Cosine Model Watermarking against Ensemble Distillation

Laurent Charette[1]*, Lingyang Chu[2]*, Yizhou Chen[3], Jian Pei[3], Lanjun Wang[4], Yong Zhang[1]

[1] Huawei Technologies Canada, Burnaby, Canada,
[2] McMaster University, Hamilton, Canada,
[3] Simon Fraser University, Burnaby, Canada,
[4] Tianjin University, Tianjin, China,
{laurent.charette, yong.zhang3}@huawei.com, chul9@mcmaster.ca, yca375@sfu.ca, jpei@cs.sfu.ca,
wang.lanjun@outlook.com

Abstract

Many model watermarking methods have been developed to prevent valuable deployed commercial models from being stealthily stolen by model distillations. However, watermarks produced by most existing model watermarking methods can be easily evaded by ensemble distillation, because averaging the outputs of multiple ensembled models can significantly reduce or even erase the watermarks. In this paper, we focus on tackling the challenging task of defending against ensemble distillation. We propose a novel watermarking technique named CosWM to achieve outstanding model watermarking performance against ensemble distillation. CosWM is not only elegant in design, but also comes with desirable theoretical guarantees. Our extensive experiments on public data sets demonstrate the excellent performance of CosWM and its advantages over the state-of-the-art baselines.

Introduction

High-performance machine learning models are valuable assets of many large companies. These models are typically deployed as web services where the outputs of models can be queried using public application programming interfaces (APIs) (Ribeiro, Grolinger, and Capretz 2015).

A major risk of deploying models through APIs is that the deployed models are easy to steal (Tramèr et al. 2016). By querying the outputs of a deployed model through its API, many model distillation methods (Orekondy, Schiele, and Fritz 2019; Jagielski et al. 2019; Papernot et al. 2017) can be used to train a replicate model with comparable performance as the deployed model. Following the context of model distillation (Hinton, Vinyals, and Dean 2015), a replicate model is called a *student model*; and the deployed model is called a *teacher model*.

A model distillation process is often imperceptible because it queries APIs in the same way as a normal user (Orekondy, Schiele, and Fritz 2019). To protect teacher models from being stolen, one of the most effective ways is model watermarking (Szyller et al. 2019). The key idea is to embed a unique watermark in a teacher model, such that a student model distilled from the teacher model will also carry the same watermark. By checking the watermark, the owner of the teacher model can identify and reclaim ownership of a student model.

Some model watermarking methods have been proposed to identify student models produced by single model distillation (Szyller et al. 2019; Lukas, Zhang, and Kerschbaum 2019; Jia et al. 2021). However, as we discuss in the Related Works section, watermarks produced by these methods can be significantly weakened or even erased by *ensemble distillation* (Hinton, Vinyals, and Dean 2015), which uses the average of outputs queried from multiple different teacher models to train a replicate model.

Ensemble distillation has been well-demonstrated to be highly effective at compressing multiple large models into a small size student model with high performance (Buciluǎ, Caruana, and Niculescu-Mizil 2006; Ba and Caruana 2014). On the other hand, the effectiveness of ensemble distillation also poses a critical threat to the safety of deployed models.

As shown by extensive experimental results in the Experiments section, ensemble distillation not only generates student models with better prediction performance, but also significantly reduces the effectiveness of existing model watermarking methods in identifying student models. As a result, accurately identifying student models produced by ensemble distillation is an emergent task with top priority to protect teacher models from being stolen.

In this paper, we focus on defending against model distillation, and we successfully tackle this task by introducing a novel model watermarking method named CosWM. To the best of our knowledge, our method is the first model watermarking method with a theoretical guarantee to accurately identify student models produced by ensemble distillation. We make the following contributions.

First, we present a novel method named CosWM that embeds a watermark as a cosine signal within the output of a teacher model. Since the cosine signal is difficult to erase by averaging the outputs of multiple models, student models produced by ensemble distillation will still carry a strong watermark signal.

Second, under reasonable assumptions, we prove that a student model with a smaller training loss value will carry a stronger watermark signal. This means a student model will have to carry a stronger watermark in order to achieve a better performance. Therefore, a student model intending to weaken the watermark will not be able to achieve a good

[*] These authors contributed equally.

performance.

Third, we also design CosWM to allow each teacher model to embed a unique watermark by projecting the cosine signal in different directions in the high-dimensional feature space of the teacher model. In this way, owners of teacher models can independently identify their own watermarks from a student model.

Last, extensive experiment results demonstrate the outstanding performance of CosWM and its advantages over state-of-the-art methods.

Related Works

In this section, we introduce two major categories of model watermarking methods and discuss why these methods can be easily evaded by ensemble distillation.

The first category of methods (Uchida et al. 2017; Rouhani, Chen, and Koushanfar 2018; Adi et al. 2018; Zhang et al. 2018; Le Merrer, Pérez, and Trédan 2019) aim to protect machine learning models from being exactly copied. To produce a watermark, an effective idea is to embed a unique pattern by manipulating the values of parameters of the model to protect (Uchida et al. 2017; Rouhani, Chen, and Koushanfar 2018). If a protected model is exactly copied, the parameters of the copied model will carry the same pattern, which can be used as a watermark to identify the ownership of the copied model. Another idea to produce a watermark is to use backdoor images that trigger prescribed model predictions (Adi et al. 2018; Zhang et al. 2018; Le Merrer, Pérez, and Trédan 2019). The same backdoor image will trigger the same prescribed model prediction on an exactly copied model. Thus, the backdoor images are also effective in identifying exactly copied models.

The above methods focus on identifying exactly copied models, but they cannot be straight-forwardly extended to identify a student model produced by ensemble distillation (Hinton, Vinyals, and Dean 2015). Because the model parameters of the student model can be substantially different from the teacher model; and simple backdoor images of the teacher model are often not transferable to the student model, that is, the backdoor images may not trigger the prescribed model prediction on the student model (Lukas, Zhang, and Kerschbaum 2019).

The second category of methods aim to identify student models that are distilled from a single teacher model by single model distillation (Tramèr et al. 2016).

PRADA (Juuti et al. 2019) is designed to identify model distillations using synthetic queries that tend to be out-of-distribution. It analyzes the distribution of API queries and detects potential distillation activities when the distribution of queries deviates from the benign distribution. However, it is not effective in identifying the queries launched by ensemble distillations, because these queries are mostly natural queries that are not out-of-distribution.

