Устойчивые методы маркировки нейросетей для выявления несанкционированного доступа

Олег Рогов, к.ф.-м.н.

Руководитель группы
«Доверенные и Безопасные Интеллектуальные Системы» AIRI

Содержание

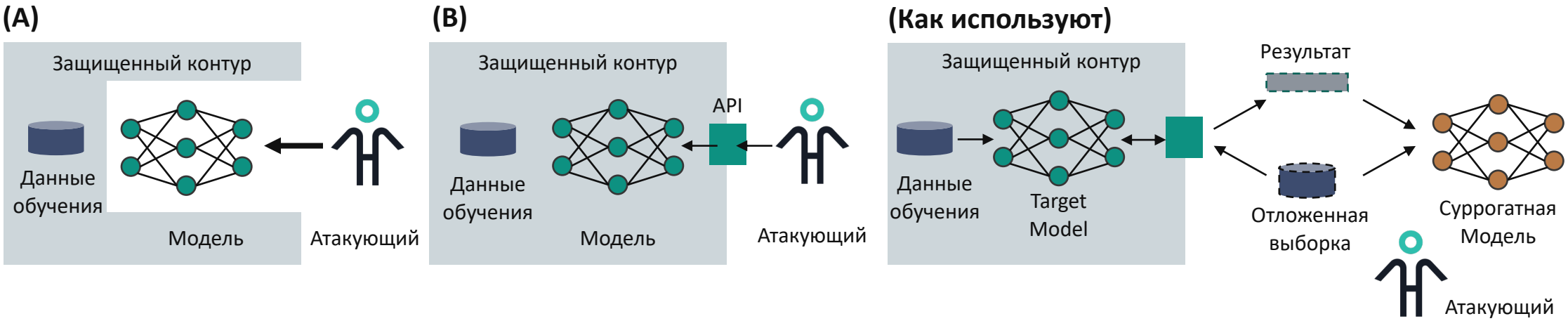
01

Введение

Задачи защиты моделей в облаке

Кража моделей

- **(A) Белый ящик:** у злоумышленника есть доступ к весам и параметрам обучения модели (в некоторых случаях).
- **(B) Черный ящик:** у злоумышленника нет прямого доступа к модели, взаимодействие ведётся через API.



