Conflicting Interactions among Protection Mechanisms for Machine Learning Models

Sebastian Szyller[1], **N. Asokan** [2,1]

[1]Aalto University
[2]University of Waterloo
contact@sebszyller.com, asokan@acm.org

Abstract

Nowadays, systems based on machine learning (ML) are widely used in different domains. Given their popularity, ML models have become targets for various attacks. As a result, research at the intersection of security/privacy and ML has flourished. Typically such work has focused on individual types of security/privacy concerns and mitigations thereof.

However, in real-life deployments, an ML model will need to be *protected against several concerns simultaneously*. A *protection mechanism* optimal for a specific security or privacy concern may interact negatively with mechanisms intended to address other concerns. Despite its practical relevance, the potential for such conflicts has not been studied adequately.

In this work, we first provide a framework for analyzing such *conflicting interactions*. We then focus on systematically analyzing pairwise interactions between protection mechanisms for one concern, *model and data ownership verification*, with two other classes of ML protection mechanisms: *differentially private training*, and *robustness against model evasion*. We find that several pairwise interactions result in conflicts.

We also explore potential approaches for avoiding such conflicts. First, we study the effect of hyperparameter relaxations, finding that there is no sweet spot balancing the performance of both protection mechanisms. Second, we explore whether modifying one type of protection mechanism (ownership verification) so as to decouple it from factors that may be impacted by a conflicting mechanism (differentially private training or robustness to model evasion) can avoid conflict. We show that this approach can indeed avoid the conflict between ownership verification mechanisms when combined with differentially private training, but has no effect on robustness to model evasion. We conclude by identifying the gaps in the landscape of studying interactions between other types of ML protection mechanisms.

1 Introduction

Machine learning (ML) models constitute valuable intellectual property. They are also increasingly deployed in risk-sensitive domains. As a result, various security and privacy requirements for ML model deployment have become apparent. This, in turn, has led to substantial recent research at the intersection of machine learning and security/privacy. The research community largely focuses on individual types of security/privacy threats and ways to defend against them. This facilitates iterative improvements, and allows practitioners to evaluate the benefit of any new approaches.

In this work, we argue that in realistic deployment setting, multiple security/privacy concerns need to be considered *simultaneously*. Therefore, any *protection mechanism* for a particular concern, needs to be tested together with defences against *other common concerns*. We show that when deployed together, ML protection mechanisms may not work as intended due to *conflicting interactions* among them.

We claim the following contributions:

1) We highlight the importance of understanding *conflicting interactions* among ML protection mechanisms, and provide a framework for studying it (Section 3).

2) We use our framework to analyse the interaction between *model ownership verification mechanisms* with two other types of protection mechanisms: *differentially private training* and *adversarial training*. We provide a theoretical justification (Section 4) for each potential pairwise conflict, and evaluate it empirically (Sections 5 and 6).

3) We explore whether conflicts can be avoided by changing (a) the hyperparameters of each protection mechanism, or (b) the design of the mechanism itself (Section 7).

2 Background

2.1 Machine Learning

The goal of a ML classification model $F_{\mathcal{V}}$ trained on some dataset $\mathcal{D}_{\text{TR}}$ is to perform well on the given classification task according to some metric ϕ measured on a test set $\mathcal{D}_{\text{TE}}$. The whole dataset is denoted as $\mathcal{D} = \{\mathcal{D}_{\text{TR}}, \mathcal{D}_{\text{TE}}\}$. An individual record consists of an input x and the corresponding label y. Throughout this work, we use the accuracy metric $\phi_{\text{ACC}}(F_{\mathcal{V}}, \mathcal{D}_{\text{TE}})$ to assess a model $F_{\mathcal{V}}$ using $\mathcal{D}_{\text{TE}}$:

$$\phi_{\text{ACC}}(F_{\mathcal{V}}, \mathcal{D}_{\text{TE}}) = \frac{1}{|\mathcal{D}_{\text{TE}}|} \sum_{x \in \mathcal{D}_{\text{TE}}} \mathbb{1}(\hat{F_{\mathcal{V}}}(x) = y). \qquad (1)$$

where $F_{\mathcal{V}}(x)$ is the full probability vector and $\hat{F_{\mathcal{V}}}(x)$ is the most likely class.

2.2 Ownership Verification

In a *white-box model stealing* attack an adversary $\mathcal{A}$ obtains an identical copy of $F_{\mathcal{V}}$ belonging to a victim $\mathcal{V}$, e.g., by breaking into a device, or bribing an employee. Watermarking can be used to deter white-box model stealing (Zhang et al. 2018; Adi et al. 2018; Uchida et al. 2017; Darvish Rouhani, Chen, and Koushanfar 2019).

On the other hand, in *black-box model extraction* attacks (Papernot et al. 2017; Juuti et al. 2019; Orekondy, Schiele, and Fritz 2019; Tramèr et al. 2016; Correia-Silva et al. 2018; Jagielski et al. 2020; Carlini, Jagielski, and Mironov 2020; Pal et al. 2019; **?**), $\mathcal{A}$ "steals" $F_{\mathcal{V}}$ by sending queries, recording responses, and using them to train a *surrogate* model $F_{\mathcal{A}}$.

Model extraction defences try to either *detect* (Juuti et al. 2019; Atli et al. 2020; Quiring, Arp, and Rieck 2018) or *slow down* (Orekondy, Schiele, and Fritz 2020; Dziedzic et al. 2022; Lee et al. 2018) the attack but cannot *prevent* it. Adversarial watermarking (Szyller et al. 2021) can *deter* extraction attacks by forcing a watermark into $F_{\mathcal{A}}$, or by ensuring that a watermark transfers from $F_{\mathcal{V}}$ to $F_{\mathcal{A}}$ (Jia et al. 2021).

Backdoor watermarking (Zhang et al. 2017; Adi et al. 2018; Szyller et al. 2021) (WM) allows $\mathcal{V}$ to embed an out-of-distribution watermark ("trigger set") $\mathcal{D}_{\text{T}}$ during training. $\mathcal{D}_{\text{T}}$ is chosen so that it does not interfere with the primary learning task, and is difficult to discover. Effectiveness of watermarks can be assessed via the accuracy of $F_{\mathcal{V}}$ on $\mathcal{D}_{\text{T}}$:

$$\phi_{\text{WM}}(F_{\mathcal{V}}, \mathcal{D}_{\text{T}}) = \frac{1}{|\mathcal{D}_{\text{T}}|} \sum_{x \in \mathcal{D}_{\text{T}}} \mathbb{1}(\hat{F_{\mathcal{V}}}(x) = y). \tag{2}$$

For $F_{\mathcal{V}}$ with m classes, the verification confidence is (Szyller et al. 2021):

$$V = \sum_{i=0}^{\lfloor e \times |\mathcal{D}_{\text{T}}| \rfloor} \binom{|\mathcal{D}_{\text{T}}|}{i} \times \left(\frac{m-1}{m}\right)^{i} \times \left(\frac{1}{m}\right)^{|\mathcal{D}_{\text{T}}|-i} \tag{3}$$

where $e = 1 - \phi_{\text{WM}}(F_{\mathcal{V}}, \mathcal{D}_{\text{T}})$ is the tolerated error rate for V. If $\mathcal{V}$ suspects a model $F_{\mathcal{A}}$ to be a copy of $F_{\mathcal{V}}$, it can query $F_{\mathcal{A}}$ with $\mathcal{D}_{\text{T}}$ to verify ownership.