Another typical idea is to produce transferable backdoor images that are likely to trigger the same prescribed model prediction on both the teacher model and the student model. DAWN (Szyller et al. 2019) generates transferable backdoor images by dynamically changing the outputs of the API of a protected teacher model on a small subset of querying images. Fingerprinting (Lukas, Zhang, and Kerschbaum 2019) makes backdoor images more transferable by finding common adversarial images that trigger the same adversarial prediction on a teacher model and any student model distilled from the teacher model. Entangled Watermarks (Jia et al. 2021) forces a teacher model to learn features for classifying data sampled from the legitimate data and watermarked data.

The above methods are effective in identifying student models produced by single model distillation, but they cannot accurately identify student models produced by ensemble distillation.

The reason is that, when an ensemble distillation averages the outputs of a watermarked teacher model and multiple other teacher models without a watermark, the prescribed model predictions of the watermarked teacher model will be weakened or even erased by the normal predictions of the other teacher models. If multiple watermarked teacher models are used for ensemble distillation, the prescribed model prediction of one teacher model can still be weakened or erased when averaged with the predictions of the other teacher models, because the prescribed model predictions of different teacher models are not consistent with each other.

The proposed CosWM method is substantially different from the other watermarking methods (Szyller et al. 2019; Lukas, Zhang, and Kerschbaum 2019; Jia et al. 2021). The watermark of CosWM is produced by coupling a cosine signal with the output function of a protected teacher model. As proved in Theorem 1 and demonstrated by extensive experiments in the Experiments section, when an ensemble distillation averages the outputs of multiple teacher models, the embedded cosine signal will persist. As a result, the watermarks produced by CosWM are highly effective in identifying student models produced by ensemble distillation.

Problem Definition

Ensemble methods, such as bagging (Bühlmann and Yu 2002), aggregate the probability predictions of all models in an ensemble to create a more accurate model on average. Ensemble models and distillation have been applied jointly since the first seminal studies on distillation (Buciluǎ, Caruana, and Niculescu-Mizil 2006; Ba and Caruana 2014; Hinton, Vinyals, and Dean 2015). These distillation methods use a combination of KL loss (Kullback and Leibler 1951) and cross-entropy loss (Bishop 2006) in the training process. Cross-entropy loss requires ground truth labels. Some recent state-of-the-art distillation methods (Vongkulbhisal, Vinayavekhin, and Visentini-Scarzanella 2019; Shen and Savvides 2020) only use KL loss, and thus can work without access to the ground truth values. This allows adversaries to replicate high performance models using ensemble model distillation and without ground truth labels.

Technically, let $\mathcal{R} = \{R_1, \dots, R_N\}$ be a set of N models trained to perform the same m-class classification task. Each model R_i outputs a probability prediction vector $R_i(\mathbf{x})$ on an input sample $\mathbf{x} \in \mathbb{R}^n$. An adversary may effectively build an ensemble model by querying an unlabeled data set $X^S =$

$\{\mathbf{x}^1, \dots, \mathbf{x}^L\}$ to each model $R_1, \dots, R_N$ and averaging the outputs, i.e., $\bar{\mathbf{q}}^l = \frac{1}{N}\sum_{i=1}^{N} R_i(\mathbf{x}^l)$ for $l = 1, \dots, L$. The averaged output $\bar{\mathbf{q}}^l$ can then be used as the soft pseudo labels to train a student model S.

We now formulate the task of watermarking against distillation from ensembles. Assume a model R to be protected and the watermarked version $w(R)$, where $w(\cdot)$ is a watermarking function. Denote by $h(R)$ a function measuring the accuracy of model R (on a given test data set) and by $g(R)$ a function measuring the strength of the watermark signal in model R.

Let S be an arbitrary model that is replicated from an ensemble distillation using $w(R)$ as a teacher. S may use some additional other teacher models. Let S' be another arbitrary model that is replicated from an ensemble distillation where $w(R)$ is not a teacher. The **task of model watermarking** is to design watermarking function $w(\cdot)$ such that it meets two requirements. First, the accuracy loss in watermarking is within a specified tolerance range $\alpha > 0$, i.e., $h(R) - h(w(R)) \leq \alpha$. Second, the watermark signal model in S is stronger than that in S', i.e., $g(S) > g(S')$.

CosWM

In this section, we present our watermarking method CosWM. We first explain the intuition of our method. Then, we develop our watermarking framework to embed a periodic signal to a teacher model. Third, we describe how the embedded signal can be extracted from a student model learned using a watermarked teacher model. Next, we provide strong theoretical results to justify our design. Last, we discuss possible extensions to ensembles containing multiple watermarked models.

Intuitions

The main idea of CosWM is to introduce a perturbation to the output of a teacher model. This perturbation is transferred onto a student model distilled from the teacher model and remains detectable with access to the output of the student model.

The idea is illustrated in Figure 1. Let R be a model to be watermarked and $\mathbf{q} = R(\mathbf{x})$ be the output of the model R on input $\mathbf{x}$. We also convert $\mathbf{x}$ into a number $p(\mathbf{x})$ in a finite range. We can select a class i^* and use the model prediction output $\mathbf{q}_{i^*}$ on the class to load our watermark. Let $\mathbf{q}_{i^*}(\mathbf{x})$ be the i^*-th element of vector $R(\mathbf{x})$. Figure 1(a) plots $(\mathbf{q}_{i^*}(\mathbf{x}), p(\mathbf{x}))$ without any added watermark signal. After adding a periodic perturbation $\phi(p(\mathbf{x}))$ of frequency f_w to the output of R, the new output $\mathbf{q}_{i^*}(\mathbf{x})$ demonstrates some oscillations, as shown in Figure 1(b). We keep the perturbation small enough so that the model predictions are mostly unaffected and the effect of the watermark on the model's performance is minimal.

A student model trying to replicate the behavior of the teacher model passively features a similar oscillation at the same frequency f_w. In addition, even with the averaging effect of an ensemble of teacher models on the outputs, the periodic signal should still be present in some form. Since the averaging is linear, the amplitude is diminished by a factor of the number of the ensemble models as shown in Figure 1(c). By applying a Fourier transform, the perturbation can be re-identified by the presence of a peak in the power spectrum at the frequency f_w as shown in Figure 1(d).