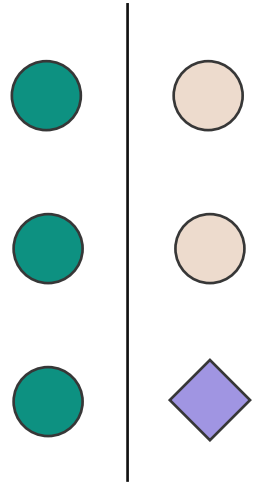
02

Цифровые водяные знаки

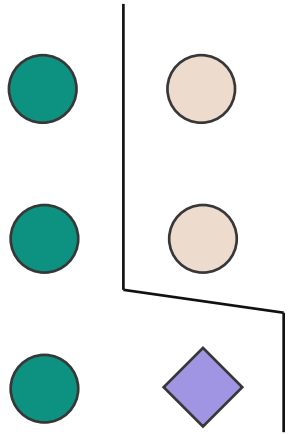
Цифровые водяные знаки для нейросетей

Обычные водяные знаки

Классификация

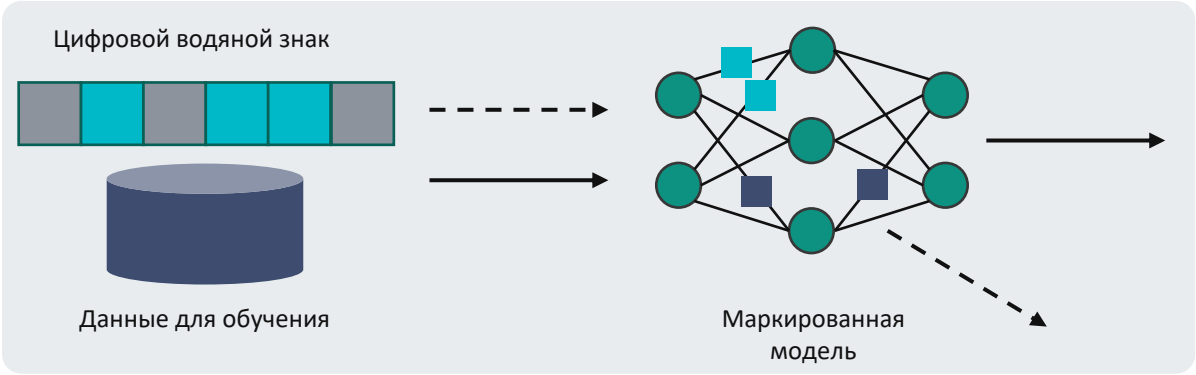


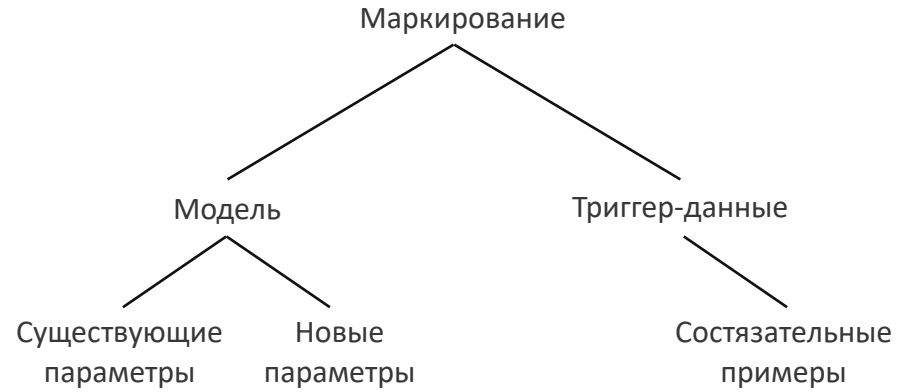
Маркировка



Цифровые водяные знаки для нейросетей

Маркировка сети





Какие есть атаки на удаление маркировки?

- **Distillation** Модель-ученик меньшего размера, обученная повторять поведение тяжелой и более точной модели-учителя, достигает схожих с ней результатов, значительно выигрывая в размере и скорости за счет упрощенной архитектуры [Hinton, 2015; Yang, 2019] .
- **Pruning**. Большое количество параметров позволяет нейросети выявлять сложные зависимости в данных и решать трудные задачи. Однако практика показывает, что для хорошей работы сети не требуется все количество параметров, которые у нее есть [Molchanov, 2019].
- **Model compression**. Оптимизация памяти, например, для мобильных устройств или IoT с ограниченными ресурсами. Сжатие модели осуществляется, например, путем удаления несущественных параметров и сокращения связей между нейронами.
- **Fine-tuning.** Можно улучшить модель для определенных видов данных, но изменение параметров может привести к удалению марки [Shafieinejad, 2021].

Hinton, G., et al(2015). Distilling the Knowledge in a Neural Network. ArXiv preprint
Le Merrer et al, (2020). Adversarial Frontier Stitching for Remote Neural Network Watermarking. Neural Comput. Appl.
Molchanov, et al (2019). Pruning Convolutional Neural Networks for Resource Efficient Inference. ICLR
Shafieinejad, et al (2021). On the Robustness of Backdoor-Based Watermarking in Deep Neural Networks. ACM

Решение – через специальный набор данных

Рассмотрим K классов в задаче классификации:

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$$

$$x_i \in \mathbb{R}^d \qquad y_i \in [1, \ldots, K]$$

Обучаем исходную модель f

$$L(\mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} l(f(x_i), y_i),$$

- Злоумышленник может попытаться украсть функциональность f посредством обучения суррогатной модели f* с использованием суррогатного набора данных для имитации результатов f.
- Чтобы обнаружить кражу, владелец исходной модели может применить водяные знаки на основе набора триггеров. Подмножество исходного набора данных подвергается (перевороту меток)

$$\mathcal{D}_s = \{(x_{i_k}, y_{i_k})\}_{k=1}^{n}$$

- Получим триггер-набор:

$$\mathcal{D}_t = \{(x_{i_k}, y_{i_k}^{'})\}_{k=1}^{n} \qquad y_{i_k}^{'} \neq y_{i_k}$$

- Затем исходная модель обучается, чтобы минимизировать эмпирический риск для измененного набора данных.

$$\mathcal{D} := (\mathcal{D} \setminus \mathcal{D}_s) \cup \mathcal{D}_t$$

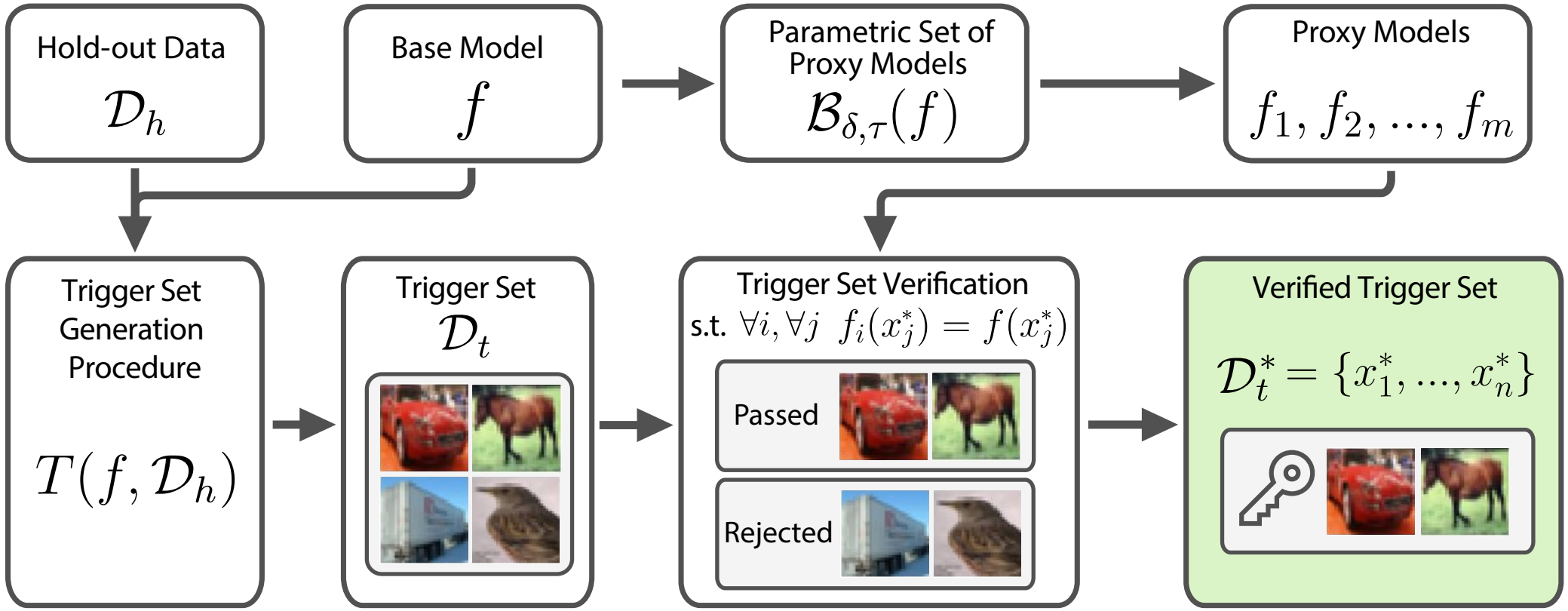
- Если производительность подозрительной модели f* на Dt аналогична производительности f, мы объявляем ее ***украденной***.

