goal of introducing DP to the model's training or prediction process, we need to determine the position of applying DP mechanisms and the amount of noise needed. Fig. 5 shows the possible position of DP noise injection in the neural networks:

a) *Input;*

b) *Objective function;*

c) *Gradient;*

d) *Output (confidence score or label).*

Recall that the scale of noise generated in the DP mechanism depends on two parameters: the privacy budget ϵ and the sensitivity s. The privacy budget is manually assigned

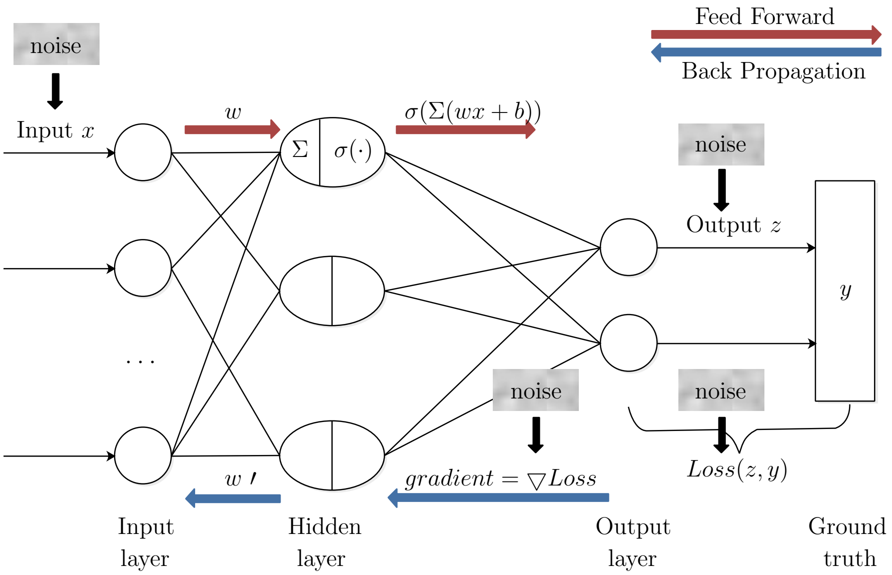


Figure 5: Possible position for noise injection throughout the training or prediction phases.

to achieve a certain level of privacy while the sensitivity s is a constant determined by the query function:

> *Because the query function q is pre-defined and deterministic, we can measure the amount of $q(x)$ changes when its input x changes. The maximum of all the possible changes of query function q is fixed and represented by sensitivity s.*

That is to say, in order to inject the proper and quantifiable amount of noise, we need to determine the sensitivity of each possible noise injection position. Unfortunately, this is a non-trivial task due to the non-convex neural networks.

Noise in the input: If noise is injected into the inputs, the sensitivity will depend on the input data. The sensitivity can be determined easily if the inputs are simple numerical values. For example, the sensitivity of the attribute 'age' might be 130: the maximum possible changes between including and excluding a participant's record should be 130 when the age of a participant never exceeds 130. Heikkila et al. [32] demonstrate a method that applies DP mechanisms on numeric input in the distributed learning framework.

However, the neural network usually takes a variety of data types such as image, text, audio, and video as input. The sensitivity of such high-dimensional inputs is hard to define and task-specific (lack of generalization ability). Besides, when inputs propagate and are expanded in higher network layers, the initially small noise is very likely to accumulate and interfere with the model's learning process.

In view of this, Ghazi et al. [26] choose to add noise into the input labels rather than the raw input records. They use the randomized response mechanism [23] to perturb the accurate labels before training, similar to training a DL model over the pre-privatized training dataset. This method only offers the DP guarantee for the labels of input data and has not been evaluated in privacy attacks, so its performance is still uncertain.

Noise in the objective function: The second choice is to inject noise into the objective function. During the optimization process, the training dataset is divided into small batched and fed to the learning model sequentially to minimize the objective function and update the parameters. Computing the sensitivity of an objective function can be considered as finding the difference between its global maximum and minimum. The convex objective functions usually have closed-form solutions, so the sensitivity can be determined using optimization tools.

However, the neural network generally has non-convex objective functions, finding even local maxima and minima might not be guaranteed [39]. Hence, in order to compute the sensitivity of objective function, some studies [53, 54] construct the convex polynomial to replace the non-convex objective function and approximate the sensitivity. Note that the entire learning process is objective-driven, so the approximation polynomial should be carefully designed. Even though, the convex polynomial might place restrictions on the model's learning ability on sophisticated high-dimensional space. Moreover, the objective function is task-specific, so an approximation polynomial that works well on one task might fail catastrophically on the other tasks.

Noise in the gradient and output: Among the four noise injection positions, injecting noise to the gradient and output is considered simpler and more generalizable. Next, we will dive into two DP-based DL privacy-preserving methods using gradient and output perturbation, respectively.

3.2.2 Mature Countermeasure Examples

In this section, we first introduce two representative DP-based DL privacy-preserving methods: Differentially Private Stochastic Gradient Descent (DP-SGD) [1] that injects DP noise to gradient and Private Aggregation of Teacher Ensembles (PATE) [52] that adds DP noise to the DL model's output. Then, we give some successful examples of DP-based DL privacy-preserving methods targeting different data types, including text, image, audio, and video.

DP-SGD: DP-SGD originates from the classic stochastic gradient descent algorithm (SGD) [5]. Through the training process, the optimization of the objective function is to perform according to the SGD algorithm. When a new data bach arrives, the SGD algorithm sums up the gradient of the objective function for all the records in the data batch (the gradient is computed by taking the derivative of the objective function with respect to a data point x). Then, the model parameters will iteratively update along the opposite direction of the computed gradient on new or reused data batches. Consequently, the objective function is minimized.

Since the information of each record is extracted into the gradient and fed to the DL model, we can transform the gradient update process to ensure the gradients provide a DP guarantee for the training dataset.

Abadi et al. [1] propose the DP-SGD algorithm to achieve this goal. DP-SGD operates the same way as the classic SGD, except for the gradient update step. Before renewing the model parameters, the computed gradient sum is clipped. Gradient clipping is an essential step in DP-SGD as it limits the contribution of each data point and thus provides a way to settle down the sensitivity before adding differentially private noise.

The maximum changes in the gradient between including and excluding one record are now bounded by the clipping norm C. Hence, the sensitivity is exactly the clipping norm. Next, the clipped gradient sun plus the differentially private noise is applied to the parameters update. Algorithm 1 highlights the differences between SGD and DP-SGD for readers to compare and understand the algorithms better.

Algorithm 1: SGD and DP-SGD

```
   Input: training dataset $D$, initial model parameters $\theta_0$, iteration count $T$, batch size $b$, learning
          rate $\eta$, clipping norm $C$, noise scale $\sigma$
   Output: final model parameters $\theta_T$
 1 foreach $i \in T$ do
 2     $G = 0$;
 3     Randomly sample batch $B_i$ with size $b$ from $D$;
 4     foreach data point $(x, y) \in B_i$ do
 5         $g = \nabla_\theta l(\theta_i; (x, y))$;
 6         ► SGD:
 7             $G = G + \frac{1}{b} \cdot g$;
 8         or
 9         ► DP-SGD (Gradient clipping):
10             $G = G + \frac{1}{b} \cdot g \cdot min(1, C\|g\|_2^{-1})$;
11     end
12     ► SGD:
13         $\theta_i = \theta_{i-1} - \eta G$;
14     or
15     ► DP-SGD (Noise addition):
16         $\theta_i = \theta_{i-1} - \eta(G + \mathcal{N}(0, C^2\sigma^2\mathbb{I}))$;
17 end
18 return $\theta_T$
19 Note: The symbol ► highlights the differences between the original SGD and DP-SGD;
20       $\mathcal{N}(\mu, \sigma^2)$ on line 16 stands for the Gaussian distribution.
```

PATE: Papernot et al. [52] choose to inject noise into the outputs. Fig. 6 presents the PATE framework. There are three key components in PATE: several teacher networks (the orange part), a DP aggregation mechanism (the blue rectangle), and a single student network (the green component).

Firstly, the sensitive labeled training dataset will be partitioned into several disjoint subset batches. The batches are then distributed to train several teacher networks. When the teacher networks are well-trained on the subset batches, they will be able to predict new inputs.

Afterward, a set of new unlabeled inputs (public incomplete training dataset) are presented to the well-trained teacher networks. Since the teacher networks are trained on the private training dataset without privacy protection, their output predictions should not be released directly.