Embedding Watermarks to a Teacher Model

mance of the teacher model.

Normally, an output $\mathbf{q}$ of a model R on a given data point $\mathbf{x}$ is calculated from the softmax of the logits $\mathbf{z} \in \mathbb{R}^m$, i.e.,

$$\mathbf{q}_i = \frac{e^{\mathbf{z}_i}}{\sum_{j=1}^{m} e^{\mathbf{z}_j}}, \text{ for } i = 1, ..., m, \tag{1}$$

where $\mathbf{z}$ is a function of $\mathbf{x}$, and $\mathbf{q}_i$ is the i-th element of vector $\mathbf{q}$. As a result, the output $\mathbf{q}$ has the following property.

Property 1. *Let* $\mathbf{q}$ *be a softmax of the logit output* $\mathbf{z}$ *of a model* R*. Then,*

1. $0 \leq \mathbf{q}_i \leq 1$ *for* $i = 1, \dots, m$*,*
2. $\sum_{i=1}^{m} \mathbf{q}_i = 1$.

We want to substitute $\mathbf{q}$ in the model inference by a modified output $\hat{\mathbf{q}} \in \mathbb{R}^m$ which features the periodic signal and satisfies Property 1. However, only modifying $\mathbf{q}$ in the model inference by itself may degrade the performance of the model, and the loss in accuracy cannot be bounded. In order to mitigate this effect, we also use the modified output $\hat{\mathbf{q}}$ in training R. That is, we use $\hat{\mathbf{q}}$ to compute cross entropy loss in the training process.

To embed watermarks, we first define a watermark key K that consists of a target class $i^* \in \{1, \dots, m\}$, an angular frequency $f_w \in \mathbb{R}$, and a random unit projection vector $\mathbf{v} \in \mathbb{R}^n$, i.e., $K = (i^*, f_w, \mathbf{v})$. Using K, we define a periodic signal function

$$\mathbf{a}_i(\mathbf{x}) = \begin{cases} \cos\left(f_w p(\mathbf{x})\right), & \text{if } i = i^*, \\ \cos\left(f_w p(\mathbf{x}) + \pi\right), & \text{otherwise}, \end{cases} \tag{2}$$

for $i \in \{1, \dots, m\}$, where

$$p(\mathbf{x}) = \mathbf{v}^\intercal \mathbf{x}. \tag{3}$$

We consider single-frequency signals in this work and we plan to study watermark signals with mixed frequencies in our future work. We adopt linear projections since they are simple one-dimensional functions of input data and can easily form a high-dimensional function space. This leads to a large-dimensional space to select $\mathbf{v}$ from, and generally little interference between two arbitrary choices of $\mathbf{v}$. As a consequence, we get a large choice of possible watermarks, and each watermark is concealed to adversaries trying to source back the signal with arbitrary projections.

We inject the periodic signal into output $\mathbf{q}$ to obtain $\hat{\mathbf{q}}$ as follows. For $i \in \{1, \dots, m\}$,

$$\hat{\mathbf{q}}_i = \begin{cases} \dfrac{\mathbf{q}_i + \varepsilon(1 + \mathbf{a}_i(\mathbf{x}))}{1 + 2\varepsilon}, & \text{if } i = i^*, \\ \dfrac{\mathbf{q}_i + \frac{\varepsilon(1 + \mathbf{a}_i(\mathbf{x}))}{m-1}}{1 + 2\varepsilon}, & \text{otherwise}, \end{cases} \tag{4}$$

where ε is an amplitude component for the watermark periodic signal. As proved in a technical appendix (Charette

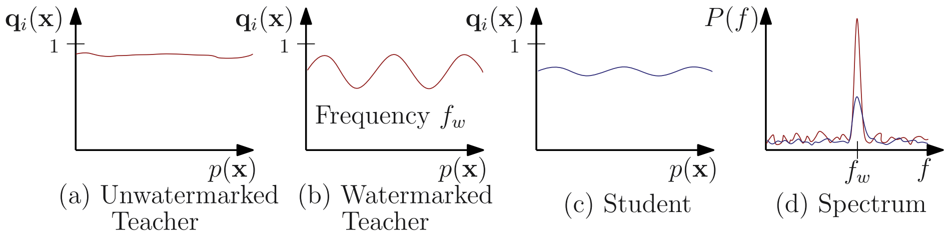


Figure 1: The idea of CosWM, where $\mathbf{q}_i(\mathbf{x})$ is an model output component for image $\mathbf{x}$, $p(\mathbf{x})$ is a projection as described in equation (3), f and $P(f)$ are frequency and power spectrum values of a $p(\mathbf{x})$-$\mathbf{q}_i(\mathbf{x})$ graph.

et al. 2022), the modified output $\hat{\mathbf{q}}$ still satisfies both requirements in Property 1. Therefore, it is natural to replace $\mathbf{q}$ by $\hat{\mathbf{q}}$ in inference.

Nevertheless, if we only modify $\mathbf{q}$ into $\hat{\mathbf{q}}$ in inference, the inference performance can be degraded by this perturbation. Since the modified output satisfies Property 1, we can use it in training as well to compensate for the potential performance drop. To do that, we directly replace $\mathbf{q}$ by $\hat{\mathbf{q}}$ in the cross-entropy loss function. Specifically, for a data point $\mathbf{x}$ with one-hot encoding true label $\mathbf{y}^t \in \mathbb{R}^m$, the cross-entropy loss during training can be replaced by

$$L_{CE,wm} = -\sum_{j=1}^{m} \mathbf{y}_j^t \log\left(\hat{\mathbf{q}}_j\right). \tag{5}$$

The model R_w trained as such carries the watermark. By directly modifying the output, we ensure that the signal is present in every output, even for input data not used during training. This generally results in a clear signal function in the output of the teacher model R_w that is harder to conceal by noise caused by distillation training or by dampening due to ensemble averaging.

Extracting Signals in Student Models

Let S be a student model that is suspected of being distilled from a watermarked model R_w or multiple ensembled teacher models including R_w. To extract the possible watermark from S, we need to query S with a sample of student training data $\widetilde{X}^S = \{\mathbf{x}^1, \ldots, \mathbf{x}^{\widetilde{L}}\}$. According to (Szyller et al. 2019), the owner of a teacher model can easily obtain $\widetilde{X}^S$ because the owner may store any query input sent by an adversary to the API.