AIRI

Сложности подхода

- Размер n набора триггеров должен быть небольшим, чтобы не вызвать заметного снижения производительности. С другой стороны, n должно быть достаточным, чтобы сходство поведения исходной модели и украденной модели было статистически значимым.
- Наборы триггеров, как правило, трудно переносить между исходной моделью и украденной моделью: выборка из набора триггеров оттесняется от выборок того же класса ближе к границе решения суррогатной модели.

Probabilistically Robust Watermarking of Neural Networks



Экспериментальные подходы

- **Soft-label.** Набор обучающих данных D известен, и, учитывая входные данные x, выходные данные f(x) исходной модели представляют собой вектор вероятностей классов. Суррогатная модель f* обучена минимизировать функционал от

$$L_{\text{ext}}(\hat{\mathcal{D}}) = \frac{1}{|\hat{\mathcal{D}}|} \sum_{\hat{x}_i \in \hat{\mathcal{D}}} D_{\text{KL}}(f(\hat{x}_i), f^*(\hat{x}_i)),$$

- **Hard-label.** Набор обучающих данных D известен, и, учитывая входные данные x, выходными данными f(x) исходной модели является метка класса, назначенная f входным данным x. Этот параметр соответствует обучению суррогатной модели на наборе данных.

$$\hat{\mathcal{D}} = \{x_i, f(x_i)\}_{i=1}^{N}.$$

- **RGT (Regularization with Ground Truth Label).** Обучение суррогатной модели, минимизируя потери в наборе обучающих данных D и KL-div между выходными данными исходной модели и суррогатной модели одновременно [Kim et al., 2023].

- Такая настройка соответствует минимизации выпуклой комбинации потерь от:

$$L_{\text{RGT}}(\mathcal{D}, \hat{\mathcal{D}}, \gamma) = \gamma L_{\text{ext}}(\hat{\mathcal{D}}) + (1 - \gamma) L(\mathcal{D}),$$

- $\gamma \in [0, 1]$ – коэффициент регуляризации. В наших экспериментах это самая сильная атака по краже функциональности.

Results

Method	Metric	Source model f	Surrogate models f^*		
			Soft-label	Hard-label	RGT
EWE [Jia *et al.*, 2021]	CIFAR-10 acc. (%)	86.10 ± 0.54	83.97 ± 1.02	82.22 ± 0.50	88.88 ± 0.35
RS [Bansal *et al.*, 2022]		84.17 ± 1.01	88.93 ± 1.18	89.62 ± 0.97	90.14 ± 0.08
MB [Kim *et al.*, 2023]		87.81 ± 0.76	91.17 ± 0.76	91.88 ± 0.40	93.05 ± 0.20
Probabilistic (Ours)		$\mathbf{91.00 \pm 0.00}$	$\mathbf{92.60 \pm 0.91}$	$\mathbf{94.87 \pm 0.59}$	$\mathbf{99.42 \pm 0.02}$
EWE [Jia *et al.*, 2021]	Trigger set acc. (%)	26.88 ± 8.32	51.01 ± 5.58	36.05 ± 6.48	1.64 ± 1.05
RS [Bansal *et al.*, 2022]		95.67 ± 4.93	7.67 ± 4.04	6.33 ± 1.15	3.00 ± 0.00
MB [Kim *et al.*, 2023]		100.00 ± 0.00	82.00 ± 1.00	51.33 ± 4.93	72.67 ± 6.66
Probabilistic (Ours)		100.00 ± 0.00	$\mathbf{85.10 \pm 6.33}$	$\mathbf{73.70 \pm 4.65}$	$\mathbf{78.00 \pm 5.58}$
EWE [Jia *et al.*, 2021]	CIFAR-100 acc. (%)	55.11 ± 1.67	53.00 ± 1.57	46.78 ± 1.00	63.73 ± 0.40
RS [Bansal *et al.*, 2022]		59.87 ± 2.78	65.66 ± 1.53	65.79 ± 0.39	64.99 ± 0.30
MB [Kim *et al.*, 2023]		62.13 ± 4.36	$\mathbf{67.66 \pm 0.36}$	$\mathbf{70.65 \pm 0.49}$	$\mathbf{70.24 \pm 0.46}$
Probabilistic (Ours)		$\mathbf{66.70 \pm 0.00}$	67.49 ± 0.03	68.05 ± 0.73	67.85 ± 0.04
EWE [Jia *et al.*, 2021]	Trigger set acc. (%)	68.14 ± 10.16	30.90 ± 11.34	15.10 ± 5.64	5.73 ± 3.42
RS [Bansal *et al.*, 2022]		99.00 ± 1.00	2.67 ± 1.53	4.33 ± 4.16	2.00 ± 1.00
MB [Kim *et al.*, 2023]		100.00 ± 0.00	70.67 ± 7.57	40.00 ± 8.89	62.66 ± 10.12
Probabilistic (Ours)		100.00 ± 0.00	$\mathbf{78.80 \pm 2.93}$	$\mathbf{74.70 \pm 3.16}$	$\mathbf{79.10 \pm 2.77}$