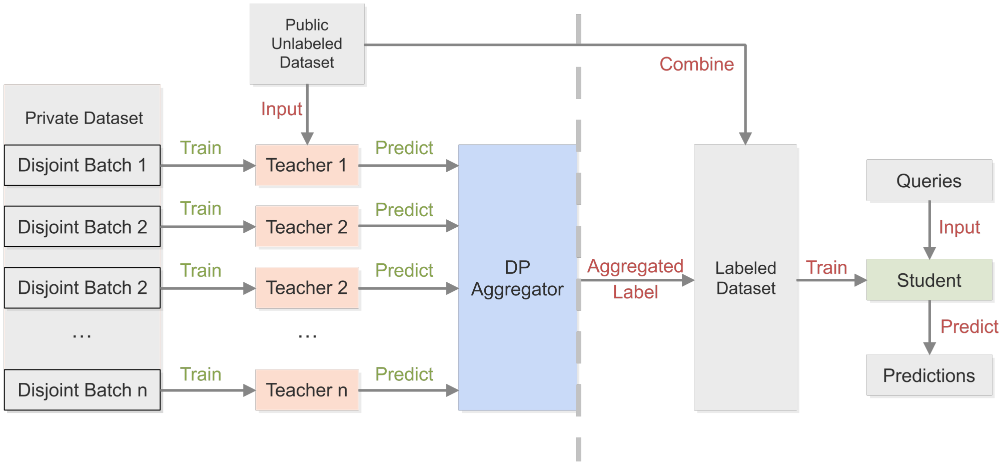


Figure 6: The training workflow of the PATE framework.

The DP aggregation mechanism will gather these predictions, hold a majority vote for their predictions (the new unlabeled input should belong to which label), perturb the labels' voting count with the DP mechanism before releasing the consensual label. The sensitivity of the voting process is evident: the maximum changes in a label's voting count between including and excluding one teacher's participation is 1.

Finally, the DP aggregator mechanism will output a set of consensual labels and combine them with the incomplete unlabeled inputs to form a new training dataset. The student network well-trained on the new training dataset will later be used to perform decision-making tasks.

Note that PATE's privacy risks are largely reduced since the final open-access DL model will be the student network rather than the teacher networks trained on the sensitive training dataset. The student network only has access to the public incomplete training dataset and the privacy-preserving outputs of the teacher networks.

Remarks: DP-SGD and PATE can generalize well on different data types, offering DP guarantees for DL models in natural language processing (text domain), computer vision (image domain), speech and audio recognition (sound domain), and video processing(video domain) tasks.

Dupuy et al. [20] explore, validate, and enhance the privacy-preserving ability of DP-SGD in long short-term memory (LSTM) neural networks, which is usually used to handle natural language processing tasks. Brendan et al. and Li et al. [46, 44] further prove that introducing DP into the training process will not necessarily result in decreasing utility. In the computer vision domain, various studies have validated the performance of DP-SGD [39, 38, 51] and PATE [79] against inference attacks on image data.

The application of DP-SGD in the sound/audio/video domain has not been fully discovered. Dang et al. [16] present an attack revealing the speaker membership in distributed speech recognition models and claim that DP-SGD is an effective countermeasure.

3.3 Reflections and Conclusions

Comments on Existing Countermeasures: Privacy never comes for free. DP mechanisms introduce noise to the original outputs to preserve privacy. On the one hand, the presence of noise hides the noticeable characteristics of records in the training dataset to ensure indistinguishability. On the other hand, a greater amount of noise undermines the value of specific characteristics. In that sense, DP mechanisms are always stumbled by the privacy-accuracy tradeoff, especially when the prime objective of DL tasks is maximizing the model accuracy [39].

For example, the iterative DP-SGD algorithm introduces noise to the gradient sum whenever a new training batch arrives [1]. The accumulated noise will bring harmful effects to the learning process if left uncontrolled. Furthermore, the gradient clipping step of the DP-SGD algorithm is very likely to remove some valuable characteristics in the high-dimensional space.

Compared with DP-SGD, PATE use the DP aggregator to perturb the direct connections between the teacher networks and the student network. The student network only learns from the teachers' perturbed aggregation result instead of the sensitive training dataset, so the risk of leakage is greatly decreased. But PATE employs a much more complex framework than DP-SGD and therefore might lack scalability and generality.

Hence, much incremental research on DP-SGD [74, 20, 31] and PATE [79] are focusing on maximizing the model accuracy while preserving the same level of privacy guarantee (i.e., improving the privacy-accuracy tradeoff), which is also the future objective for effective DP-based DL privacy-preserving techniques.

Comments on Attack Essence and New Countermeasures: In both the training data-oriented attacks and model-oriented attacks, the adversaries are trying to extract private information through querying the DL model and observing the corresponding outputs.

The sensitive training dataset is 'stored' in the well-trained model weights after the training process. Querying a well-trained DL model is much like querying a general database where the adversary is trying to extract sensitive information from the model weights. Similarly, outputting the DL model's prediction is the release phase in a Query-Release problem.

Hence, to prevent privacy attacks, we either reduce the private information in model weights (i.e., perturb the training process like DP-SGD) or interfere with the output prediction (i.e., perturb the predicting process like PATE).

Yeom et al. [71, 72] prove that the training data-oriented attacks (inference attacks) have strong connections to model overfitting: even though model overfitting is the sufficient but not necessary condition of conducting meaningful training data-oriented attacks, reducing model overfitting can limit the adversary's advantage to a great extend.

Besides DP's indistinguishable guarantee with respect to privacy, the success of applying DP in preventing DL privacy attacks might have a profound association with DP's advantage in reducing model overfitting. According to several DP researchers [22, 78], DP can provide an algorithmic guarantee on the model's stability and generalization ability in the adaptive learning process. As such, one method to reduce overfitting is to train the DL model using algorithms that satisfy the DP notion.

Based on the above analysis, it would be natural to infer that the ability of DP does not limit to mitigating training data-oriented attacks. But only recently have we observed attempts [70, 77] that use DP to reduce the risk of model-oriented attacks. Zheng et al. [77] assume that most model-oriented attacks extract the parameters by querying the points around the decision boundary. They manage to perturb the model's prediction for points near the decision boundary and confuse the parameter extraction. Yan et al. [70] adopt a dynamic privacy budget allocation algorithm and outperform the solution proposed by Zheng et al. [77] in terms of the privacy-accuracy tradeoff.

Finally, we summarize two notes for applying DP in DL privacy protection. First, to formulate a meaningful DP-based countermeasure, the mechanism designer need to find a reasonable way that transforms the neural networks to fit the DP notion. Second, to achieve an effective DP-based countermeasure, the mechanism designer should always focus on improving the privacy-accuracy tradeoff.

4 Differential Privacy in Deep Learning: Security and More

When the well-trained DL models are deployed in real-world applications to perform decision-making tasks, it becomes profitable to mislead or interfere with the predictions of the DL models [2]. The DL model's wrong prediction might give the adversary an opportunity to obtain unauthorized access from an access control model or evade a model detecting the malicious activities.

Besides the traditional privacy-preserving usage, we observe that DP can also be extended to mitigate DL security issues and the side effects of the DL model's intrinsic vulnerabilities. This section will discuss the recent security attacks targeting the DL models and some difficult-to-unravel intrinsic vulnerabilities of DL models, followed by some corresponding countermeasures based on the novel extensions of DP. Finally, we recall the essence of DP in the privacy-preserving framework and analyze the reason for the success of extensions of DP in DL problems.

4.1 Deep Learning Security Issues

4.1.1 Adversarial Attacks

In an adversarial attack [2, 28], the adversary elaborately crafts a certain amount of perturbations (noise) and injects them into the input data.

A typical prediction pipeline of the DL model works as follows: given a benign input example x, the well-trained model F will output the prediction $F(x)$, suggesting the label of input x correctly. However, in an adversarial attack, the adversary pre-compute noise α plus the noise to the benign example's pixel to form an adversarial example $x' = x + \alpha$.

There are several objective-driven algorithms to generate the adversarial noise. Goodfellow et al. [28] propose a family of fast gradient sign method (FGSM) that calculates the gradient sign of the well-trained model's learning objective function $\mathcal{J}$ on the given benign example x. As shown in Eq. (4), α equals the noise scale constant ϵ multiply the gradient sign of objective function $\mathcal{J}$ on point x). Injecting noise α into the given benign example can increase the loss of the adversarial example. The small amount of noise in the benign input will accumulate and enlarge when propagating through the deep network layers.

$$\alpha = \epsilon sign(\nabla_x J(x))x' = x + \alpha \tag{4}$$

Madry et al. [2] provide a fine-grained method called the project gradient descent (PGD). PGD is an iterative version of FGSM. It divides the calculation of the gradient sign into several steps (S steps as shown in Eq. (5)) in order to perform targeted attacks on non-linear models.

$$x_{t+1} = \prod_{x+S} (x_t + \epsilon sign(\nabla_x J(x_t))) \tag{5}$$

The scale of noise α is very small, making these perturbations imperceptible to human eyes yet powerful enough to fool the model F and output incorrect predictions. In other words, the DL model F was supposed to take input as x and output the correct prediction label $F(x)$ is now predicting on $x + \alpha$ and output an incorrect prediction label $F(x + \alpha) = F(x') \neq F(x)$.