Let the output of model S on the input data $\widetilde{X}^S$ be $\widetilde{Q}^S = \{\mathbf{q}^1, \ldots, \mathbf{q}^{\widetilde{L}}\}$, where $\mathbf{q}^l \in \mathbb{R}^m$ for $l = 1, \ldots, \widetilde{L}$. For every pair $(\mathbf{x}^l, \mathbf{q}^l)$, we extract a pair of results $(\mathbf{p}_l, \mathbf{q}^l_{i^*})$, where $\mathbf{p}_l = \mathbf{v}^\intercal \mathbf{x}^l$ as per Equation (3), $\mathbf{v}$ is in the watermark key of R_w and i^* is the target class when embedding watermarks to R_w. We filter out the pairs $(\mathbf{p}_l, \mathbf{q}^l_{i^*})$ with $\mathbf{q}^l_{i^*} \leq q_{min}$ in order to remove outputs with low confidence, where the threshold value q_{min} is a constant parameter of the extraction process. The surviving pairs are re-indexed into a set $\widetilde{D}^S = \{(\mathbf{p}_l, \mathbf{q}^l_{i^*})\}_{l=1,\ldots,\widetilde{M}}$, where $\widetilde{M}$ is the number of remaining pairs. These surviving pairs $(\mathbf{p}_l, \mathbf{q}^l_{i^*}) \in \widetilde{D}^S$ are then used to compute the Fourier power spectrum, for evenly spaced frequency values spanning a large interval containing the frequency f.

To approximate the power spectrum, we use the Lomb-Scargle periodogram method (Scargle 1982), which allows one to approximate the power spectrum $P(f)$ at frequency f using unevenly sampled data. We give the formal definition of $P(f)$ in the next section when we analyze the theoretical bounds of $P(f)$. Due to noise in the model outputs, it is preferable to have more sample pairs in $\widetilde{D}^S$ than the few required to detect a pure cosine signal. In our experience, we reliably detect a watermark signal using 100 pairs for a single watermarked model and 1,000 pairs for an 8-model ensemble.

To measure the signal strength of the watermark, we define a maximum frequency F and a window $\left[f_w - \frac{\delta}{2}, f_w + \frac{\delta}{2}\right]$, where δ is a parameter for the width of the window and f_w is the frequency in watermark key of R_w. Then, we calculate P_{signal} and P_{noise} by averaging spectrum values $P(f)$ on frequencies inside and outside the window, i.e., $P_{signal} = \frac{1}{\delta}\int_{f_w-\frac{\delta}{2}}^{f_w+\frac{\delta}{2}} P(f)df$ and $P_{noise} = \frac{1}{F-\delta}\left[\int_0^{f_w-\frac{\delta}{2}} P(f)df + \int_{f_w+\frac{\delta}{2}}^{F} P(f)df\right]$, respectively. We use the signal-to-noise ratio to measure the signal strength of the watermark, i.e.,

$$P_{snr} = P_{signal}/P_{noise}. \tag{6}$$

The algorithm is summarized in Algorithm 1.

Theoretical Analysis

Here, we analyze the signal strength of P_{signal} and P_{noise} and provide theoretical bounds for the power spectrum $P(f)$. Let us first recall two results from (Scargle 1982).

Given a paired data set $D = \{(\mathbf{a}^l, \mathbf{b}_l) \in \mathbb{R}^n \times \mathbb{R}, l = 1, \ldots, L\}$, an angular frequency f, a projection vector $\mathbf{v}$, and a sinusoidal function $s(\mathbf{x}) = \alpha + \beta\cos(f\mathbf{v}^\intercal\mathbf{x} + \gamma)$, where α, β and γ are the parameters of $s(\mathbf{x})$, the *best fitting points* $\mathbf{s}^*(D)$ for this paired data are

$$[\mathbf{s}^*(D)]_l = \alpha^* + \beta^*\cos(f\mathbf{v}^\intercal\mathbf{a}^l + \gamma^*) \text{ for } l = 1, \ldots, L, \tag{7}$$

where the parameters α^*, β^*, γ^* minimize the square error $\chi_f^2(D) = \sum_{l=1}^{L}[\mathbf{b}_l - s(\mathbf{a}^l)]^2$.

Moreover, given a paired data set $D = \{(\mathbf{a}^l, \mathbf{b}_l) \in \mathbb{R}^n \times \mathbb{R}, l = 1, \ldots, L\}$ and a frequency f, the *unnormalized Lomb-Scargle periodogram* can be written as

$$P_D(f) = \frac{1}{2}\left[\chi_0^2(D) - \chi_f^2(D)\right], \tag{8}$$

Algorithm 1: Extracting signal in a model

```
Inputs : A suspected model $S$,
         Samples $\widetilde{X}_S$ of the training data of $S$,
         A watermark key $K = (i^*, f_w, \mathbf{v})$ of the
         watermarked model $R_w$,
         Filtering threshold value $q_{min}$.
Output: Signal strength.
Query $\widetilde{X}_S$ to $S$ and obtain outputs
    $\widetilde{Q}^S = \{\mathbf{q}^1, \dots, \mathbf{q}^{\widetilde{L}}\}$.
Compute projections $\mathbf{p}_l = \mathbf{v}^\intercal \cdot \mathbf{x}^l$, for $l = 1, \dots, \widetilde{L}$.
Filter out outputs where $\mathbf{q}^l_{i^*} \leq q_{min}$, remaining pairs
    form the set $\widetilde{D}^S = \{(\mathbf{p}_l, \mathbf{q}^l_{i^*})\}_{l=1,\dots,\widetilde{M}}$.
Compute the Lomb-Scargle periodogram from the
    pairs $(\mathbf{p}_l, \mathbf{q}^l_{i^*})$ in $\widetilde{D}^S$.
Compute $P_{signal}$ and $P_{noise}$ by averaging spectrum
    values on frequencies inside and outside the
    window $\left[f_w - \frac{\delta}{2}, f_w + \frac{\delta}{2}\right]$, respectively.
Compute $P_{snr} = P_{signal} / P_{noise}$.
return Signal strength $P_{snr}$.
```

where $\chi^2_0(D)$ is the square error of the best constant fit to $\mathbf{b}_1, \dots, \mathbf{b}_L$.