Radioactive data (Sablayrolles et al. 2020) (RADDATA) is a *dataset watermarking* technique allowing $\mathcal{V}$ to identify models trained using their datasets. RADDATA embeds imperceptible perturbations in a subset of the training images $x_{\phi} = x + \phi$ which constitute a watermark $\mathcal{D}_{\text{RAD}}$. The perturbations are crafted iteratively using an optimization procedure similar to adversarial example search. Its goal is to align the images with a particular part of the manifold. Intuitively, $\mathcal{D}_{\text{RAD}}$ is more difficult to train on than the (clean) counterpart, and subsequently, more difficult to classify correctly with high confidence. Hence, any model trained on $\mathcal{D}_{\text{RAD}}$ would perform better on it.

The effectiveness of RADDATA can be measured using a white-box approach based on hypothesis testing or a black-box based on the loss difference. In this work, we use the black-box approach as it performs better in the original work (Tekgul and Asokan 2022). For a loss function $\mathcal{L}(F_{\mathcal{V}}(x), y)$ the black-box verification metric is defined as:

$$\begin{aligned}\phi_{\text{RAD}}(F_{\mathcal{V}}, \mathcal{D}_{\text{RAD}}) &= \frac{1}{|\mathcal{D}_{\text{RAD}}|} \times \\ &\times \sum_{\{x, x_{\phi}\} \in \mathcal{D}_{\text{RAD}}} \mathbb{1}(\mathcal{L}(F_{\mathcal{V}}(x), y) - \mathcal{L}(F_{\mathcal{V}}(x_{\phi}), y))\end{aligned} \tag{4}$$

$\phi_{\text{RAD}} > 0$ indicates that that $F_{\mathcal{V}}$ was trained on $\mathcal{D}_{\text{RAD}}$. The higher the value, the more confident the verification.

Dataset inference (Maini, Yaghini, and Papernot 2021) (DI) is a model *fingerprinting* technique. It assumes that $F_{\mathcal{V}}$ was trained on a private training dataset. It exploits the fact that an $F_{\mathcal{A}}$ extracted from $F_{\mathcal{V}}$ would learn similar features as $F_{\mathcal{V}}$.

To create the fingerprint, $\mathcal{V}$ extracts the feature embeddings from $F_{\mathcal{V}}$ that characterise their prediction margin using several distance metrics. The embeddings for $\mathcal{D}_{\text{TR}}$ and $\mathcal{D}_{\text{TE}}$ are used to train a distinguisher. During verification, $\mathcal{V}$ queries $F_{\mathcal{A}}$ with a subset of $\mathcal{D}_{\text{TR}}$, $D_{ver} \subset \mathcal{D}_{\text{TR}}$. $F_{\mathcal{A}}$ is deemed stolen if the distances are similar to $F_{\mathcal{V}}$ with sufficient confidence, under a hypothesis test. The verification is successful if the p-value (ϕ_{DI}) is below a certain threshold.

The embeddings can be obtained using a white-box technique (*MinGD*) or a black-box one (*Blind Walk*). In this work, we use Blind Walk approach as it performs better in the original work. The success of DI (ϕ_{DI}) is measured by comparing the embeddings using a hypothesis test. Distinguishable embeddings indicate that the model is stolen.

2.3 Model Evasion

In a model evasion attack (Biggio et al. 2013; Szegedy et al. 2014), an adversary $\mathcal{A}$ aims to craft an *adversarial example* $x_{\gamma} = x + \gamma$ such that it is misclassified by $F_{\mathcal{V}}$, $\hat{F_{\mathcal{V}}}(x) \neq \hat{F_{\mathcal{V}}}(x_{\gamma})$. Typically, the attack is restricted to produce inputs that are within γ distance (according to some distance measure, typically L_2 or L_{∞}) from the originals.

Adversarial training (ADVTR) is designed to provide robustness against adversarial examples. During training, each clean sample x is replaced with an adversarial example x_{γ}. Robustness can be measured by calculating the accuracy of the model on the adversarial test set:

$$\phi_{\text{ADV}}(F_{\mathcal{V}}, \mathcal{D}_{\text{TE}}) = \frac{1}{|\mathcal{D}_{\text{TE}}|} \sum_{x \in \mathcal{D}_{\text{TE}}} \mathbb{1}(\hat{F_{\mathcal{V}}}(x + \gamma) = y). \tag{5}$$

ADVTR is successful if ϕ_{ADV} is high (ideally the same as ϕ_{ACC}), and ϕ_{ACC} does not deteriorate.

There exist many techniques for crafting adversarial examples that in turn can be used in ADVTR. We use *projected gradient descent* (Madry et al. 2017) (PGD) a popular optimization technique for crafting adversarial examples.

2.4 Differential Privacy

Differential privacy (Dwork 2006) (DP) bounds $\mathcal{A}$'s capability to infer information about any individual record in $\mathcal{D}_{\text{TR}}$. The learning algorithm A satisfies (ϵ, δ)-differential privacy if for any two datasets D, D' that differ in one record, and any set of models Q:

$$Pr[A(D) \in Q] \leq e^{\epsilon} Pr[A(D') \in Q] + \delta \tag{6}$$

ϵ corresponds to the privacy budget, and δ is the probability mass for events where the privacy loss is larger than e^{ϵ}. Together, these two can be considered as ϕ_{DP}.

In this work, we use the most popular algorithm for differentially private training DPSGD (Abadi et al. 2016).

3 Problem Statement

The efficacy of a ML protection mechanism M with hyperparameters θ_M is measured by an associated metric ϕ_M. Many ML protection mechanisms tend to decrease ϕ_{ACC}. Therefore, the goal of a mechanism is to maximise *both* ϕ_M and ϕ_{ACC}. An individual instantiation m of M applied to the model F, $m(F, \theta_M)$, seeks to maximise both ϕ_M and ϕ_{ACC}:

$$m^* = \underset{m}{\operatorname{argmax}}\{\phi_M(m(F, \theta_M), \cdot), \phi_{\text{ACC}}(m(F, \theta_M), \cdot)\} \tag{7}$$

Consequently, the instantiation is effective iff.:

1. the difference, $\Delta_{\phi_M} = |\phi_M(F) - \phi_M(m(F, \theta_M))|$, for a given metric is *above* a threshold t_{ϕ_M}: $\Delta_{\phi_M} > t_{\phi_M}$.
2. the difference, $\Delta_{\phi_{\text{ACC}}} = |\phi_{\text{ACC}}(F) - \phi_{\text{ACC}}(m(F, \theta_M))|$, is *below* a threshold $t_{\phi_{\text{ACC}}}$: $\Delta_{\phi_{\text{ACC}}} < t_{\phi_{\text{ACC}}}$.

For instance, $\mathcal{V}$ may find that ADVTR with $\Delta_{\phi_{\text{ADV}}} < 0.3$ and $\Delta_{\phi_{\text{ACC}}} > 0.2$ is unacceptable. In reality, acceptable thresholds are application- and deployment-specific.