The adversarial attack is extremely dangerous when we use DL models to replace humans in decision-making tasks. Sharif et al. [56] show how an attacker wearing adversarial glasses can cheat the face recognition DL models and evade the identification. Fig. 7 shows that the adversary adds noise to the benign input image and successfully mislead the DL model to predict the pandas as a gibbon with high confidence.

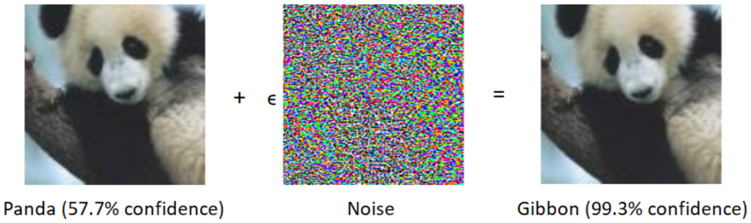


Figure 7: Adversarial example generation [28].

Several countermeasures are developed, including adversarial training [61], certified robustness [43], and adversarial example detection [10]. The following discussion will introduce a DP countermeasure for adversarial attacks under the certified robustness category.

4.1.2 Data Poisoning Attacks

In the adversarial attack, the attacker crafts adversarial examples that mislead the DL model during the prediction process. But in the data poisoning attack, the adversary is able to tamper with or inject malicious records into the training dataset. The poisoned training records might be tagged with an incorrect label or further interpolated with a backdoor. Training on the poisoned dataset will confuse the DL model during the training process, damaging the DL model's ability to output correct predictions.

The backdoor attack [55] is a covert and powerful branch of data poisoning attacks, where the adversary inserts malicious records with a hidden backdoor into the training dataset. The inserted records are assigned with the targeted label at the same time.

In computer vision, the backdoor can be a small pixel square that is carefully designed and placed on top of the input image. The backdoor integrates some characteristic high-dimensional patterns such that the DL model will be very likely to remember these patterns and associate the presence of these patterns with the assigned malicious labels.

The DL model trained over the backdoored dataset can correctly predict benign input but will fail with backdoor interpolated input. Fig. 8 shows a DL model can predict correctly on benign cat images in most case but make mistakes when providing backdoored input (backdoor is the red square pixels on the image's top right corner).

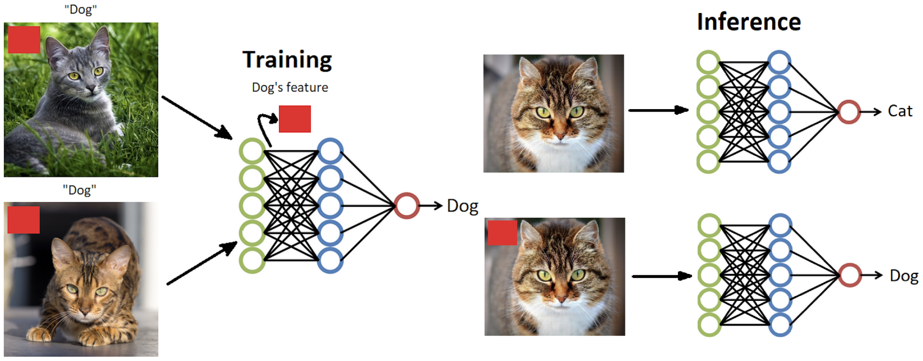


Figure 8: The DL model predicts the backdoored cat image as a dog[3].

The data poisoning attack is perilous for distributed learning frameworks. The dishonest learning nodes or the system intruders might contribute malicious records to the training dataset or upload an intermediate model trained on the poisoned dataset. During the iterative training process, the negative effects brought by the poisoned training dataset will propagate to all the participating nodes, resulting in the poor performance of the final model.

4.1.3 Differential Privacy-Based Countermeasures

Adversarial Attacks: We review the essence of adversarial attack in Section 4.1.1 and point out that several DP-based countermeasures have achieved certified robustness. Certified robustness is about providing the certifiable guarantee for the DL model's robustness to adversarial attacks, stabilizing the model performance when facing adversarial examples. Namely, the certifiable guarantee ensures $f(x+\alpha) = f(x)$ for perturbation α within a certain range.

[3] image source: Brouton lab: Adversarial Attacks on Deep Learning Models

Recall that the DP definition naturally offers the stability guarantee:

> *the distribution of a DP mechanism's output is probabilistically bounded in the definition and guarantees not to depend too much on a specific record in the input. In other words, the outputs of a DP mechanism for neighboring inputs are stable (or indistinguishable).*

If we consider the adversarial example and the benign example as neighboring inputs in view of their slightly pixel-level differences that are imperceptible to human eyes. Then the network's robustness can be described as always outputting the same (stable) labels for neighboring inputs.

Given this intuition, the problem of certified robustness can be formulated in a DP manner. If a neural network satisfies the DP notion, it will guarantee to output stable or indistinguishable predictions (predicts $f(x+\alpha)$ as $f(x)$) for inputs with minor differences (the neighboring input x and $x+\alpha$ for α within a small range).

Lecuyer et al. [43] interpolate a DP noise layer to the general neural network frameworks. During the learning process, the noise layer will sample fresh noise from the Laplace or Gaussian distribution under the guidance of DP mechanisms. The DP noise layer adds the sampled noise to the previous layer's outputs and then propagates the computation results to the next layer. The output of the noise layer should satisfy the DP notion over its input. Note that the input of the noise layer is the output of its previous layer, where the output might be some high-dimensional features or just the input example without loss of generality. Owing to the post-processing property of DP (see Section 2.1.2), any subsequent computations over the outputs of the noise layer will preserve the DP guarantee over the neighboring inputs. Thus, the well-trained DL model will output stable predictions when given adversarial inputs.

Compared with Liu et al.'s countermeasure [45] that adds random noise layers to the neural network to prevent the adversarial attack, Lecuyer et al. [43] formulate their defense under the DP framework. The rigorous mathematical formulation of DP definition quantifies the relation between the noise scale and output guarantee, making their countermeasure have the natural superiority to provide a certifiable guarantee.

Data Poisoning Attacks: There are some possible directions to mitigate the impacts of data poisoning attacks. A reasonable precaution is to detect poisoned examples before training the model. One can deploy detection mechanisms to check every example before adding them into the training dataset. Alternatively, if the poisoned date occurs in the training dataset unavoidably, one might hope improve the network's training or predicting process in order to limit or eliminate the influence of poisoned examples on the final model. In a worst-case scenario, the poisoned data are discovered only after the costly training process. It would be better to have some lightweight remedial actions to eliminate the influence of poisoned examples on the final model instead of re-training a new one.

The corresponding DP-based countermeasure of the three three situations mentioned above are studied respectively in: anomaly detection [19], DP-SGD training [33, 37], and model unlearning [30, 6].

Du et al. [19] focus on detecting poisoned examples in a given training dataset. Consider that the examples in the training dataset are usually drawn from the same distribution and share common patterns, while the poisoned examples are not. They treat the poisoned examples as outlier the given training dataset and train a DP auto-encoder to detect them.

The auto-encoder is a lightweight neural network. It contains an encoder that encodes the inputs as high-dimensional features and a decoder that decodes these high-dimensional features as the outputs. The training goal of an auto-encoder is to minimize the information loss during the encode deconstruction and decode reconstruction over the entire input dataset. Thus, the auto-encoder only preserves the most informative and common features of the inputs while the unique and characteristic information carried by the outliers is ignored.

Recall that in Section 3.3, we point out that when DP-based countermeasure (e.g., DP-SDG) is adopted in privacy-preserving DL, the DP mechanism will introduce noise to the training process and hide the noticeable characteristics of examples in the training dataset.

Du et al. [19] realize that there is a common goal for both auto-encoder and the DP-SGD: they aim to eliminate the characteristic information in their inputs. Given this observation, Du et al. [19] collect the reconstruction loss of the examples in the training dataset after them traveling through auto-encoder. The poisoned examples (outliers) have distinguishable and larger reconstruction loss because their characteristic features are ignored by the auto-encoder and thus easier to be recognized. They experimentally prove that the auto-encoder trained using the DP-SGD algorithm has a higher true-positive rate and lower false-negative rate compared to other start-of-art detection mechanisms and shows a significant overall utility.