Now we are ready to give a theoretical bound on $P_D(f)$ for the output of the student model.

Theorem 1. *Suppose there are N teacher models $R_1, \dots, R_N$. Without loss of generality, let R_1 be a watermarked teacher model with watermark key $K = (i^*, f_w, \mathbf{v})$, and S a student model distilled from an ensemble model of $R_1, \dots, R_N$ on the student training data X^S. Let $\widetilde{X}^S = \{\mathbf{x}^1, \dots, \mathbf{x}^L\}$ be a sample subset of X^S. Let $\hat{\mathbf{q}}^l = R_1(\mathbf{x}^l)$ be the output of model R_1, $\widetilde{\mathbf{q}}^l = \frac{1}{N-1}\sum_{i=2}^N R_i(\mathbf{x}^l)$ be the output of the ensemble model of $R_2, \dots, R_N$, $\bar{\mathbf{q}}^l = \frac{1}{N}(\hat{\mathbf{q}}^l + (N-1)\widetilde{\mathbf{q}}^l)$ be the output of the ensemble model of $R_1, \dots, R_N$, and $\mathbf{q}^l = S(\mathbf{x}^l)$ the output of S for the training data point $\mathbf{x}^l$. Let $\hat{D} = \{(\mathbf{x}^l, \hat{\mathbf{q}}^l_{i^*}), l = 1, \dots, L\}$, $\widetilde{D} = \{(\mathbf{x}^l, \widetilde{\mathbf{q}}^l_{i^*}), l = 1, \dots, L\}$, $\bar{D} = \{(\mathbf{x}^l, \bar{\mathbf{q}}^l_{i^*}), l = 1, \dots, L\}$ and $D = \{(\mathbf{x}^l, \mathbf{q}^l_{i^*}), l = 1, \dots, L\}$ be paired data sets. Then, the unnormalized Lomb-Scargle periodogram value $P_D(f)$ for the student output at angular frequency f has the following bounds*

$$\frac{1}{2}\left[\chi^2_0(D) - \tau_1 + L_{se}\right] \geq P_D(f) \geq \frac{1}{2}\left[\chi^2_0(D) - \tau_2 - L_{se}\right], \quad (9)$$

where

$$\tau_1 = \chi^2_f(\bar{D}), \tau_2 = \frac{1}{N^2}\chi^2_f(\hat{D}) + \left(\frac{N-1}{N}\right)^2 \chi^2_f(\widetilde{D}),$$

$$L_{se} = \sum_{l=1}^{L} \left(\bar{\mathbf{q}}^l_{i^*} - \mathbf{q}^l_{i^*}\right)^2.$$

Proof. See technical appendix (Charette et al. 2022).

Theorem 1 provides several insights.

Remark 1. *When a student model is well trained by a teacher model, L_{se} is generally small.*

Remark 2. *Consider the case where $f = f_w$. If we choose our sample $\widetilde{X}^S$ with high confidence output scores on the i^*-th class, for example by filtering as described in Algorithm 1, $\chi^2_{f_w}(\hat{D})$ should be small enough to be negligible by our watermark design in the teacher model. We then discuss the following two cases.*

***Case I**: When $N = 1$, there is only one watermarked teacher to distill a student model. Then, after neglecting $\chi^2_{f_w}(\hat{D})$, the left inequality of Equation (9) becomes*

$$P_D(f_w) \geq \frac{1}{2}\left[\chi^2_0(D) - L_{se}\right].$$

This implies that we can observe a significant signal for the output of the student model at frequency f_w when the output of the student model is close to that of the teacher model.

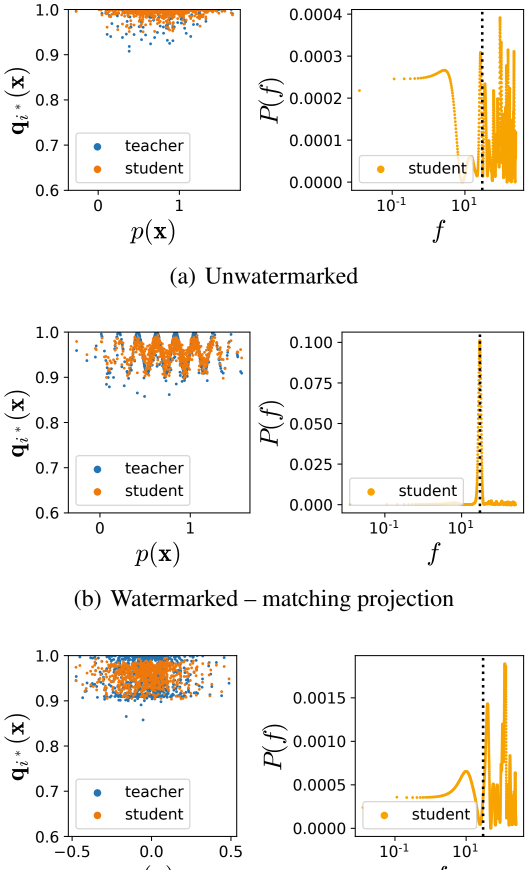
***Case II**: When $N \neq 1$, since there is no sinusoidal signal in $\widetilde{\mathbf{q}}^l_{i^*}$, for $l = 1, \dots, L$, and the sinusoidal signal in $\bar{\mathbf{q}}^l_{i^*}$, for $l = 1, \dots, L$ is, proportional to $\frac{\varepsilon}{N}$, τ_2 increases as N increases. However, to keep the watermark signal significant in the output of the student model, one can increase the watermark signal amplitude ε in the teacher model R_1, which indirectly increases $\chi^2_0(D)$. This is due to the fact that if ε increases, $\chi^2_0(\hat{D})$ also increases. Since L_{se} is small when a student model is well trained by the teacher model, $\chi^2_0(D)$ increases as well. This implies that we can detect the watermark in the output of the student model at frequency f_w by increasing the watermark signal in the teacher model R_1 when N is large. We validate this observation in the Experiments section.*

Remark 3. *When $f \neq f_w$, since there is no sinusoidal signal at frequency f in $\hat{\mathbf{q}}^l_{i^*}$, $\widetilde{\mathbf{q}}^l_{i^*}$, and $\bar{\mathbf{q}}^l_{i^*}$ for $l = 1, \dots, L$, $\chi^2_f(\hat{D})$, $\chi^2_f(\widetilde{D})$ and $\chi^2_f(\bar{D})$ are generally large. Thus, the values of both sides of the inequality in Equation (9) are small, which implies that there is no sinusoidal signal for the output of the student model at frequency $f \neq f_w$.*

Multiple Watermarked Teacher Models

Consider a student model trained on the output of an ensemble model that consists of two or more teacher models with watermarks. Can those watermarks be detected in the student model?