We can extend this to a combination of mechanisms $C = \{M_1, M_2, \ldots, M_n\}$. A combination c of multiple individual instantiations $\{m_1, m_2, \ldots, m_n\}$ applied to F, $c(F, \theta)$ where $\theta = \{\theta_{M_1}, \theta_{M_2}, \ldots, \theta_{M_n}\}$ is effective iff. all metrics $\phi_{M_1, M_2, \ldots, M_n}$ and ϕ_{ACC} are sufficiently high:

$$\begin{aligned} c^* = \underset{c=\{m_1, m_2, \ldots, m_n\}}{\operatorname{argmax}} \quad & \{\phi_{M_1}(c(F, \theta), \cdot), \\ & \phi_{M_2}(c(F, \theta), \cdot), \\ & \ldots \\ & \phi_{M_n}(c(F, \theta), \cdot), \\ & \phi_{\text{ACC}}(c(F, \theta), \cdot)\} \end{aligned} \tag{8}$$

In other words, the combination is effective iff. *all* $\Delta_{\phi_M} \in \{\Delta_{\phi_{M_1}}, \Delta_{\phi_{M_2}}, \ldots, \Delta_{\phi_{M_n}}\}$ are *below* their corresponding thresholds, and $\Delta_{\phi_{\text{ACC}}} < t_{\phi_{\text{ACC}}}$. Unlike for a single mechanism, here, Δ_M is calculated with c applied: $\Delta_\phi = |\phi(m(F, \theta_M), \cdot) - \phi(c(F, \theta), \cdot)|$.

Given a pair of mechanisms $C_{M_1, M_2} = \{M_1, M_2\}$, our goal is to determine if there exists an effective combination of instantiations c_{m_1, m_2} such that $\Delta_{\phi_{M_1}} < t_{\phi_{M_1}}$, $\Delta_{\phi_{M_2}} < t_{\phi_{M_2}}$ and $\Delta_{\phi_{\text{ACC}}} < t_{\phi_{\text{ACC}}}$. Subsequently, a pair is in *conflict* if any of these three inequalities does not hold. For a single mechanism, its threshold denotes required gain; while for a combination it corresponds to a maximum decrease.

Crucially, for a given pair C_{M_1, M_2}, both ϕ_Ms have an upper bound: $\phi_M(c(F, \theta), \cdot) \leq \phi_M(m(F, \theta_M), \cdot)$; and similarly, ϕ_{ACC} has an upper bound:

$$\begin{aligned} \phi_{\text{ACC}}(c(F, \theta)) \leq \min(&\phi_{\text{ACC}}(m_1(F, \theta_{M_1}), \cdot), \\ &\phi_{\text{ACC}}(m_2(F, \theta_{M_2}), \cdot)) \end{aligned} \tag{9}$$

Choosing thresholds. In this work, we use the following thresholds to decide if a combination of mechanisms is ineffective: $\Delta_{\phi_{\text{ACC}}} > 10pp$, or: 1) WM: $\Delta_{\phi_{\text{WM}}} > 30pp$; 2) ADVTR: $\Delta_{\phi_{\text{ADV}}} > 10pp$; 3) DI: $\phi_{\text{DI}} > 10^{-3}$; 4) RADDATA: $\phi_{\text{RAD}} < 10^{-2}$;

Note that DPSGD has a tight bound for the given (ϵ, δ) (Nasr et al. 2021). However, we consider increasing $\epsilon \times 1.5$ too permissive for the purpose of the changes discussed in Section 7.

4 Conflicting Interactions

We consider *pair-wise* interactions between protection mechanisms that allow for ownership verification {WM, DI, RADDATA } and techniques based on strong regularisers {ADVTR, DPSGD }. We first explain why a given pair may conflict. In Section 6 we verify our hypotheses empirically.

4.1 Pair-wise Conflicts

DPSGD with WM. WM relies on overfitting $F_{\mathcal{V}}$ to the trigger set $\mathcal{D}_{\text{T}}$ (memorisation) while simultaneously trying to learn the primary dataset $\mathcal{D}_{\text{TR}}$. In turn, the gradient norm of $\mathcal{D}_{\text{T}}$ is high which is necessary to provide sufficient signal. On the other hand, DPSGD relies on two primary mechanisms that limit the contribution of individual samples: 1) clipping and 2) adding noise to the gradients. These induce strong regularization on $\mathcal{D}_{\text{TR}}$. Hence, these two techniques impose contradictory objectives for the training to optimize for. Therefore, we conjecture that they conflict.

ADVTR with WM. ADVTR provides a strong regularising property to the model that typically results in a decrease in ϕ_{ACC} for non-trivial values of γ. Similarly to the interaction with DPSGD, we suspect that the regularization induced by ADVTR will harm the embedding of $\mathcal{D}_{\text{T}}$. Furthermore, $\mathcal{D}_{\text{T}}$ introduces *pockets* of OOD data, and it was shown that $\mathcal{D}_{\text{T}}$ is indistinguishable from $\mathcal{D}_{\text{TR}}$ in the final layer (Szyller et al. 2021). These may make finding adversarial examples easier.

DPSGD or ADVTR with DI. DI relies on the fact that $F_{\mathcal{A}}$ derived from $F_{\mathcal{V}}$ has similar decision boundaries. Because DPSGD limits the contribution of individual samples at the gradient level, the decision boundaries of $F_{\mathcal{V}}$ trained with and without it may differ. ADVTR changes the decision boundary around training records, and may conflict with DI.

ADVTR with RADDATA. RADDATA relies on the optimization procedure that is similar to finding adversarial examples. Hence, we expect ADVTR to prevent $\mathcal{D}_{\text{RAD}}$ from being embedded. It is unclear if the presence of $\mathcal{D}_{\text{RAD}}$ is going to negatively impact ϕ_{ADV}.

DPSGD with RADDATA. Like WM, RADDATA requires embedding information that differs from $\mathcal{D}_{\text{TR}}$. We expect DPSGD to limit the contribution of $\mathcal{D}_{\text{RAD}}$ and decrease, or invalidate ϕ_{RAD}.

4.2 Result Significance

For all experiments we measure the statistical significance of the result. We test ϕ_{ACC}, and then each ϕ_M separately. We first conduct a *t-test* under the null hypothesis $\mathcal{H}_0$ of *equivalent population distributions* with $\alpha = 0.05$. Next, if $\mathcal{H}_0$ is rejected, we conduct a *two one-sided test* to see if the result falls within the *equivalence bounds* (for which we use the abovementioned thresholds). Here, the null hypothesis $\mathcal{H}_0^*$ is reversed and we assume *non-equivalence*, $\alpha^* = 0.05$. $\mathcal{H}_0^*$ is rejected if the results fall within the bound. For both, we use Welch's t-test since sample variances are unequal.

5 Experimental Setup

Datasets. we use three benchmark datasets for our evaluation. MNIST (LeCun, Cortes, and Burges 2010) contains $60,000$ train and $10,000$ test grayscale images of digits.

Dataset	$\|\mathcal{D}_{\text{TR}}\|$	$\|\mathcal{D}_{\text{TE}}\|$	Arch.	LR.	Epochs
MNIST	$50k$	$10k$	4L-CNN	0.001	25/100
FMNIST	$60k$	$10k$	4L-CNN	0.001	25/100
CIFAR10	$50k$	$10k$	RN20	0.005	100/200
GTSRB	$\approx 40k$	$\approx 13k$	-	-	-

Table 1: Datasets, model architecture (Arch.), and training learning rate (LR.); number of epochs reported as: *baseline (unprotected)/protected*.