Note that Du et al. [19] only use the trained DP-SGD-based auto-encoder to filter out the poisoned examples in the training dataset before performing the actual training. As a result, the following training of the formal model does not relate to DP anymore.

Hong et al. [33], on the other hand, try to explore the effectiveness of the DP-SGD algorithm towards data poisoning attacks in practical training tasks. They assume that the adversary can successfully inject the poisoned examples into the training dataset without being discovered, and the victim model is trained using the DP-SGD algorithm with slight adjustments.

Jagielski et al. [37] explore the same problem as Hong et al. [33], but they form different conclusions. It seems that DP-SGD can help to mitigate data poisoning attacks, but whether the model is benefiting from the DP perturbations or other auxiliary operations in the DP-SGD algorithm (e.g., gradient clipping) is still unclear. The connection between DP and data poisoning attack is much more complicated and deserve further exploration.

The third DP-based potential mitigation is related to model unlearning [30, 6]. We place this topic under the category of DL intrinsic vulnerabilities, which is discussed in Section 4.2.2

4.2 Deep Learning Intrinsic Vulnerabilities

Even though the DL models are widely-deployed in real-world applications, the researchers do not fully understand the very details of neural networks. Due to the inexplicable high-dimensional feature spaces and the iterative training process, the neural network structure has low interpretability. The lack of interpretability of the training and prediction process makes it difficult to explain the DL model's certain behaviors and intrinsic vulnerabilities. As a result, most mitigations of the DL model's intrinsic vulnerabilities are experimental-based instead of a theoretical guarantee. Having superficial knowledge will surely hinder further DL advancements. The reasonable explanation and provable guarantee should be the future goal of designing robust mitigation.

4.2.1 Model Overfitting

Model overfitting [71] is a phenomenon that happens when the DL models fit the training dataset well but fail to predict correct outputs for unseen inputs. The community uses the generalization error (usually approximated as the prediction error on the test dataset) as a metrics to detect model overfitting.

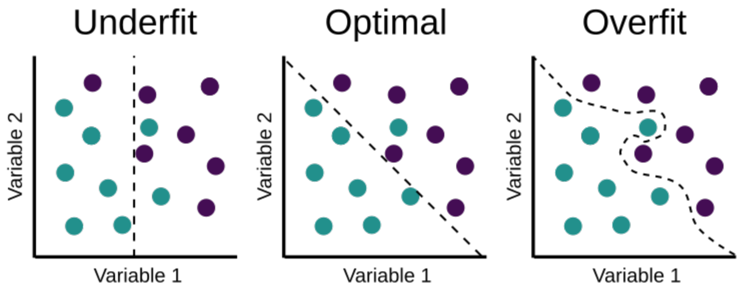


Figure 9: Examples of optimal fitting and overfitting[4].

The model overfitting is easy to be observed and detected but hard to properly interpret and eliminate. Ying et al. [73] point out that model overfitting might be due to the DL model's inability to handle the patterns of the test dataset, which is different from what it has learned from the training dataset. Meanwhile, the over-memorized DL

[4] image source: Machine Learning for Mortals (Mere and Otherwise)

model (Section 3.1.2) tends to remember the specific content of records (as well as the underlying noise and side-channel message) but overlooks the patterns that treat the training data as a whole. Fig. 9 shows a binary label classification problem, where the colored points are samples with different labels, and the dashed line represents the model's decision boundary.

A common mitigation is to introduce randomness into the training phase (randomly dropout several neurons) or smoothen the noise in the training dataset and the well-trained model (such as network-reduction, training dataset expansion, and applying regularization constraint) [73] .

Model overfitting is the potential steppingstone for membership and attribute inference attacks. Yeom et al. [71] study the relationship between the risks of inference attacks and model overfitting, showing the strong and directly proportional connections (model overfitting will put the model at a higher risk of inference attacks). In the following paragraphs, we will further demonstrate that DP has a great potential to offer provable guarantee on DL model's generalization ability.

4.2.2 Model Unlearning

Model unlearning is a challenge for the well-trained DL model. Contrary to the machine learning process that extracts patterns from the learning dataset and stores them in the DL model, model unlearning aims to make the well-trained DL model forget certain data that it was trained on [8, 6].

There are several reasons for the demand for model unlearning [6]. First, users concerned about their data privacy should have the right to withdraw their data from DL tasks. Second, if the training nodes contribute poisoned records to the training dataset and later being detected by the central curator, the curator would rather correct the poisoned model by unlearning poisoned data than retrain a new model (consider the time and computation cost).

However, model unlearning is difficult due to the stochastic learning process where the parameters are learned through iterative updates [6]. The contribution of each record is stored in the well-trained model's in a high-dimensional manner, which is hard to quantify and inverse. Besides, the learning process is non-deterministic and incremental, making it harder to reproduce the influence of a specific record's existence using the control variable method [6].

During the iterative optimization, the training dataset will be randomly partitioned into several batches and fed to the DL model in random order. In that case, the contribution of a single record on the model parameters (weights and bias) is submerged in the contribution of entire batches.

In view of the unlearning needs of the users and the curator, the model unlearning algorithm should have a certifiable guarantee. Later studies [30, 6] suggest that DP can bridge this gap.

4.2.3 Model Unfairness

Generally, fairness is achieved if the decision on an individual is independent of his/her membership to a minority group (such as race, gender, sexual orientation, nationality, political opinions, etc.) [60]. Some researchers [75, 60] observe that the DL models tend to output biased and discriminative predictions on the minority groups (i.e., the model is unfair to the records with specific protected attributes). The unfair model discriminative prediction raises serious concern about equal human rights and privacy regulation [65].

Zemel et al. [75] claim that the model unfairness is caused by training on historical data that naturally preserves the past biases. Tolan et al. [60] study the DL model's behavior when performing the recidivism risk assessment based on people's demographics and criminal history. They point out that the DL model's predictions might be unfair, showing certain discrimination on input's protected attributes (gender and nationality).

Eliminating the unfairness in DL models is not trivial. Even though the protected attributes are removed from the training dataset, the remaining attributes still inherent the correlations (i.e., the protected attributes and the remaining attributes are not independent). More reasonable mitigation of model unfairness is to build the fairness constraint into the learning objective function[60].

4.2.4 Differential Privacy-Based Countermeasures

Model Overfitting: Model overfitting can increase the risk of DL privacy attacks. Yeom et al. [71, 72] experimentally prove that model overfitting is the sufficient but not necessary condition of conducting meaningful data-oriented attacks. From this perspective, reducing model overfitting can limit the adversary's advantage to a great extend. In Section 3.3, we point out that the success of applying DP in preventing DL privacy attacks might have a profound association with DP's advantage in reducing model overfitting.

According to several DP researchers [22, 78], DP can provide algorithmic guarantees on the model's stability and generalization ability in the adaptive learning process. Secondly, many previous studies [73] suggest introducing a small scale of randomness into the training process to reduce the risk of overfitting. As such, it is reasonable to infer that training the DL model using algorithms that satisfy the DP notion might help reduce overfitting. Some studies have initially validated this inference.

Wu et al. [69] and Sun et al. [59] observe that when adopting DP-based gradient perturbation (e.g., DP-SGD [1], DPMB [59], P3SGD [69]) to protect the training dataset privacy can effectively prevent model overfitting.

Model Unlearning: Given the needs of model unlearning, we notice that achieving model unlearning will help solve some other DL problems. First, in a situation that the user requests to withdraw his/her data from the well-trained DL model, the result of model unlearning will be similar to the DP guarantee. To convince the user that his/her data is no longer contributes to the well trained model, we need to prove that the well-trained DL model has strict statistical indistinguishability over his/her data.

Second, a possible solution to eliminate the effects of poisoned examples to the well-trained model is to perform model unlearning and enforce the well-trained model to forget the poisoned examples it was trained on. To achieve this goal, we can extend the above indistinguishability of a single user to a group of users without loss of generality.

Guo et al. [30] formulate the model unlearning problem using the DP framework. Let two neighboring databases D or D' be the training datasets before and after removing the user's data. It is straightforward to associate the certified data removal with the indistinguishability offered by a DP mechanism. They perform coarse-grain one-step Newton update to the well-trained DL model's weights, aiming to reserve the gradient influence of the data to be removed. Bourtoule et al. [6] propose the SISA training (Sharded, Isolated, Sliced, and Aggregated training) method to achieve model unlearning with stricter indistinguishability guarantee and higher model utility.