We argue that it should be possible to extract each signal if the watermark keys are different. The reason for this is that a signal embedded using watermark key $K_1 = (i_1, f_1, \mathbf{v}^1)$ appears as noise for an independent watermark $K_2 = (i_2, f_2, \mathbf{v}^2)$. Since noise has low overall spectrum values, the resulting ensemble output spectrum will be similar to an ensemble with only one watermarked model. Therefore, each signal should be detectable using its respective key. This highlights the importance that $\mathbf{v}$ should preferably be a high dimensional vector that can provide more independent random choices for the watermark key K.



(a) Unwatermarked

(b) Watermarked – matching projection

(c) Watermarked – non-matching projection

Figure 2: A case study of the watermarking mechanism in CosWM. The black vertical line indicates $f = 30.0$. In each subgraph, the left plots the target class output $\mathbf{q}_i(\mathbf{x})$ of the teacher model and the student model as a function of projection value $p(\mathbf{x})$, and the right plots the power spectrum value $P(f)$ for the output of the student model as a function of frequency f.

Experiments

In this section, we evaluate the performance of CosWM on the model watermarking task. We first describe the settings and data sets. Then we present a case study to demonstrate the working process of CosWM. We compare the performance of all the methods in two scenarios. We analyze the effect of the amplitude parameter ε and the signal frequency parameter f_w on the performance of CosWM in a technical appendix (Charette et al. 2022), where we also analyze the effects of using ground truth labels during distillation.

Experiment Settings and Data Sets

We compare CosWM with two state-of-the-art methods, DAWN (Szyller et al. 2019) and Fingerprinting (Lukas, Zhang, and Kerschbaum 2019). We implement CosWM and replicate DAWN in PyTorch 1.3. The Fingerprinting code is provided by the authors of the corresponding paper (Lukas, Zhang, and Kerschbaum 2019) and is implemented in Keras using a TensorFlow v2.1 backend. All the experiments are conducted on Dell Alienware with Intel(R) Core(TM) i9-9980XE CPU, 128G memory, NVIDIA 1080Ti, and Ubuntu 16.04.

We conduct all the experiments using two public data sets, FMNIST (Xiao, Rasul, and Vollgraf 2017), and CIFAR10 (Krizhevsky 2009). We report the experimental results on CIFAR10 in this section and the results on FMNIST in a technical appendix (Charette et al. 2022).

The CIFAR10 data set contains natural images in 10 classes. It consists of a training set of 50,000 examples and a test set of 10,000 examples. We partition all the training examples randomly into two halves, with use one half for training the teacher models and the other half for distilling the student models. For each data set the feature vectors are normalized to the range $[0, 1]$.

In all experiments, we use ResNet18 (He et al. 2016). All models are trained or distilled for 100 epochs to guarantee convergence. The models with the best testing accuracy during training/distillation are retained.

A Case Study

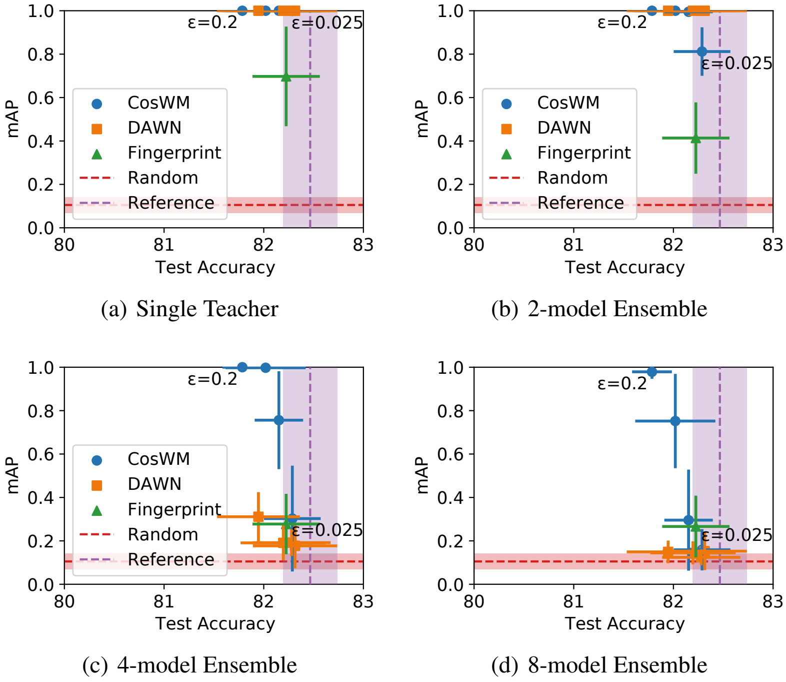
We conduct a case study to demonstrate the watermarking mechanism in CosWM. We first train one watermarked teacher model and one non-watermarked teacher model using the first half of the training data, and then distill one student model from each teacher model using the second half of the training data. To train the watermarked teacher model, we set the signal amplitude $\varepsilon = 0.05$ and the watermark key $K = (f_w, i^*, \mathbf{v}^0)$ with $f_w = 30.0$, $i^* = 0$ and $\mathbf{v}^0$ a unit random vector. For extraction, we set q_{min} to be the first quartile of all $\mathbf{q}_{i^*}(\mathbf{x})$ values for 1,000 randomly selected training examples whose ground truth is class i^*. Code for this case study is available online [1].

We analyze the output of the teacher models and the student models for both the time and frequency domains in Figure 2 for three cases. In Figures 2(a), (b), and (c) for the three cases, we plot $\mathbf{q}_{i^*}(\mathbf{x})$ vs. $p(\mathbf{x})$ in time domain for both the teacher model and the student model in the left graph, and $P(f)$ vs. f in the frequency domain for the student model in the right graph.

In the first case, Figure 2(a) shows the results for the non-watermarked teacher model and the student model. There is no sinusoidal signal in the output for either the teacher model or the student model at frequency f_w with projection vector $\mathbf{v}^0$.