The corresponding label is the digit presented in the image. FashionMNIST (Xiao, Rasul, and Vollgraf 2017) (FMNIST) contains $60,000$ train and $10,000$ test grayscale images of articles of clothing, divided into 10 classes. The corresponding label is the piece of clothing presented in the image. CIFAR10 (Krizhevsky and Hinton 2009) contains $50,000$ train and $10,000$ test RGB images which depict miscellaneous animals or vehicles, divided into 10 classes.

For MNIST and FMNIST, we use each as the out-of-distribution dataset from which watermarks for the other are drawn. For CIFAR10, we follow prior work (Szyller et al. 2021) in using GTSRB (Stallkamp et al. 2011) as the source of out-of-distribution watermarks. GTSRB is a traffic sign dataset that contain $39,209$ train and $12,630$ test RGB images, divided into 43 classes.

Models and training. for MNIST and FMNIST, we use a simple 4-layer CNN. We train the models for 25 epochs for the baselines, and 100 for experiments with protection mechanisms deployed. For all experiments we use the initial learning rate of 0.001 and maximum learning rate of 0.005 with a one-cycle scheduler (Smith and Topin 2019).

For CIFAR10, we use a ResNet20. We train the models for 100 epochs for the baselines, and 200 for experiments with protection mechanisms deployed. Similarly to MNIST and FMNIST case, we use a one-cycle scheduler. However, we use the initial learning rate of 0.005 and maximum starting learning of 0.1. We summarise these details in Table 1.

For baselines models, and those with only one mechanism, training was repeated five times; for pair-wise comparisons, training was repeated ten times. All training was done, on a workstation with two NVIDIA RTX 3090, Threadripper 3960X, and 128 GB of RAM. We used the PyTorch library to train the models. We use official repositories of techniques that we evaluate: WM[1], DI[2], RADDATA[3]. The code for this project is available on GitHub[4].

6 Evaluation

We report on our experiments for studying how ownership verification mechanisms interact with DP (Section 6.1) and ADVTR (Section 6.2). We color-code all results to convey potential conflicts (e.g., 0.3).

We used the following hyperparameters: 1) for DPSGD, clipping norm $c = 1.0$, $\epsilon = 3$, $\delta = 10^{-6}$ for MNIST and FMNIST, and $\delta = 10^{-5}$ for CIFAR10; 2) for ADVTR, $\gamma = 0.25$ for MNIST and FMNIST, and $\gamma = 10/255$ for CIFAR10; 3) for WM, $|\mathcal{D}_{\text{T}}| = 100$; for RADDATA, watermarked ratio of 10%. Table 2 gives the baseline results for all three datasets; Table 3 summarizes the hyperparameters of each protection mechanism.

Note that we chose the CIFAR10 architecture (ResNet20) that is capable of supporting WM, RADDATA and DI. As a result, ϕ_{ACC} with DPSGD is relatively low. There exist mechanisms that achieve higher ϕ_{ACC}; however, they have additional implications. We discuss this in Section 8.

6.1 Impact of Differential Privacy

Table 4 gives the results for combining DPSGD with WM, RADDATA, and DI. In all cases, ϕ_{ACC} remains close to the single-mechanism baselines. For MNIST and CIFAR10, $\mathcal{H}_0$ cannot be rejected (not enough evidence to indicate that accuracy differs between the baseline(s) and the jointly protected instances). For FMNIST $\mathcal{H}_0$ is rejected, but $\mathcal{H}_0^*$ is rejected as well, leading us to conclude that the accuracy is the same within the equivalence bounds.

WM. For all datasets, ϕ_{WM} drops significantly. $\mathcal{H}_0$ is rejected, and $\mathcal{H}_0^*$ is close to 1.0 for the equivalence bound of $30pp$. Samples in $\mathcal{D}_{\text{T}}$ are outliers, and require memorization for successful embedding. DPSGD, by design, bounds the contribution of individual samples during training.

RADDATA. DPSGD lowers ϕ_{RAD} but it remains high enough for successful verification. Both $\mathcal{H}_0$ and $\mathcal{H}_0^*$ are rejected indicating that while the results are different they are within equivalence bounds. The regularizing effect of DPSGD is insufficient to prevent $F_{\mathcal{V}}$ from learning $\mathcal{D}_{\text{RAD}}$. Unlike in WM, RADDATA modifies the samples such that they align with a few selected carriers. Hence, multiple samples in $\mathcal{D}_{\text{RAD}}$ nudge the model in the same direction allowing it to learn the watermark.

DI. DI retains high verification confidence ($\mathcal{H}_0$ cannot be rejected).

In summary, ϕ_{ACC} remains high in all cases. WM performance is destroyed, RADDATA has reduced effectiveness but not enough to declare a conflict, and DI has no conflict.

6.2 Impact of Adversarial Training

Table 5 gives the results for combining ADVTR with WM, RADDATA, and DI. In all cases, ϕ_{ACC} on average remains close to the single-mechanism baselines. For the equivalence bound of $10pp$, $\mathcal{H}_0$ can and $\mathcal{H}_0^*$ cannot be rejected only for MNIST. For FMNIST and CIFAR10, $\mathcal{H}_0^*$ gives a p-value of 0.15 and 0.07 respectively. However, a slightly larger threshold $t_{\phi_{\text{ACC}}} = 11pp$, we obtain a p-value of 0.005 and 0.002. Therefore, we deem that the combination does not affect ϕ_{ACC} enough to declare a conflict based on ϕ_{ACC}.

WM. ϕ_{WM} remains close to the baselines. However, ϕ_{ADV} drops at least $10pp$ for FMNIST and CIFAR10; $\mathcal{H}_0^*$ is close to 1.0. This is a surprising result because $\mathcal{D}_{\text{T}}$s are chosen to be far from the distribution of $\mathcal{D}_{\text{TR}}$. We conjecture that $\mathcal{D}_{\text{T}}$ is in fact quite close to $\mathcal{D}_{\text{TR}}$ in the weight manifold, and be-

[1] https://github.com/ssg-research/dawn-dynamic-adversarial-watermarking-of-neural-networks

[2] https://github.com/cleverhans-lab/dataset-inference

[3] https://github.com/facebookresearch/radioactive_data

[4] https://github.com/ssg-research/conflicts-in-ml-protection-mechanisms

Dataset	No Def.	ADVTR		DPSGD	WM		RADDATA		DI
	ϕ_{ACC}	ϕ_{ACC}	ϕ_{ADV}	ϕ_{ACC}	ϕ_{ACC}	ϕ_{WM}	ϕ_{ACC}	ϕ_{RAD}	ϕ_{DI}
MNIST	0.99±0.00	0.99±0.00	0.95±0.00	0.98±0.00	0.99±0.00	0.97±0.01	0.98±0.00	0.284±0.001	$< 10^{-30}$
FMNIST	0.91±0.00	0.87±0.00	0.69±0.00	0.86±0.01	0.87±0.02	0.99±0.02	0.88±0.01	0.191±0.002	$< 10^{-30}$
CIFAR10	0.92±0.00	0.88±0.00	0.82±0.00	0.38±0.00	0.82±0.00	0.97±0.02	0.85±0.00	0.202±0.001	$< 10^{-30}$

Table 2: Baseline models without any protection mechanisms (No Def.), and with a single mechanism deployed. We provide ϕ_{ACC} and the corresponding metric for each mechanism. Results are averaged over 5 runs; we report the mean and standard deviation rounded to two decimal places (three for RADDATA).