Table 2: Watermarking performance is reported against functionality stealing methods. The best performance is highlighted in bold. It can be seen that our approach outperforms the other methods of ownership verification by a notable margin.

Method	f^*	$\hat{\mathcal{D}}$	$\text{acc}(\mathcal{D}, f)$	$\text{acc}(\mathcal{D}^*, f)$	$\text{acc}(\mathcal{D}, f^*)$	$\text{acc}(\mathcal{D}^*, f^*)$
MB [Kim *et al.*, 2023]	ResNet34	SVHN	87.81 ± 0.76	100.0 ± 0.00	63.99 ± 3.90	72.00 ± 6.08
	VGG11	CIFAR-10	87.81 ± 0.76	100.0 ± 0.00	86.00 ± 2.17	32.00 ± 7.21
Probabilistic (ours)	ResNet34	SVHN	91.00 ± 0.00	100.0 ± 0.00	73.01 ± 1.18	77.70 ± 2.90
	VGG11	CIFAR-10	91.00 ± 0.00	100.0 ± 0.00	89.24 ± 2.69	80.10 ± 3.86

Table 3: Results of watermarking approaches in the setting when either the training dataset or source model's architecture is unknown to the adversary. Our approach outperforms the baseline in terms of the initial accuracy of the source model and the trigger set accuracy of surrogate models.

AIRI

Заключение

- Новый подход к формированию цифровых водяных знаков на основе набора триггеров для защиты интеллектуальной собственности в контексте атак с кражей модели «черного ящика».
- Метод создает наборы триггеров, которые с высокой вероятностью можно переносить между исходной моделью и суррогатными моделями.
- Подход не зависит от модели. Никакого дополнительного обучения модели не требуется и не накладывается никаких ограничений на размер набора триггеров.
- Таким образом, метод применим к любой модели без ущерба для производительности и минимальных вычислительных затрат для генерации набора триггеров.

Спасибо за внимание

AIRI

@AIRI_Research_Institute

@BASELINEAI

Artificial Intelligence Research Institute

airi.net

Олег Рогов

Руководитель группы «Доверенные и Безопасные Интеллектуальные Системы»
AIRI
rogov@airi.net