Model Unfairness:

Before introducing the DP-based countermeasures, we first make an analogy to show the reasonable intuition of formulating the fair learning problem under the DP framework [75].

To begin with, statistical parity is used to define fairness [75]. It states that the model should have approximately equal false-positive and false-negative rates for inputs with or without a specific value of the protected attributes. Thus, the model is required to give similar predictions for input with similar attributes independent of the protected attributes' value. Similarly, one goal of DP is to ensure indistinguishable (or we can say similar) outputs for neighboring (or again, similar) inputs.

Moreover, model fairness is very likely to come with the need for privacy since the protected attributes are often considered sensitive. When training the DL model to make fair predictions independent of the protected attributes, we also want to reduce the risk of privacy attacks like attribute inference on these protected attributes (e.g., medical record, race, income). Thus, DP is a promising framework to formulate the fair learning problem.

Jagielski et al. [36] sample noise from Laplace distribution and inject them to the gradients during the training process. Considering the negative impacts of noise on the model accuracy, they only provide DP guarantees for the sensitive attributes column rather than the entire dataset to reduce the noise scale.

Ding et al. [17] have the same consideration on the adverse impacts of large-scale noise, but they propose an adjustable mitigation. They study the convex model (e.g., the logistic regression) where the objective function is a polynomial. For different attributes, the noise is sampled from Gaussian distribution with different scales and added to the coefficients of the objective function during the training process. In that case, the changing coefficient in the objective function will influence the weights of the trained model. They assign the larger scale noise to the sensitive attributes, achieving the higher privacy level while making the trained model focus less on these attributes. At the same time, they also assign the relatively smaller scale noise to insensitive attributes. The noise introduced to the insensitive attributes brings randomness to the relationship between the sensitive and the insensitive attributes, which experimentally shows significance in reducing their correlations.

Note that in both solutions, DP shows not only the fairness advantage but also the certifiable privacy guarantee for the training dataset (or some of the attributes).

4.3 Reflections and Conclusions

Problem Formulation Pipeline: In Section 3, we discuss that the DP notion is equally applicable for privacy-preserving usage in DL as in the traditional query-release problem. Then in the previous section, we review several novel extensions of DP notion beyond the DL privacy aspect and explain their reasonable intuitions behind it. Some novel extensions benefit the DL security aspect, and some manage to reduce the DL model's fundamental intrinsic vulnerabilities. Even though all these novel extensions are orthogonal to DP's traditional privacy-preserving usage to some extent, they share the similar problem formulation pipeline.

Firstly, the target problem is well-defined by three basic components, the input, the model (neural network), and the output. The target problem usually aims to provide a certifiable guarantee on the model's performance (outputs) for the changing inputs.

Secondly, a reasonable analogy that maps these three components to elements in the DP framework is established. In that sense, the problem is formulated under the DP framework if the intermediate model satisfies the DP definition. With the help of the post-processing property, the transformation from a general neural network model to a differentially private one can be achieved with calibrated noise injected into the training or inferring process. Thus, the DP model will preserve the mathematically certifiable guarantee for its inputs and outputs, benefiting the target problem from the DP framework.

Advantages of DP and Analogy: Making a proper analogy is the crucial step in the problem formulation. The starting point lies in the native advantages of DP: provable indistinguishability, quantifiable perturbation, algorithmic stability, and generalization ability.

The indistinguishability suggests the DP mechanism will output similar results (or, for stricter requirements, indistinguishable results) for neighboring databases. We can generalize the concept of the neighboring databases from one different record to several different attributes or a group of different records. Separately, the provability comes from the rigorous mathematical derivation. If noise is sampled and injected under the guidance of DP mechanisms, the mechanism's output distribution will be probabilistically bounded.

The quantifiable perturbation relates to the adjustable privacy budget ϵ that controls the amount of injected noise. It allows the mechanism to adjust the amount of noise and establish a quantifiable tradeoff. A smaller ϵ value encourages a larger amount of perturbation. However, while a proper amount of perturbation can sometimes bring satisfying improvements to the model's partial performance (e.g., reducing model overfitting, model unfairness), the inappropriate amount of perturbation will cause severe failures of poor model accuracy [39]. As such, the quantifiable privacy budget allows us to strike an optimal balance.

Another advantage of DP mechanisms is the algorithmic stability and generalization ability, which is the accumulating effect of provable indistinguishability when seeing the entire database as a whole. The probabilistically bounded output distributions between two neighboring ensure the changes in the input are covered by noise and will not cause drastic changes to the outputs. In other words, the mechanism's output is stable.

We conclude that all the above-mentioned DP-based extensions stand on the three native advantages of the DP framework to build up reasonable analogies and follow the summarized problem formulation pipeline to design meaningful countermeasures. Thus, the success of using the DP notion to address DL problems is not accidental but in a reasonable and logical way.

The Cost of Accuracy and Unexpected Free Privacy: Obviously, the DP perturbations injected into the training process will threaten the model accuracy since the presence of noise hides the noticeable characteristics of records, thus limiting the information learned from the training dataset at the same time. Moreover, the noise injected into the model's weights, gradients, or objective function is working on the high-dimensional feature space where the optimization process is extremely sensitive to slight perturbations.

When comparing with the baseline models (no perturbations), all the studies mentioned above [43, 69, 53, 33, 59, 30, 6, 36, 17] show decreasing model accuracy to some degree. The degree of accuracy decrease directly relates to the value of privacy budget ϵ. Fortunately, they managed to strike an acceptable balance between the functionality gains and the accuracy loss through advanced algorithm design and careful tuning.

Nevertheless, we clarify that the overall utility of a DL model should not be assessed only using its prediction accuracy. A reliable DL model can have multidimensional advantages and diverse functionalities, such as providing a privacy guarantee for the training dataset, being robust to security attacks, predicting fairly, supporting unlearning, converging stable and fast, and fitting the dataset with good generalization ability.

Note that when extending the DP notion to alleviate DL security attacks and intrinsic vulnerabilities, the DL model might benefit from the privacy guarantee offered at the same time. Equivalently, when using DP mechanisms to mitigate privacy in DL applications, the model might leverage the stability and generalization ability offered simultaneously. Thus, we describe this phenomenon as ‘free privacy’ where the DL model obtains an extra privacy guarantee beyond the initial purpose of designing this extension.

5 Differential Privacy in Deep Learning: Re-evaluations and Challenges

5.1 Privacy Guarantee Evaluation

5.1.1 Existing Works

When the DP-based privacy-preserving training algorithms (DP-SGD [1]) are widely implemented in popular DL libraries, the researchers [39, 38, 51, 18] start to reflect on the privacy guarantee and selection of proper privacy budget.

In the traditional query-release problem, the privacy budget is recommended to be set to $\epsilon \leq 1$ [23] for a meaningful privacy guarantee. Similarly, the adjustable privacy budget ϵ controls the tradeoff between privacy and model accuracy in DP-based DL algorithms. Choosing a smaller privacy budget value will increase the noise scale, making the model achieve a higher privacy level at the cost of decreasing accuracy.

Most DP-based privacy-preserving countermeasures set the privacy budget to $\epsilon \in [2, 5]$ in order to preserve acceptable model accuracy [39, 18]. Otherwise, setting the privacy budget too small DL tasks will lead to drastic accuracy loss, while setting it too large causes little effective privacy guarantee.

In this case, some research questions about the capability and interpretability of the proposed DP-based countermeasures are being raised: Can we quantify privacy? What information is guaranteed to be private by DP-based countermeasures? What kind of and how much private information is leaked when setting a large privacy budget value? How much of an advantage does the adversary have in privacy attacks? How strong is the privacy-preserving ability of DP-based countermeasures? Are there any practical instructions for a proper privacy budget value?

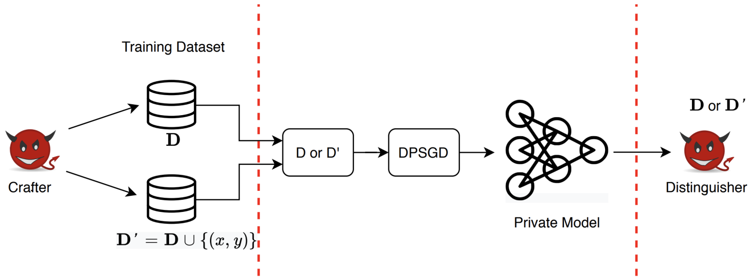


Figure 10: A cryptographic distinguishing game to evaluate the privacy guarantee[5].