Dataset	ADVTR	DPSGD			WM	RADDATA
	γ	ϵ	δ	c	$\lvert\mathcal{D}_{\text{T}}\rvert$	$\mathcal{D}_{\text{RAD}}$ %
MNIST	0.25	3	10^{-6}	1.0	100	10%
FMNIST	0.25	3	10^{-6}	1.0	100	10%
CIFAR10	10/255	3	10^{-5}	1.0	100	10%

Table 3: Summary of the hyperparameters for each protection mechanism.

cause it has random labels, it is easier for the evasion attack to find a perturbation that leads to a misclassification.

RADDATA. On the other hand, RADDATA is rendered ineffective while ϕ_{ADV} stays high. ϕ_{RAD} drops close to zero which leads to a low confidence verification. $\mathcal{H}_0$ is rejected but $\mathcal{H}_0^*$ is not (the result is significantly below 10^{-2}). RADDATA relies on replacing some samples in $\mathcal{D}_{\text{TR}}$ with samples similar to adversarial examples. It then exploits the difference in the loss on clean and perturbed samples for dataset ownership verification. ADVTR replaces all data in $\mathcal{D}_{\text{TR}}$ with an adversarial variant, and hence, invalidates the mechanism used by RADDATA.

DI. Similarly to the pairing with DPSGD, confidence of DI remains high.

In summary, ϕ_{ACC} remains high in all cases. WM and DI remain effective, while RADDATA performance is destroyed. ϕ_{ADV} stays high both for DI and RADDATA but is decreased for WM. Only DI has no conflict with ADVTR. However, we observed that DI can result in false positives when $F_{\mathcal{V}}$ and $F_{\mathcal{A}}$ where trained using $\mathcal{D}_{\text{TR}}$ from the same distribution, even though, $F_{\mathcal{A}}$ is benign. We discuss this further in Section 8 and the extended technical report (Szyller and Asokan 2022).

7 Addressing the Conflicts

Having established that there are multiple instances of conflicting interactions among ML protection mechanisms, we now explore how we might avoid conflicts.

First, we investigated whether settling for a weaker protection guarantee of one mechanism meaningfully boosts the performance of the other. However, the changes either do not sufficiently improve any metric, or require significantly lowering the protection guarantee. See the extended technical report for details (Szyller and Asokan 2022).

Second, we separated the training objective of WM and from the regularization imposed by ADVTR and DPSGD, and check if it allows the model to recover some of its original effectiveness. So far, we used the mechanisms without differentiating between $\mathcal{D}_{\text{TR}}$, and $\mathcal{D}_{\text{T}}$ or $\mathcal{D}_{\text{RAD}}$. Instead, one could apply these mechanisms only to the primary training task with $\mathcal{D}_{\text{TR}}$, and use a separate parameter optimizer for $\mathcal{D}_{\text{T}}$ or RADDATA. We evaluate such modification for the pairs in conflict: 1) WM with ADVTR; 2) RADDATA with ADVTR; 3) WM with DPSGD.

WM with ADVTR. We observed a minor improvement in ϕ_{ACC}. However, it does not substantially improve ϕ_{ADV}, although, it does reduce the standard deviation across runs. ϕ_{WM} remains high as in the previous experiments (Table 6).

RADDATA with ADVTR. We observed a minor improvement in ϕ_{ACC} and ϕ_{ADV}. However, ϕ_{RAD} did not substantially improve (Table 7). The pair remains in conflict.

WM with DPSGD. For MNIST and FMNIST, $F_{\mathcal{V}}$ achieves high ϕ_{ACC} and ϕ_{WM}, comparable to the baseline (Table 8). For CIFAR10, ϕ_{WM} improves significantly but remains low enough to declare a conflict.

However, it begs the question if these models are still (ϵ, δ)-private. The goal of DPSGD is to provide private *training* by restricting the updates to model's *weights*. On one hand, using a regular SGD for WM, and DPSGD for $\mathcal{D}_{\text{TR}}$ breaks this assumption. On the other, pre-training on public data (without privacy) and fine-tunning on a private $\mathcal{D}_{\text{TR}}$ has become the de facto way of training accurate private models (Tramèr and Boneh 2021; Kurakin et al. 2022).

Additionally, DP is often further relaxed to consider only *computationally restricted adversaries* (Mironov et al. 2009), which provides the guarantee only for realistic datasets as opposed to any $\mathcal{D}_{\text{TR}}$. $\mathcal{D}_{\text{T}}$ could be considered irrelevant from the privacy standpoint.

8 Discussion

Model Size & Convergence. Insufficient model capacity and lack of convergence could be the source of the conflicts. The conflict between RADDATA and ADVTR arises because ADVTR prevents watermarks from being embedded, and is thus independent of model size. In our experiments, all models reach low training loss, and expected accuracy. Hence, larger models are unlikely to resolve the conflict between ADVTR and WM. Finally, for DP, larger models deplete the privacy budget faster leading to lower accuracy.

Other Mechanisms. DPSGD is the most popular mechanism for DP training, but not the best performing one. A recently proposed mechanism, ScatterDP (Tramèr and Boneh 2021), relies on training a classifier (logistic regression or small CNN) with DPSGD on top of features in the fre-

Dataset	DPSGD	WM			RADDATA			DI
	Baseline	Baseline	+DPSGD		Baseline	+DPSGD		+DPSGD
	ϕ_{ACC}	ϕ_{WM}	ϕ_{ACC}	ϕ_{WM}	ϕ_{RAD}	ϕ_{ACC}	ϕ_{RAD}	ϕ_{DI}
MNIST	0.98±0.00	0.97±0.01	0.97±0.00	$\underline{0.36\pm0.06}$	0.284±0.001	0.97±0.00	0.091±0.01	$< 10^{-30}$
FMNIST	0.86±0.01	0.99±0.02	0.86±0.00	$\underline{0.30\pm0.05}$	0.191±0.002	0.84±0.01	0.11±0.01	$< 10^{-30}$
CIFAR10	0.38±0.00	0.97±0.02	0.38±0.01	$\underline{0.12\pm0.01}$	0.202±0.001	0.35±0.01	0.19±0.01	$< 10^{-30}$

Table 4: Simultaneous deployment of DPSGD with WM, RADDATA and DI. WM drops over $30pp$. The loss difference for RADDATA is reduced but still allows for confident verification. DI is unaffected. ϕ_{ACC} remains close to the baseline value in all cases. Results are averaged over 10 runs; we report the mean and standard deviation rounded to two decimal places (three for RADDATA). Underline indicates conflict - outside the equivalence bound.