Канал в телеграм:
https://t.me/baselineAI

Appendix

Idea of DNN marking

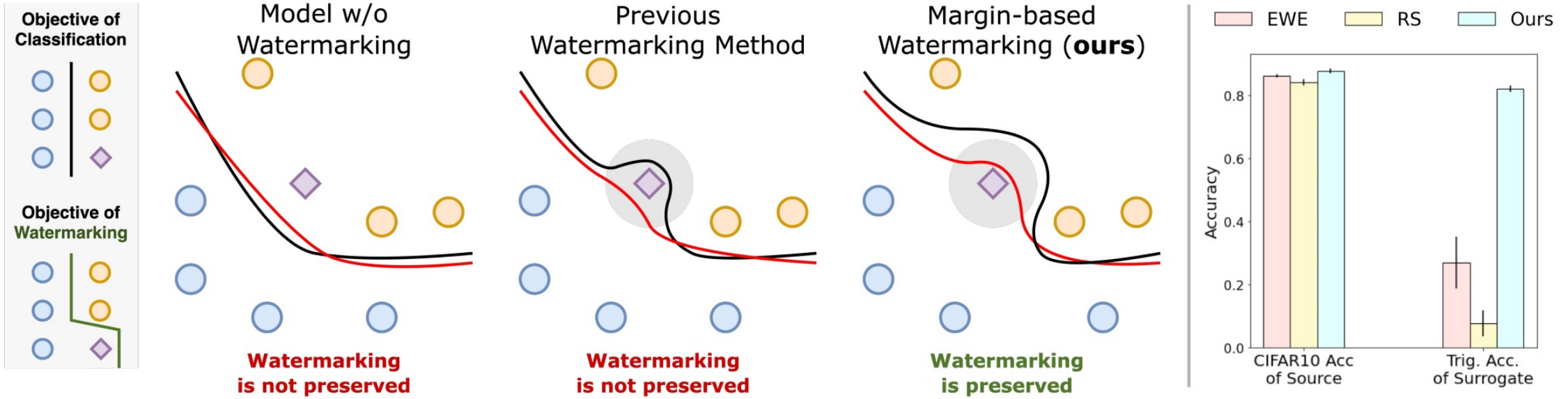


Figure 1: **Concept. (left)** The circle denotes the samples for classification, rhombus is the sample in the trigger set. Black and red line indicates the decision boundary of the source model and its surrogate via stealing the functionality. To claim the ownership, prediction of both source and surrogate model on the trigger set should be the same, which is done by margin-based watermarking since the margin leads the surrogate model to include the rhombus in the same region. **(right)** The performance of the surrogate models with our method and baseline watermarking methods on CIFAR-10 dataset. Ours outperforms the baselines in terms of both source model accuracy and watermarking accuracy.

Kim et al. 2023, Margin-based Neural Network Watermarking

Parameters

Parameters of Experiments

Unless stated otherwise, we use the following values of hyperparameters in our experiments: the size of the verified trigger set n is set to be $n = 100$ for consistency with the concurrent works, confidence level α for Clopper-Pearson test from Eq. (9) is set to be $\alpha = 0.05$. In our experiments, we found that better transferability of the verified trigger set is achieved when no constraint on the performance of the proxy models is applied, so the performance threshold parameter is set to be $\tau = 1.0$.

Integrity

It should be mentioned that a watermarking approach not only should not affect the source model's performance and be robust to stealing attacks, it should also satisfy the property of integrity. In other words, it should not judge non-watermarked networks as watermarked ones. In trigger set-based watermarking settings, checking if a model is stolen may be thought of as a detection problem with certain false positive and false negative rates. The first one corresponds to the probability that a benign model is detected as stolen, and the second one corresponds to the probability that a stolen model is not detected as such.

Assuming that a stolen model belongs to the parametric set of proxy models $\mathcal{B}_{\delta,\tau}(f)$, it is possible to provide probabilistic guarantees that the one would be detected as stolen by our method. In contrast, it is in general nontrivial to guarantee that a benign model would not be detected as stolen. With our method, such guarantees may be provided under certain modifications of the verification procedure. Namely, one may assume that all the models that belong to the compliment $\bar{\mathcal{B}}_{\delta,\tau}(f)$ of the set of proxy models $\mathcal{B}_{\delta,\tau}(f)$ *are not stolen* ones. Then, the verification procedure may be adapted: given models $f_1, \ldots, f_m \in \mathcal{B}_{\delta,\tau}(f)$ and models $\bar{f}_1, \ldots, \bar{f}_m \in \bar{\mathcal{B}}_{\delta,\tau}(f)$, the sample (x^*, y^*) is verified iff:

$$\begin{cases} y^* = f_1(x^*) = \cdots = f_m(x^*), \\ y^* \neq \bar{f}_1(x^*), \\ \ldots \\ y^* \neq \bar{f}_m(x^*). \end{cases} \tag{10}$$