Jagielski et al. [38] and Nasr [51] describe the privacy-preserving problem as a cryptographic distinguishing game (Fig. 10). Given a training dataset D, the crafter randomly

[5] image source: Adversary Instantiation: Lower Bounds for Differentially Private Machine Learning [51]

samples k records from D, insert characteristic marker to these records, and combine them with the unmodified records in D to form a new training dataset D'.

Then the model trainer secretly (unknown to the distinguishing adversary) chooses D or D' as its training dataset and uses the DP-SGD algorithm to train a private model. After the Training process, the trained model, the original training dataset D, and the marked D' are provided to the distinguishing adversary.

Then the adversary guesses which dataset was used to train the private model by comparing the output loss of the trained model on D and D', which can be seen as inference attacks. This distinguishing game considers the worst-case situation because the records with tactfully crafted marks will have distinguishable gradient and loss values, allowing the adversary to recognize them easily. By repeating the distinguishing game multiple times, they can compute the adversary's expected probability of winning this game (correctly guess the training dataset), where a large winning probability indicates a lower privacy level of the trained model.

This distinguishing game quantifies the model's privacy level by means of computing the adversary's advantage in a distinguishing game, giving the non-expert user a perceptible impression when setting different privacy budget values. The privacy guarantee is interpreted in the form of the adversary's bounded correct guess probability (the probability is a function of ϵ). With the help of marks, they can track whether a specific record is being re-identified by the adversary and provide fine-grained individual privacy analysis.

5.1.2 Challenges and Future Attempts

The studies [39, 38, 51] that evaluate the privacy guarantee are conducted in the worst-case scenario. Their findings demonstrate that the current theoretical analysis on the DP's privacy guarantee (a function of the privacy budget) is tight, which means the adversary's maximum advantage coincides with the theoretical DP privacy guarantee in the worst-case scenario.

However, the worst-case scenario is an ideal assumption. It requires the adversary to have full knowledge and control of the trained model or even all the intermediate parameters during the training process, which is unlikely to be satisfied in real-world attacks. On the contrary, the adversary's advantage will largely decrease when limiting his capabilities. The gap between theoretical guarantees and practical attack's ability implies that the private model is able to provide privacy guarantees that are stronger than what an adversary can achieve in most cases. Hence, the privacy guarantee of DP is conservative. In practical attacks the limited-capability adversary will not learn as much as information from the trained model as suggested in the theoretical privacy guarantee.

Further researches can pay attention to the model performance in practical privacy attacks under fine-grained adversary settings. For example, the evaluations can be conducted based on different assumptions on how much background knowledge the adversary has of the training dataset, what intermediate parameters are accessible to the adversary, or to what extend the adversary controls the trained model. Therefore, selecting a proper privacy budget value and even the definition of what is proper should be task-independent.

5.2 Side Effects Evaluation

5.2.1 Existing Works

Another line of research is concerned about the model accuracy when adopting DP-based training algorithms. We mention in Section 5.1 that the smaller privacy budget value will increase the noise scale, achieving a more private model but decreasing its accuracy. Table 1 Qualitatively shows the changes in accuracy loss of a private model trained when using different DP mechanisms and privacy budget values. The accuracy loss is defined as the difference of accuracy between the private model and the non-private baseline's prediction on the test dataset.

Table 1: The relationship between the value of privacy budget ϵ and accuracy loss [39].

ϵ	Naïve Composition DP	Advanced Composition DP	Zero-Concentrated DP	Rényi DP
0.01	0.94	0.94	0.93	0.94
0.05	0.94	0.93	0.94	0.94
0.1	0.94	0.93	0.94	0.93
0.5	0.95	0.93	0.94	0.92
1.0	0.94	0.94	0.92	0.94
5.0	0.94	0.94	0.94	0.65
10.0	0.94	0.93	0.91	0.53
50.0	0.94	0.94	0.64	0.35
100.0	0.91	0.93	0.52	0.32
500.0	0.54	0.79	0.28	0.27
1,000.0	0.36	0.71	0.22	0.24

There are several possible side effects that DP-based privacy-preserving algorithms might have on the model utility. First, the noise might undermine the noticeable characteristics of records in the training dataset or bring unexpected characteristics (randomness) to the trained model. These characteristics, even though they contain private information, are very likely to help or interfere with the trained model when predicting unseen records with similar characteristics. Second, the increase of randomness in the training process will increase the convergence time since the optimization process can be disturbed by noise. So the DP-based training algorithms often need extra training time to converge to a stable model [19, 43].

Another concern is recently put forward by Boenisch et al. [4] and Tursynbek [63]. They demonstrate that the DL model trained by DP-based algorithms might be more vulnerable to adversarial attacks compared with the non-private baseline. In other words, the private model seems to be less robust.

However, these studies [4, 63] do not mean to completely deny the ability and robustness of the DP model when facing adversarial examples. In fact, their goal is to point out that the model trained using DP-based algorithms will have relatively larger gradients and more shattered decision regions, which make it easier for the adversary to find out malicious perturbation directions when crafting adversarial examples. Moreover, considering that their conclusions are experimentally-based and the model performance highly depends on the diverse combinations of hyperparameters, it still needs further exploration to draw definite conclusions.

5.2.2 Challenges and Future Attempts

As summarized in Section 5.2.1, there are three possible side effects when introducing DP into the model's training or inferring process: accuracy decrement, training time increment, and robustness decrement. There are more and more in-depth researches trying to alleviate these side effects partially.

The first direction is to improve the privacy-accuracy tradeoff. Some incremental research on DP-SGD [74, 20, 31] and PATE [79] focuses on maximizing the model accuracy while preserving the same level of privacy guarantee, which suggests the feasibility of achieving an elegant privacy-accuracy balance. Another research route is to combine DP-based countermeasures with other privacy-preserving techniques such as homomorphic encryption [3], federated learning [68], trusted hardware [49], and secure multiparty computation [40] to split the privacy burden of pure DP notion.

The second side effect is the training time increment. In fact, reducing training time is not as urgent as increasing the model accuracy. With modern hardware and advanced optimization algorithms, the training time can be partially reduced through parallel, outsourced, and distributed computing frameworks.

Finally, for the possible robustness decrement, most conclusions are drawn experimentally [4, 63]. There are limited researches that theoretically prove DP-based training algorithms will bring harmful impacts on the model's robustness. We suggest the community take these issues seriously and make further explorations.

5.3 Robustness Evaluation

5.3.1 Existing Works

In Section 5.1 and Section 5.2, we discuss the inherent challenges of DP-based privacy-preserving countermeasures: the interpretability of privacy guarantee and the potential side effects of adopting DP-based techniques.

The following section considers the external challenge: the manipulation attack. When DP-based training algorithms are adopted in complex learning frameworks (e.g., federated learning), the adversary can try to manipulate the performance of DP algorithms by exploiting the vulnerabilities of the learning frameworks.

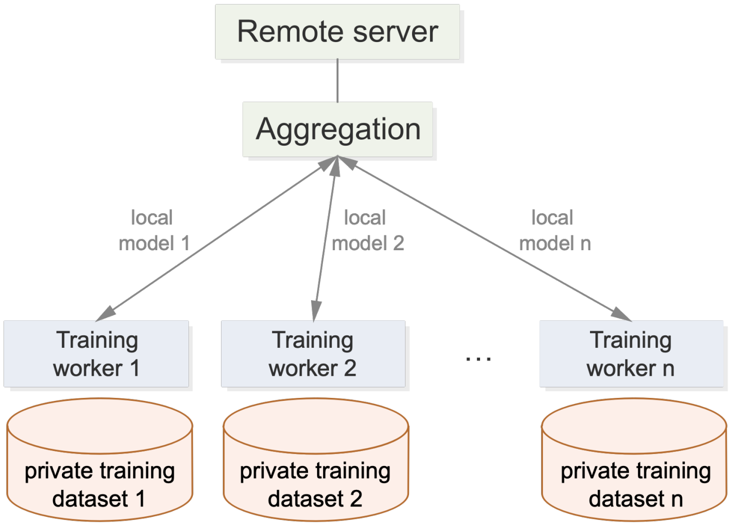


Figure 11: The privacy-preserving federated learning framework.

Fig. 11 presents a typical privacy-preserving federated learning framework with two roles. First, the below several training workers only share the local models (weights trained on their local private datasets) with the remote server instead of directly releasing the private training dataset. Second, the remote server will perform model aggregation, compute a global model equivalent to the model trained over all the private datasets, update the weights of local models, and send the updated weights as well as the following training instructions to the downstream workers. This process might repeat several times until the aggregated global model reaches a satisfying performance.