Dataset	ADVTR	WM			RADDATA				DI
	Baseline	+ADVTR			Baseline	+ADVTR			+ADVTR
	ϕ_{ADV}	ϕ_{ACC}	ϕ_{WM}	ϕ_{ADV}	ϕ_{RAD}	ϕ_{ACC}	ϕ_{RAD}	ϕ_{ADV}	ϕ_{DI}
MNIST	0.95±0.00	0.97±0.02	0.99±0.01	0.88±0.09	0.284±0.001	0.94±0.01	$\underline{0.001\pm0.001}$	0.95±0.01	$< 10^{-30}$
FMNIST	0.69±0.00	0.80±0.06	0.99±0.00	$\underline{0.51\pm0.11}$	0.191±0.002	0.87±0.02	$\underline{0.000\pm0.001}$	0.69±0.02	$< 10^{-30}$
CIFAR10	0.82±0.00	0.78±0.00	0.97±0.01	$\underline{0.65\pm0.01}$	0.202±0.001	0.81±0.01	$\underline{0.003\pm0.002}$	0.81±0.01	$< 10^{-30}$

Table 5: Simultaneous deployment of ADVTR with WM, RADDATA and DI. ADVTR does not interfere with WM, ϕ_{WM} remains high; however, ϕ_{ADV} drops at least $10pp$ for FMNIST and CIFAR10. RADDATA is rendered ineffective as ϕ_{ADV} drops almost to zero. DI is unaffected. ϕ_{ACC} remains close to the baseline value in all cases. Results are averaged over 10 runs; we report the mean and standard deviation rounded to two decimal places (three for RADDATA). Underline indicates conflict - outside the equivalence bound.

quency domain, obtained by transforming images with a ScatterNet. A small classifier does not have enough capacity to embed a watermark, and is more robust to perturbed inputs. Using a bigger CNN removes the benefit of using ScatterDP. Therefore, we deem that ScatterDP conflicts with ownership verification mechanisms because it does not admit joint deployment.

We do not evaluate pre-training on public data or any mechanisms that require it (e.g. PATE (Papernot et al. 2018)). Use of public data is not realistic in many industries, and has been primarily used for general purpose image and text models. For instance, healthcare data typically cannot be disclosed due to the privacy regulation; financial institutions have lengthy and restrictive compliance procedures.

Limitations of Protection Mechanisms. Most protection mechanisms evaluated in this work were shown to fall short when faced with a strong $\mathcal{A}$. WM can be removed or prevented from embedding (Lukas et al. 2022). ADVTR does not generalise to higher γ values (Nie et al. 2022). Generally, attacks and defences against model evasion are defeated by novel approaches (Carlini and Wagner 2017; Radiya-Dixit and Tramèr 2021). DP requires careful, *a priori* assumptions which often are not realistic (Domingo-Ferrer, Sánchez, and Blanco-Justicia 2021), and was recently shown to be vulnerable to side-channel timing attacks (Jin et al. 2022).

We also observed that DI results in false positives for models independently trained on a dataset with the same distribution as $F_{\mathcal{V}}$'s $\mathcal{D}_{\text{TR}}$ even if it *does not* overlap with $F_{\mathcal{V}}$'s $\mathcal{D}_{\text{TR}}$. Consequently, DI may result in innocent parties being falsely accused of stealing $F_{\mathcal{V}}$ (see the technical report for more information (Szyller and Asokan 2022)). While DI avoids conflicts with other protection mechanisms we studied so far, we caution against using DI in domains where uniqueness of $F_{\mathcal{V}}$'s training data cannot be guaranteed.

Stakeholders in the Training Loop. In a simple setting, a single party gathers the data, trains and deploys the model. Hence, if $\mathcal{V}$ cares about data or model ownership they could decide to forgo ADVTR or DP.

However, as ML services increasingly specialise, it is likely that different parties will be responsible for gathering data, providing the training platform, deploying the model, and using it. Thus, even if the party deploying a model may not care about traceability with RADDATA, another involved party may. Similarly, the training platform provider may want to embed a watermark to ensure that users conform to the terms of service, and not e.g. share it with others, or offer their own service using a knock-off of $F_{\mathcal{V}}$.

We can consider a scenario where $\mathcal{V}$ concerned about model evasion, buys data from a party that uses RADDATA, and $F_{\mathcal{V}}$ is trained by some service that embeds a watermark. ADVTR conflicts both with RADDATA and WM. Hence, data/platform provider needs to communicate up front that their offering is not compatible with certain training strategies, or resort to changes discussed in Section 8.

Combinatorial Explosion. This work could be further extended to include triples or quadruples of protection mechanisms simultaneously. Although, some of them could be considered toy cases, there are many combinations that reflect actual deployment considerations, e.g. DP, ADVTR, WM, while ensuring fairness.

However, increasing the size of the tuple leads to a *combinatorial explosion* of the number of ways we can combine the protection mechanisms. This number is likely to grow as new types of vulnerabilities are discovered, and does not

Dataset	ADVTR	WM							
	Baseline	Baseline		+ADVTR			+ADVTR Relaxed		
	ϕ_{ADV}	ϕ_{ACC}	ϕ_{WM}	ϕ_{ACC}	ϕ_{WM}	ϕ_{ADV}	ϕ_{ACC}	ϕ_{WM}	ϕ_{ADV}
MNIST	0.95±0.00	0.99±0.00	0.97±0.01	0.97±0.02	0.99±0.01	0.88±0.09	0.97±0.01	0.99±0.01	0.89±0.01
FMNIST	0.69±0.00	0.87±0.02	0.99±0.02	0.80±0.06	0.99±0.00	$\underline{0.51\pm0.11}$	0.84±0.01	0.99±0.00	$\underline{0.51\pm0.05}$
CIFAR10	0.82±0.00	0.82±0.00	0.97±0.02	0.78±0.00	0.97±0.01	$\underline{0.65\pm0.01}$	0.80±0.01	0.90±0.01	$\underline{0.69\pm0.01}$

Table 6: Training on $\mathcal{D}_{\text{TR}}$ with ADVTR and without on $\mathcal{D}_{\text{T}}$. The change does not result in any meaningful improvement. Underline indicates conflict.

Dataset	ADVTR	RADDATA						
	Baseline	Baseline	+ADVTR			+ADVTR Relaxed		
	ϕ_{ADV}	ϕ_{RAD}	ϕ_{ACC}	ϕ_{RAD}	ϕ_{ADV}	ϕ_{ACC}	ϕ_{RAD}	ϕ_{ADV}
MNIST	0.95±0.00	0.284±0.001	0.94±0.01	$\underline{0.001\pm0.001}$	0.95±0.01	0.94±0.02	$\underline{0.002\pm0.001}$	0.94±0.03
FMNIST	0.69±0.00	0.191±0.002	0.87±0.02	$\underline{0.000\pm0.001}$	0.69±0.02	0.87±0.01	$\underline{0.002\pm0.002}$	0.69±0.02
CIFAR10	0.82±0.00	0.202±0.001	0.81±0.01	$\underline{0.003\pm0.002}$	0.81±0.01	0.82±0.02	$\underline{0.004\pm0.001}$	0.81±0.02

Table 7: Training on $\mathcal{D}_{\text{TR}}$ with ADVTR and without on $\mathcal{D}_{\text{RAD}}$. The change does not result in any meaningful improvement. Underline indicates conflict.