In the second case, Figure 2(b) shows the results for the watermarked teacher model and the student model. The accuracy loss of the watermarked teacher model is within 1% of the accuracy of the unwatermarked teacher model in Figure 2(a). We extract the output of the watermarked teacher model and the student model using the watermark key K. The output of the teacher follows an almost perfect sinusoidal function and the output of the student model is close

[1] https://developer.huaweicloud.com/develop/aigallery/notebook/detail?id=2d937a91-1692-4f88-94ca-82e1ae8d4d79



(a) Single Teacher

(b) 2-model Ensemble

(c) 4-model Ensemble

(d) 8-model Ensemble

Figure 3: mAP of CosWM, DAWN, and Fingerprinting under different parameter values as a function of accuracy of the watermarked model. Each watermarked model is part of an ensemble of teacher models and is the only watermarked model within that ensemble.

to the output of the teacher model in the time domain. In the frequency domain, the student model has a very prominent peak at frequency f_w. This observation validates Remark 2 in the Theoretical Analysis section when $N = 1$.

In the last case, we replace $\mathbf{v}^0$ by a different unit random vector $\mathbf{v}^1$ in the watermark key K to extract the output of the watermarked teacher model and the student model. The results are shown in Figure 2(c). The output of both the teacher model and the student model is almost indiscernible from noise. Thus, there is no significant peak for the output of the student model in the power spectrum at frequency f_w. This observation validates Remark 3 in the Theoretical Analysis section.

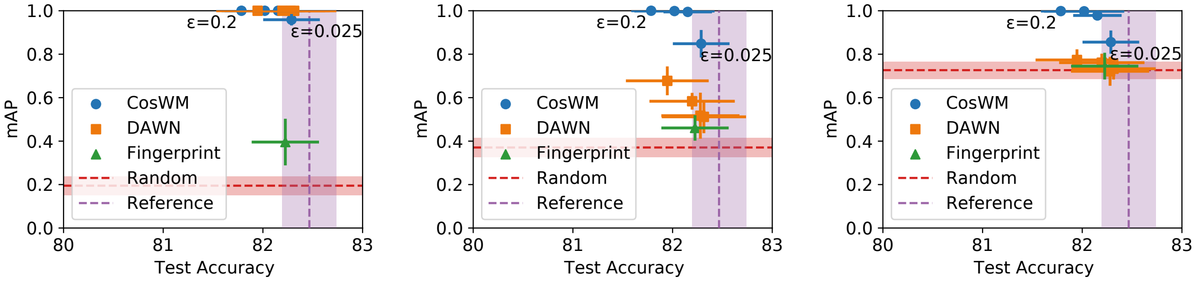
Protection with a Single Watermarked Teacher

To compare CosWM with DAWN and Fingerprinting in protecting watermarked teacher models, we set up a series of ranking tasks with different ensemble size N. In each ranking task, we have 10 student models distilled from the watermarked teacher model (positive student models) and 100 student models not distilled from the watermarked teacher model (negative students). For different methods, we use their own watermark signal strength values to rank those 110 students. Specifically, we use P_{snr} defined in Equation (6) for CosWM, the fraction of matching watermark predictions for DAWN, and the fraction of matching fingerprint predictions for Fingerprinting. To evaluate the performance of all three methods, we compute the average precision (AP) for each ranking task and repeat each task for all 10 watermarked models to calculate the mean average precision (mAP) and its standard deviation.

For all three methods, we use the first half of the training data to train 10 unwatermarked teacher models with different initialization and 10 teacher models with different watermark or fingerprint keys. We tune the parameters to make sure that the accuracy losses of all watermarked teacher models are within 1% of the averaged accuracy of the unwatermarked teacher models. To create a ranking task with 110 student models, for every watermarked teacher model we assemble it with $N - 1$ randomly selected unwatermarked teacher models to distill 10 student models with different initialization. In addition, we train 10 independent student models with ground truth labels and different initialization. The above process gives us 10 positive and 100 negative student models for each watermarked teacher model.

For CosWM, all watermarked teacher models have the same frequency $f_w = 30.0$ and target class $i^* = 0$, but have 10 different unit random projection vectors $\mathbf{v^0}, \dots, \mathbf{v^9}$. We set q_{min} to the median of all $\mathbf{q}_{i^*}$ values and vary the watermark amplitude ε in 0.025, 0.05, 0.1, and 0.2. For DAWN, we vary the fraction of watermarked input τ in 0.0005, 0.001, 0.002, and 0.005. For Fingerprinting, we generate one single set of fingerprint input per teacher model using parameter $\varepsilon_{fp} = 0.095$, which results in a large enough set of fingerprints with the best conferrability score. During extraction, all fingerprint input and labels are tested on a model to compute the fingerprint strength value for ranking.

Figure 3 shows the results on the CIFAR10 data set for



(a) 2-model Ensemble (b) 4-model Ensemble (c) 8-model Ensemble

Figure 4: mAP of CosWM, DAWN and Fingerprinting under different parameter values as a function of accuracy of the watermarked model. Each watermarked model is part of an ensemble of teacher models where every model is watermarked.

different ensemble size values $N = 1, 2, 4, 8$. In this figure, we plot the mAP scores as a function of the average teacher model accuracy. As a baseline, we add a *Random* method that ranks all student models randomly, whose mAP and standard deviation is represented by the horizontal red dashed line. The vertical purple dashed line shows the average and standard deviation of the accuracy of the unwatermarked teacher models.

As shown for both CosWM and DAWN in Figure 3, a stronger watermark will negatively affect model performance. A model owner must consider this effect when tuning the watermark.

When the ensemble size is small, i.e., $N = 1, 2$, the best mAP of CosWM and DAWN are generally comparable, and are both significantly larger than that of Fingerprinting, as shown in Figures 3(a) and 3(b). When the ensemble size is larger, i.e., $N = 4, 8$, the best mAP of CosWM is significantly larger than that of DAWN and Fingerprinting, whose watermarked model is consistently outnumbered, as shown in Figures 3(c) and 3(d). This superior performance of CosWM is due to our watermark signal design that is robust to ensemble distillation. When the ensemble size increases, CosWM needs a larger ε to keep mAP high. This confirms the discussion in Remark 2 in the Theoretical Analysis section.