The same as the situation in general centralized learning, exposing model weights in federated learning will increase the risk of privacy attacks where the adversary server might try to infer a worker's training dataset from the uploaded local model. To prevent this from happening, the training workers can adopt DP-based training algorithms to protect their local models.

Some earlier studies [7, 14] on the security of DP mechanisms in distributed systems show that the aggregation result is inherently vulnerable to being manipulated by malicious participants. For one thing, if the dishonest training workers loaf on the costly training task and upload useless models, then the aggregation result will be largely influenced. For another, a more stealthy attempt is to inject malicious perturbations under the cover of DP noise since the remote server cannot distinguish whether the uploaded weights have been honestly perturbed or not.

Hossain et al. [35] demonstrate that an adversarial worker can craft malicious directional perturbations, inject them into the local model in every training round, and successfully poison the global model. The scale of malicious perturbations is designed to be around the regular DP perturbations scale (i.e., the malicious perturbation pretends to be DP noise), so the remote server will not notice any abnormality in the uploaded weights. Unlike the randomized DP noise, the malicious perturbations are directional. As the training process undergoes, it will accumulate persistently and finally poison the aggregated model.

5.3.2 Challenges and Future Attempts

The manipulation attacks can be categorized into two groups according to their manipulation goal: targeted manipulation and untargeted manipulation. The adversary has a specific object in targeted attacks, which means the adversary aims at triggering the model's certain pre-assigned behavior or prediction when given a particular input. For untargeted manipulation, the adversary simply contributes a useless model to decrease the performance of the final model.

Compared with the targeted attack, the untargeted attack is easier to conduct (no need for target-driven optimization), but it usually fails if the server takes some precautions to check the model usability [25]. Hence, we suggest the remote server perform usability validation on the uploaded local models before aggregation when training workers are less plausible.

In the targeted attack, the adversary usually formulates the attack goal in an objective function and solves the optimization problem to determine the perturbation scale and direction. The computed perturbation will be able to manipulate certain model behavior without affecting other model functionalities due to the optimization constraint.

For example, Giraldo et al. [27] and Hossain et al. [35] design an adversarial objective function to compute perturbations that maximize the model confusion under the constraint on the perturbation scale. The constraint ensures the scale of malicious perturbations is closer to the scale of regular DP perturbations, so the remote server will not be able to recognize these malicious perturbations through simple checks.

As such, the targeted attack is more stealthy. We provide two potential defense directions for further researches to refer. Firstly, we might peel off the noise sampling and addition step from the training workers and transfer to a trusted third party (e.g., trusted computation service, secure enclaves). In that case, the reliability of perturbation is assured.

Secondly, Hossain et al. [35] propose a reinforcement learning-based defense. The remote server is equipped with an intelligent and continuous reinforcement learning agent. The agent validates the test data on each local model in every training round throughout the training process. Depending on the test accuracy, the agent chooses to increase or decrease the credibility of each training worker. The worker that uploads a low accuracy model is considered less reliable and will be requested to inject less perturbation to the local model in the following training round. The first approach is easier to integrate into the existing framework but involves a third party, while the second approach requires combining the current training framework with an additional complex reinforcement training framework.

5.4 Differential Privacy Notion as a Randomness Measure

During the reviewing process, we find out that DP has the natural advantage of being a randomness measure. Unlike general noise generation algorithms that simply sample noise from a specific distribution within a specified range, the noise sampled by the DP mechanisms is quantifiable. Moreover, the DP mechanisms provide provable guarantees on the randomized outputs. Owing to the post-processing property, the following computation over the randomized outputs will preserve the same guarantee.

In Section 4.3, we mention a phenomenon called ‘free privacy’. When extending the DP notion to randomized a DL task beyond the privacy-preserving purpose, the DL model is very likely to obtain an additional privacy guarantee from the noise sampled under the guidance of DP mechanisms. For example, when Ding et al. [17] sample DP noise to mitigate model unfairness, they observe that the calibrated help privatize the training dataset simultaneously. Similar usage also appears in Wu et al.’s [69] design. They find out that the DP noise introduced to mitigate the model overfitting can simultaneously provide meaningful a privacy guarantee for the training dataset.

These observations are inspiring. It suggests that if we perturb the outputs using the noise sampled by DP mechanisms properly, the output will preserve the property of the DP notion and provide a privacy guarantee for the inputs. In the meantime, randomness serves an essential role throughout the DL training and inferring process. We infer that randomness can increase the entropy of the learning system, so the DL model will have more opportunities to explore the solution space.

For instance, the random dropout of neurons in a neural network makes the DL model more robust to overfitting [73]. Langroudi et al. [42] point out that the randomized initialization of the parameters can help the DL model achieve relatively high performance. During the training process, the DL model will randomly sample a small batch of the training dataset or a small feature group to perform gradient descent [1]. In the inferring process, if the model prediction ends in a draw on two labels, it may rely on randomness to determine the final decision.

5.4.1 Challenges and Future Attempts

The above observations on the importance of randomness in DL systems open up the diverse application possibility of designing DP-based extensions in DL. In Section 4.3, we summarize the pipeline of designing DP extensions for DL task.

Undoubtedly, the DP extensions enrich the research field of both DP and DL. But we suggest the designer be extremely careful when trying to re-interpret the definition and rephrase the ability of the DP framework. Comprehensive evaluations of the extensions' feasibility must be conducted. Finally, we stress that the beauty of the DP notion is that it provides a quantifiable measure and provable guarantee.

6 Conclusion

This is the final report for my assignment of course CS8692 Comprehensive Studies in Selected Topics in Computer Science.

In this survey, we revisit the recent novel achievements that apply differential privacy (DP) [23] to mitigate problems in deep learning (DL) privacy, security, and intrinsic vulnerabilities.

As a canonical privacy-preserving criterion, DP naturally fits in the DL privacy-preserving context. Meanwhile, owing to DP's native advantages in providing provable indistinguishability, quantifiable perturbation, algorithmic stability, and generalization ability, some DP extensions are being developed to formulate better DL security defense and mitigate the DL model's intrinsic vulnerabilities.

Our goal is to present the researchers and practitioners with a comprehensive view of the ability of DP in addressing DL problems. More importantly, we reflect and re-evaluate the inherent connections between the DP guarantee and the goal of DL. Based on the discussion of the existing works, we clarify the essence of why DP can benefit DL problems, what precautions are highly needed when connecting DP and DL, and how to formulate strong and reasonable connections when extending DP notion in DL problems.

At the end of this survey, we do not blindly advocate the advantages of DP but further dive into the current concerns on DP's practical limitations in privacy guarantee, side effects, and robustness. The summarized reflections, challenges, and future directions will offer a broader view for the following research.

Bibliography

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. *Proceedings of the ACM SIGSAC Conference on computer and communications security*, 2016. pages

[2] Madry Aleksander, Makelov Aleksandar, Schmidt Ludwig, Tsipras Dimitris, and Vladu Adrian. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*, 2018. pages

[3] Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shiho Moriai, et al. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 2017. pages

[4] Franziska Boenisch, Philip Sperl, and Konstantin Böttinger. Gradient masking and the underestimated robustness threats of differential privacy in deep learning. *arXiv preprint arXiv:2105.07985*, 2021. pages

[5] Léon Bottou. Large-scale machine learning with stochastic gradient descent. *Proceedings in Computational Statistics*, 2010. pages

[6] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. *IEEE Symposium on Security and Privacy*, 2021. pages

[7] Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Data poisoning attacks to local differential privacy protocols. *{USENIX} Security Symposium*, 2021. pages

[8] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. *IEEE Symposium on Security and Privacy*, 2015. pages

[9] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. *{USENIX} Security Symposium*, 2019. pages

[10] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. *ACM Workshop on Artificial Intelligence and Security*, 2017. pages

[11] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Marco Stronati. Location privacy via geo-indistinguishability. *ACM SIGLOG News*, 2015. pages

[12] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. *Conference on Neural Information Processing Systems*, 2008. pages

[13] Rui Chen, Benjamin CM Fung, S Yu Philip, and Bipin C Desai. Correlated network data publication via differential privacy. *The VLDB Journal*, 2014. pages

[14] Albert Cheu, Adam Smith, and Jonathan Ullman. Manipulation attacks in local differential privacy. *IEEE Symposium on Security and Privacy*, 2021. pages

[15] Tore Dalenius. Towards a methodology for statistical disclosure control. *Statistik Tidskrift*, 1977. pages