Dataset	WM			
	+DPSGD		+DPSGD Relaxed	
	ϕ_{ACC}	ϕ_{WM}	ϕ_{ACC}	ϕ_{WM}
MNIST	0.97±0.00	$\underline{0.36\pm0.06}$	0.97±0.01	0.97±0.01
FMNIST	0.86±0.00	$\underline{0.30\pm0.05}$	0.87±0.01	0.99±0.02
CIFAR10	0.38±0.01	$\underline{0.12\pm0.01}$	0.39±0.02	$\underline{0.67\pm0.04}$

Table 8: Training on $\mathcal{D}_{\text{TR}}$ with DPSGD and without on $\mathcal{D}_{\text{T}}$. We recover performance close to the baseline. Underline indicates conflict.

account for multiple mechanisms within a single category.

9 Related Work

We summarise the prior work on the interactions between properties in the context of ML security/privacy.

It was shown that DP can be used to certify robustness to model evasion (Lecuyer et al. 2019) by limiting the contribution of an individual pixel. Prior work has extensively proved that using DP degrades fairness of the models and can exacerbate bias present in the dataset as well as the performance on the downstream tasks (Chang and Shokri 2021; Cheng et al. 2021; Pearce 2022). Membership inference attacks (MIAs) were used to evaluate the privacy guarantee of DPSGD (Nasr et al. 2021), and it was argued that DP should provide resistance to them (Humphries et al. 2020).

It was suggested that poisoning attacks can be used to make models vulnerable to other threats. One can inject samples into the training set to make MIAs easier (Tramèr et al. 2022). Also, there is a connection between adversarial robustness and susceptibility to poisoning (Pang et al. 2020). Furthermore, adversarial robustness can make models more vulnerable to MIAs (Song, Shokri, and Mittal 2019).

Nevertheless, it was shown that ADVTR can make models more interpretable (Tsipras et al. 2019). Lastly, LIME (Ribeiro, Singh, and Guestrin 2016), a popular explainability method, was used to compare the similarity of models (Jia et al. 2022). However, post-hoc explainability methods can be used to speed up model evasion, model extraction, and membership inference (Quan et al. 2022).

10 Conclusion

In this work, we pose the problem of *conflicting interactions* between different ML protection mechanisms. We provide a framework for evaluating simultaneous deployment of multiple mechanisms. We use it explore the interaction between three ownership verification mechanisms (WM, DI, RADDATA) with the most popular methods for preventing model evasion (ADVTR), and differentially private training (DPSGD). We show there exists a theoretical and empirical conflict that limits the effectiveness of multiple mechanisms.

Moving forward, researchers working on ML protection mechanisms should extend their evaluation benchmarks to include conflicts with other common concerns. In turn, it allows practitioners to choose the most appropriate mechanisms for their deployment scenario and threat model. We emphasize that this is not merely an "engineering problem" that can be ignored during the research phase but a key consideration for any technique and its deployability prospects.

Certain considerations are relevant only to particular applications, e.g. fairness is at odds with privacy but it might not be important when used in a closed loop system. Similarly, adversarial training may hurt data-based watermarking and fingerprinting mechanisms but is outside of the threat model of systems that do not have a user facing interface.

Many pairs still require explicit analysis that could unravel surprising limitations. We encourage the community to build upon this work, and extend it to more pairs and tuples.

Acknowledgements

We thank Pui Yan Chloe Sham, Dmytro Shynkevych and Asim Waheed for contributing to the early stages of this project. We thank Jason Martin whose remarks set us on the path of exploring conflicting interactions among ML protection mechanisms. This work was supported in part by Intel (in the context of the Private-AI Institute).

References

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, 308–318. New York, NY, USA: Association for Computing Machinery. ISBN 9781450341394.

Adi, Y.; Baum, C.; Cisse, M.; Pinkas, B.; and Keshet, J. 2018. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX Security Symposium*, 1615–1631.

Atli, B. G.; Szyller, S.; Juuti, M.; Marchal, S.; and Asokan, N. 2020. Extraction of Complex DNN Models: Real Threat or Boogeyman? In *Engineering Dependable and Secure Machine Learning Systems*, 42–57.

Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Š rndić, N.; Laskov, P.; Giacinto, G.; and Roli, F. 2013. Evasion Attacks against Machine Learning at Test Time. In *Advanced Information Systems Engineering*, 387–402. Springer Berlin Heidelberg.

Carlini, N.; Jagielski, M.; and Mironov, I. 2020. Cryptanalytic Extraction of Neural Network Models. In Micciancio, D.; and Ristenpart, T., eds., *Advances in Cryptology – CRYPTO 2020*, 189–218. Cham: Springer International Publishing. ISBN 978-3-030-56877-1.

Carlini, N.; and Wagner, D. 2017. Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, AISec '17, 3–14. New York, NY, USA: Association for Computing Machinery. ISBN 9781450352024.

Chang, H.; and Shokri, R. 2021. On the Privacy Risks of Algorithmic Fairness. In *2021 IEEE European Symposium on Security and Privacy (EuroS P)*, 292–303.

Cheng, V.; Suriyakumar, V. M.; Dullerud, N.; Joshi, S.; and Ghassemi, M. 2021. Can You Fake It Until You Make It? Impacts of Differentially Private Synthetic Data on Downstream Classification Fairness. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, 149–160. New York, NY, USA: Association for Computing Machinery. ISBN 9781450383097.

Correia-Silva, J. R.; Berriel, R. F.; Badue, C.; de Souza, A. F.; and Oliveira-Santos, T. 2018. Copycat CNN: Stealing Knowledge by Persuading Confession with Random Non-Labeled Data. In *2018 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Darvish Rouhani, B.; Chen, H.; and Koushanfar, F. 2019. DeepSigns: An End-to-End Watermarking Framework for Ownership Protection of Deep Neural Networks. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, 485–497. ACM.

Domingo-Ferrer, J.; Sánchez, D.; and Blanco-Justicia, A. 2021. The Limits of Differential Privacy (and Its Misuse in Data Release and Machine Learning). *Commun. ACM*, 64(7): 33–35.

Dwork, C. 2006. Differential privacy. volume 2006, 1–12. ICALP.

Dziedzic, A.; Kaleem, M. A.; Lu, Y. S.; and Papernot, N. 2022. Increasing the Cost of Model Extraction with Calibrated Proof of Work. In *International Conference on Learning Representations*.

Humphries, T.; Oya, S.; Tulloch, L.; Rafuse, M.; Goldberg, I.; Hengartner, U.; and Kerschbaum, F. 2020. Investigating Membership Inference Attacks under Data Dependencies. arXiv:2010.12112.

Jagielski, M.; Carlini, N.; Berthelot, D.; Kurakin, A.; and Papernot, N. 2020. High Accuracy and High Fidelity Extraction of Neural Networks. In *29th USENIX Security Symposium (USENIX Security 20)*.

Jia, H.; Chen, H.; Guan, J.; Shamsabadi, A. S.; and Papernot, N. 2022. A Zest of LIME: Towards Architecture-Independent Model Distances. In *International Conference on Learning Representations*.

Jia, H.; Choquette-Choo, C. A.; Chandrasekaran, V.; and Papernot, N. 2021. Entangled Watermarks as a Defense against Model Extraction. In Bailey, M.; and Greenstadt, R., eds., *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, 1937–1954. USENIX Association.