In addition, we observe a trade-off between ensemble size and mAP when choosing different signal amplitude ε for CosWM and different fraction τ of watermarked input for DAWN. We analyze the effect of the amplitude parameter ε in more details in the technical appendix (Charette et al. 2022).

Protection with Multiple Watermarked Teachers

We compare CosWM with DAWN and Fingerprinting in assembling only watermarked teacher models to train a student model by undertaking another series of ranking tasks for different ensemble sizes N. We train 10 watermarked teacher models as described in the previous single watermark experiment, and assemble 10 sets of teacher models for each ensemble size in a round-robin manner. The training of all other models and watermark settings in this experiment remain exactly the same as described in the single watermark experiment. As a result, in an N-ensemble teacher model experiment, each ranking task associated to a teacher model has $10N$ positive and $110 - 10N$ negative student models.

Figure 4 shows the results on the CIFAR10 data set for different ensemble size values, i.e., $N = 2, 4, 8$. It is plotted in the same way as in Figure 3, described in the previous section. Similar to the previous experiments, we also add the *Random* baseline to provide a lower bound performance.

The accuracy losses of all watermarked models are within 1% of the average accuracy of all unwatermarked teacher models. When the ensemble size is small, i.e., $N = 2$, the best mAP of CosWM and DAWN are generally comparable to each other, and are both significantly larger than that of Fingerprinting, as shown in Figure 4(a). However, CosWM has a significantly higher best mAP for larger ensemble sizes, i.e., $N = 4, 6, 8$, as shown in Figures 4(b) and (c). This shows that CosWM watermarks are generally unaffected by other watermarks in a teacher ensemble and confirms the possibility of detecting watermarks if the ensemble features multiple watermarked teacher models as discussed in the Theoretical Analysis section.

We also observe a similar trade-off between ensemble size and mAP when choosing different signal amplitude ε for CosWM and different fraction τ of watermarked input for DAWN. This is further analyzed in a technical appendix (Charette et al. 2022).

Conclusion

In this paper, we tackle a novel problem of protecting neural network models against ensemble distillation. We propose CosWM, an effective method relying on a signal embedded into all output of a watermarked model, and therefore transferring the signal to training data for student models. We prove that the embedded signal in CosWM is strong in a well-trained student model by providing lower and upper bounds on the watermark strength metric. In addition, CosWM can be extended to identify student models distilled from an ensemble featuring multiple watermarked models. Our extensive experiments demonstrate the superior performance of CosWM in providing models defense from ensemble distillation.

References

Adi, Y.; Baum, C.; Cisse, M.; Pinkas, B.; and Keshet, J. 2018. Turning Your Weakness Into a Strength: Watermarking Deep Neural Networks by Backdooring. *arXiv preprint arXiv:1802.04633*.

Ba, J.; and Caruana, R. 2014. Do Deep Nets Really Need to be Deep? In *Advances in Neural Information Processing Systems*, volume 27.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer New York.

Buciluă, C.; Caruana, R.; and Niculescu-Mizil, A. 2006. Model Compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 535–541.

Bühlmann, P.; and Yu, B. 2002. Analyzing bagging. *The Annals of Statistics*, 30(4): 927 – 961.

Charette, L.; Chu, L.; Chen, Y.; Pei, J.; and Zhang, Y. 2022. Cosine Model Watermarking Against Ensemble Distillation. *arXiv preprint arXiv:2203.02777*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.

Jagielski, M.; Carlini, N.; Berthelot, D.; Kurakin, A.; and Papernot, N. 2019. High Accuracy and High Fidelity Extraction of Neural Networks. *arXiv preprint arXiv:1909.01838*.

Jia, H.; Choquette-Choo, C. A.; Chandrasekaran, V.; and Papernot, N. 2021. Entangled Watermarks as a Defense against Model Extraction. *arXiv preprint arXiv:2002.12200*.

Juuti, M.; Szyller, S.; Marchal, S.; and Asokan, N. 2019. PRADA: Protecting Against DNN Model Stealing Attacks. *IEEE European Symposium on Security and Privacy*.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Kullback, S.; and Leibler, R. A. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1): 79 – 86.

Le Merrer, E.; Pérez, P.; and Trédan, G. 2019. Adversarial frontier stitching for remote neural network watermarking. *Neural Computing and Applications*, 32(13): 9233–9244.

Lukas, N.; Zhang, Y.; and Kerschbaum, F. 2019. Deep Neural Network Fingerprinting by Conferrable Adversarial Examples. *arXiv preprint arXiv:1912.00888*.

Orekondy, T.; Schiele, B.; and Fritz, M. 2019. Knockoff Nets: Stealing Functionality of Black-Box Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z. B.; and Swami, A. 2017. Practical Black-Box Attacks against Machine Learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 506–519.

Ribeiro, M.; Grolinger, K.; and Capretz, M. A. M. 2015. MLaaS: Machine Learning as a Service. In *IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 896–902.

Rouhani, B. D.; Chen, H.; and Koushanfar, F. 2018. DeepSigns: A Generic Watermarking Framework for IP Protection of Deep Learning Models. *arXiv preprint arXiv:1804.00750*.

Scargle, J. D. 1982. Studies in astronomical time series analysis. II-Statistical aspects of spectral analysis of unevenly spaced data. *The Astrophysical Journal*, 263: 835–853.

Shen, Z.; and Savvides, M. 2020. MEAL V2: Boosting Vanilla ResNet-50 to 80%+ Top-1 Accuracy on ImageNet without Tricks. *arXiv preprint arXiv:2009.08453*.

Szyller, S.; Atli, B. G.; Marchal, S.; and Asokan, N. 2019. DAWN: Dynamic Adversarial Watermarking of Neural Networks. *arXiv preprint arXiv:1906.00830*.

Tramèr, F.; Zhang, F.; Juels, A.; Reiter, M. K.; and Ristenpart, T. 2016. Stealing Machine Learning Models via Prediction APIs. *arXiv preprint arXiv:1609.02943*.

Uchida, Y.; Nagai, Y.; Sakazawa, S.; and Satoh, S. 2017. Embedding Watermarks into Deep Neural Networks. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, 269–277.

Vongkulbhisal, J.; Vinayavekhin, P.; and Visentini-Scarzanella, M. 2019. Unifying heterogeneous classifiers with distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3175–3184.

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

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 *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, 159–172.