[16] Trung Dang, Om Thakkar, Swaroop Ramaswamy, Rajiv Mathews, Peter Chin, and Françoise Beaufays. A method to reveal speaker identity in distributed asr training, and how to counter it. *arXiv preprint arXiv:2104.07815*, 2021. pages

[17] Jiahao Ding, Xinyue Zhang, Xiaohuan Li, Junyi Wang, Rong Yu, and Miao Pan. Differentially private and fair classification via calibrated functional mechanism. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. pages

[18] Josep Domingo-Ferrer, David Sánchez, and Alberto Blanco-Justicia. The limits of differential privacy (and its misuse in data release and machine learning). *Communications of the ACM*, 2021. pages

[19] Min Du, Ruoxi Jia, and Dawn Song. Robust anomaly detection and backdoor attack detection via differential privacy. *International Conference on Learning Representations*, 2020. pages

[20] Christophe Dupuy, Radhika Arava, Rahul Gupta, and Anna Rumshisky. An efficient dp-sgd mechanism for large scale nlp models. *International Conference on Machine Learning*, 2021. pages

[21] Cynthia Dwork. Differential privacy. *International Colloquium on Automata, Languages, and Programming*, 2006. pages

[22] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. *Proceedings of the International Conference on Neural Information Processing Systems*, 2015. pages

[23] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 2014. pages

[24] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. {*USENIX*} *Security Symposium*, 2014. pages

[25] Liang Gao, Li Li, Yingwen Chen, Wenli Zheng, ChengZhong Xu, and Ming Xu. Fifl: A fair incentive mechanism for federated learning. *International Conference on Parallel Processing*, 2021. pages

[26] Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. On deep learning with label differential privacy. *Conference on Neural Information Processing Systems*, 2021. pages

[27] Jairo Giraldo, Alvaro Cardenas, Murat Kantarcioglu, and Jonathan Katz. Adversarial classification under differential privacy. *Network and Distributed Systems Security Symposium*, 2020. pages

[28] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015. pages

[29] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013. pages

[30] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. *International Conference on Machine Learning*, 2020. pages

[31] Shangwei Guo, Tianwei Zhang, Guowen Xu, Han Yu, Tao Xiang, and Yang Liu. Topology-aware differential privacy for decentralized image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 2020. pages

[32] Mikko Heikkilä, Eemil Lagerspetz, Samuel Kaski, Kana Shimizu, Sasu Tarkoma, and Antti Honkela. Differentially private bayesian learning on distributed data. *Conference on Neural Information Processing Systems*, 2017. pages

[33] Sanghyun Hong, Varun Chandrasekaran, Yiğitcan Kaya, Tudor Dumitraş, and Nicolas Papernot. On the effectiveness of mitigating data poisoning attacks with gradient shaping. *arXiv preprint arXiv:2002.11497*, 2020. pages

[34] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 1982. pages

[35] Md Tamjid Hossain, Shafkat Islam, Shahriar Badsha, and Haoting Shen. Desmp: Differential privacy-exploited stealthy model poisoning attacks in federated learning. *arXiv preprint arXiv:2109.09955*, 2021. pages

[36] Matthew Jagielski, Michael Kearns, Jieming Mao, Alina Oprea, Aaron Roth, Saeed Sharifi-Malvajerdi, and Jonathan Ullman. Differentially private fair learning. *International Conference on Machine Learning*, 2019. pages

[37] Matthew Jagielski and Alina Oprea. Does differential privacy defeat data poisoning? *International Conference on Learning Representations*, 2021. pages

[38] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private sgd? *Conference on Neural Information Processing Systems*, 2020. pages

[39] Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. {*USENIX*} *Security Symposium*, 2019. pages

[40] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Secure multi-party differential privacy. *Advances in neural information processing systems*, 2015. pages

[41] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 2011. pages

[42] Hamed F Langroudi, Cory Merkel, Humza Syed, and Dhireesha Kudithipudi. Exploiting randomness in deep learning algorithms. *International Joint Conference on Neural Networks*, 2019. pages

[43] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. *IEEE Symposium on Security and Privacy*, 2019. pages

[44] Xuechen Li, Florian Tramèr, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*, 2021. pages

[45] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. *Proceedings of the European Conference on Computer Vision*, 2018. pages

[46] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *International Conference on Learning Representations*, 2018. pages

[47] Frank D McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 2009. pages

[48] Xuying Meng, Suhang Wang, Kai Shu, Jundong Li, Bo Chen, Huan Liu, and Yujun Zhang. Personalized privacy-preserving social recommendation. *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. pages

[49] Fan Mo and Hamed Haddadi. Efficient and private federated learning using tee. *European Conference on Computer Systems*, 2019. pages

[50] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. Introduction to linear regression analysis. *John Wiley & Sons*, 2021. pages

[51] Milad Nasr et al. Adversary instantiation: Lower bounds for differentially private machine learning. *IEEE Symposium on Security and Privacy*, 2021. pages

[52] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with PATE. *International Conference on Learning Representations*, 2018. pages

[53] NhatHai Phan, Yue Wang, Xintao Wu, and Dejing Dou. Differential privacy preservation for deep auto-encoders: an application of human behavior prediction. *AAAI Conference on Artificial Intelligence*, 2016. pages

[54] NhatHai Phan, Xintao Wu, and Dejing Dou. Preserving differential privacy in convolutional deep belief networks. *Machine learning*, 2017. pages

[55] Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P Dickerson, and Tom Goldstein. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. *International Conference on Machine Learning*, 2021. pages

[56] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. *Proceedings of the ACM SIGSAC Conference on computer and communications security*, 2016. pages

[57] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. *IEEE Symposium on Security and Privacy*, 2017. pages

[58] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine learning models that remember too much. *Proceedings of the ACM SIGSAC Conference on computer and communications security*, 2017. pages

[59] Zongkun Sun, Yinglong Wang, Minglei Shu, Ruixia Liu, and Huiqi Zhao. Differential privacy for data and model publishing of medical data. *IEEE Access*, 2019. pages

[60] Songül Tolan, Marius Miron, Emilia Gómez, and Carlos Castillo. Why machine learning may lead to unfairness: Evidence from risk assessment for juvenile justice in catalonia. *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, 2019. pages

[61] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *International Conference on Learning Representations*, 2018. pages

[62] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. {*USENIX*} *Security Symposium*, 2016. pages

[63] Nurislam Tursynbek, Aleksandr Petiushko, and Ivan Oseledets. Robustness threats of differential privacy. *arXiv preprint arXiv:2012.07828*, 2020. pages

[64] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 1984. pages

[65] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 2017. pages

[66] Binghui Wang and Neil Zhenqiang Gong. Stealing hyperparameters in machine learning. *IEEE Symposium on Security and Privacy*, 2018. pages

[67] Yu-Xiang Wang, Jing Lei, and Stephen E Fienberg. Learning with differential privacy: Stability, learnability and the sufficiency and necessity of erm principle. *The Journal of Machine Learning Research*, 2016. pages

[68] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 2020. pages

[69] Bingzhe Wu, Shiwan Zhao, Guangyu Sun, Xiaolu Zhang, Zhong Su, Caihong Zeng, and Zhihong Liu. P3sgd: Patient privacy preserving sgd for regularizing deep cnns in pathological image classification. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. pages

[70] Haonan Yan, Xiaoguang Li, Hui Li, Jiamin Li, Wenhai Sun, and Fenghua Li. Monitoring-based differential privacy mechanism against query flooding-based model extraction attack. *IEEE Transactions on Dependable and Secure Computing*, 2021. pages

[71] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. *IEEE Computer Security Foundations Symposium*, 2018. pages

[72] Samuel Yeom, Irene Giacomelli, Alan Menaged, Matt Fredrikson, and Somesh Jha. Overfitting, robustness, and malicious algorithms: A study of potential causes of privacy risk in machine learning. *Journal of Computer Security*, 2020. pages

[73] Xue Ying. An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 2019. pages

[74] Da Yu, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. *International Conference on Learning Representations*, 2021. pages

[75] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. *International Conference on Machine Learning*, 2013. pages

[76] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 2021. pages

[77] Huadi Zheng, Qingqing Ye, Haibo Hu, Chengfang Fang, and Jie Shi. Bdpl: A boundary differentially private layer against machine learning model extraction attacks. *European Symposium on Research in Computer Security*, 2019. pages

[78] Tianqing Zhu, Gang Li, Wanlei Zhou, and S Yu Philip. Differential privacy and applications. *Springer*, 2017. pages

[79] Yuqing Zhu, Xiang Yu, Manmohan Chandraker, and Yu-Xiang Wang. Private-knn: Practical differential privacy for computer vision. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. pages