Jin, J.; McMurtry, E.; Rubinstein, B. I. P.; and Ohrimenko, O. 2022. Are We There Yet? Timing and Floating-Point Attacks on Differential Privacy Systems. In *2022 IEEE Symposium on Security and Privacy (SP)*, 473–488.

Juuti, M.; Szyller, S.; Marchal, S.; and Asokan, N. 2019. PRADA: Protecting against DNN Model Stealing Attacks. In *IEEE European Symposium on Security & Privacy*, 1–16. IEEE.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario.

Kurakin, A.; Song, S.; Chien, S.; Geambasu, R.; Terzis, A.; and Thakurta, A. 2022. Toward Training at ImageNet Scale with Differential Privacy. arXiv:2201.12328.

LeCun, Y.; Cortes, C.; and Burges, C. 2010. MNIST handwritten digit database. *AT&T Labs*.

Lecuyer, M.; Atlidakis, V.; Geambasu, R.; Hsu, D.; and Jana, S. 2019. Certified Robustness to Adversarial Examples with Differential Privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, 656–672.

Lee, T.; Edwards, B.; Molloy, I.; and Su, D. 2018. Defending Against Model Stealing Attacks Using Deceptive Perturbations. *arXiv preprint arXiv:1806.00054*.

Lukas, N.; Jiang, E.; Li, X.; and Kerschbaum, F. 2022. SoK: How Robust is Image Classification Deep Neural Network Watermarking? *2022 IEEE Symposium on Security and Privacy (SP)*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Maini, P.; Yaghini, M.; and Papernot, N. 2021. Dataset Inference: Ownership Resolution in Machine Learning. In *International Conference on Learning Representations*.

Mironov, I.; Pandey, O.; Reingold, O.; and Vadhan, S. 2009. Computational Differential Privacy. In *Advances in Cryptology""CRYPTO 2009*, Lecture Notes in Computer Science. Springer.

Nasr, M.; Songi, S.; Thakurta, A.; Papernot, N.; and Carlin, N. 2021. Adversary Instantiation: Lower Bounds for Differentially Private Machine Learning. In *2021 IEEE Symposium on Security and Privacy (SP)*, 866–882.

Nie, W.; Guo, B.; Huang, Y.; Xiao, C.; Vahdat, A.; and Anandkumar, A. 2022. Diffusion Models for Adversarial Purification. In *International Conference on Machine Learning (ICML)*.

Orekondy, T.; Schiele, B.; and Fritz, M. 2019. Knockoff Nets: Stealing Functionality of Black-Box Models. In *CVPR*, 4954–4963.

Orekondy, T.; Schiele, B.; and Fritz, M. 2020. Prediction Poisoning: Towards Defenses Against DNN Model Stealing Attacks. In *ICLR*.

Pal, S.; Gupta, Y.; Shukla, A.; Kanade, A.; Shevade, S.; and Ganapathy, V. 2019. A framework for the extraction of Deep Neural Networks by leveraging public data. arXiv:1905.09165.

Pang, R.; Shen, H.; Zhang, X.; Ji, S.; Vorobeychik, Y.; Luo, X.; Liu, A.; and Wang, T. 2020. *A Tale of Evil Twins: Adversarial Inputs versus Poisoned Models*, 85–99. New York, NY, USA: Association for Computing Machinery. ISBN 9781450370899.

Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z. B.; and Swami, A. 2017. Practical black-box attacks against machine learning. In *ACM Symposium on Information, Computer and Communications Security*, 506–519. ACM.

Papernot, N.; Song, S.; Mironov, I.; Raghunathan, A.; Talwar, K.; and Erlingsson, U. 2018. Scalable Private Learning with PATE. In *International Conference on Learning Representations*.

Pearce, A. 2022. Can a Model Be Differentially Private and Fair? https://pair.withgoogle.com/explorables/private-and-fair/. Online; accessed 7 April 2022.

Quan, P.; Chakraborty, S.; Jeyakumar, J. V.; and Srivastava, M. 2022. On the amplification of security and privacy risks by post-hoc explanations in machine learning models. arXiv:2206.14004.

Quiring, E.; Arp, D.; and Rieck, K. 2018. Forgotten Siblings: Unifying Attacks on Machine Learning and Digital Watermarking. In *IEEE European Symposium on Security & Privacy*, 488–502.

Radiya-Dixit, E.; and Tramèr, F. 2021. Data Poisoning Won't Save You From Facial Recognition. *CoRR*, abs/2106.14851.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. KDD '16, 1135–1144. New York, NY, USA: Association for Computing Machinery. ISBN 9781450342322.

Sablayrolles, A.; Douze, M.; Schmid, C.; and Jégou, H. 2020. Radioactive data: tracing through training. *arXiv preprint arXiv:2002.00937*.

Smith, L. N.; and Topin, N. 2019. Super-convergence: very fast training of neural networks using large learning rates. *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*.

Song, L.; Shokri, R.; and Mittal, P. 2019. Privacy Risks of Securing Machine Learning Models against Adversarial Examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, 241–257. Association for Computing Machinery. ISBN 9781450367479.

Stallkamp, J.; Schlipsing, M.; Salmen, J.; and Igel, C. 2011. The German traffic sign recognition benchmark: a multi-class classification competition. In *IEEE International Joint Conference on Neural Networks*.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I. J.; and Fergus, R. 2014. Intriguing properties of neural networks. In Bengio, Y.; and LeCun, Y., eds., *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Szyller, S.; and Asokan, N. 2022. Conflicting Interactions Among Protection Mechanisms for Machine Learning Models. arXiv:2207.01991.

Szyller, S.; Atli, B. G.; Marchal, S.; and Asokan, N. 2021. *DAWN: Dynamic Adversarial Watermarking of Neural Networks*, 4417–4425. New York, NY, USA: Association for Computing Machinery. ISBN 9781450386517.

Tekgul, B. G. A.; and Asokan, N. 2022. On the Effectiveness of Dataset Watermarking. In *Proceedings of the 2022 ACM on International Workshop on Security and Privacy Analytics*. ACM.

Tramèr, F.; and Boneh, D. 2021. Differentially Private Learning Needs Better Features (or Much More Data). In *International Conference on Learning Representations (ICLR)*.

Tramèr, F.; Shokri, R.; San Joaquin, A.; Le, H.; Jagielski, M.; Hong, S.; and Carlini, N. 2022. Truth Serum: Poisoning Machine Learning Models to Reveal Their Secrets. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, 2779–2792. New York, NY, USA: Association for Computing Machinery. ISBN 9781450394505.

Tramèr, F.; Zhang, F.; Juels, A.; Reiter, M. K.; and Ristenpart, T. 2016. Stealing machine learning models via prediction apis. In *25th USENIX Security Symposium*, 601–618.

Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; and Madry, A. 2019. Robustness May Be at Odds with Accuracy. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Uchida, Y.; Nagai, Y.; Sakazawa, S.; and Satoh, S. 2017. Embedding watermarks into deep neural networks. In *ACM International Conference on Multimedia Retrieval*, 269–277. ACM.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv:1708.07747.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2017. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*.

Zhang, J.; Gu, Z.; Jang, J.; Wu, H.; Stoecklin, M. P.; Huang, H.; and Molloy, I. 2018. Protecting intellectual property of deep neural networks with watermarking. In *ACM Symposium on Information, Computer and Communications Security*, 159–172.