In other words, it is also required that the models from $\bar{\mathcal{B}}_{\delta,\tau}(f)$ are *not* agreed with the source model on the samples from trigger set. It is notable that such a verification procedure requires careful parameterization of the set of proxy models: underestimation of its parameters would lead to some stolen models not being included in it, and overestimation of its parameters may lead to the inclusion of the benign models. In the supplementary material, we include additional experiments on the integrity of our method.

Transferability

In our approach, we assume that all the models from the parametric set $\mathcal{B}_{\delta,\tau}(f)$ are agreed in predictions on data samples from unknown *common set* $\mathcal{S}(f,\delta,\tau)$. In other words, if $f(x)$ is the class assigned by model f to sample x, the set $\mathcal{S}(f,\delta,\tau)$ is defined as follows:

$$\mathcal{S}(f,\delta,\tau) = \{x : f(x) = f^{'}(x)\ \forall f^{'} \in \mathcal{B}_{\delta,\tau}(f)\}. \quad (7)$$

If the stolen model belongs to the set of proxy models $\mathcal{B}_{\delta,\tau}(f)$, a trigger set build-up from points from common set $\mathcal{S}(f,\delta,\tau)$ would be a good candidate for ownership verification: by design, the predictions of the source model and the stolen model would be identical on such a set.

Since it is impossible to guarantee that a certain data point belongs to the common set $\mathcal{S}(f,\delta,\tau)$, we perform the screening of the input space for the candidates to belong to $\mathcal{S}(f,\delta,\tau)$.

Namely, given a candidate x, we check the agreement in predictions of m randomly sampled proxy models $f_1,\dots,f_m$ from $\mathcal{B}_{\delta,\tau}(f)$ and accept x as the potential member of $\mathcal{S}(f,\delta,\tau)$ only if all m models have the same prediction. One can think of the selection process of such points as tossing a coin: checking the predictions of m proxy models represents m coin tosses. The input data points represent unfair coins, i.e., those with different probabilities of landing on heads and tails. If the input point x and the index of proxy model i is fixed, such an experiment $A_i = A_i(x)$ is a Bernoulli trial:

$$A_i(x) = \begin{cases} 1 & \text{with probability } p(x), \\ 0 & \text{with probability } 1-p(x). \end{cases} \quad (8)$$

Let the success of the Bernoulli trial from Eq. (8) correspond to the agreement in predictions of the source model f and $i-$th proxy model f_i. Thus, the screening reduces to the search of input points with the highest probability $p(x)$.

In our experiments, we estimate the parameter $p(x)$ of the corresponding random variable by observing the results of m experiments $A_1(x),\dots,A_m(x)$. We use interval estimation for $p(x)$ in the form of Clopper-Pearson test [Clopper and Pearson, 1934] that returns one-sided $(1-\alpha)$ confidence interval for $p(x)$:

$$\mathbb{P}\left(p(x) \geq B\left(\frac{\alpha}{2}, t, m-t+1\right)\right) \geq 1-\alpha. \quad (9)$$

In Eq. 9, $\hat{p}(x) = B\left(\frac{\alpha}{2}, t, m-t+1\right)$ is the quantile from the Beta distribution and the number of successes $t = m$.

AIRI

Experimental Setup + Evaluation

Datasets and Training

In our experiments, we use CIFAR-10 and CIFAR-100 [Krizhevsky *et al.*, 2009] as training datasets for our source model f. As the source model, we use ResNet34 [He *et al.*, 2016], which is trained for 100 epochs to achieve high classification accuracy (namely, 91.0% for CIFAR-10 and 66.7% for CIFAR-100). We used SGD optimizer with learning rate of 0.1, weight decay of 0.5×10^{-3} and momentum of 0.9.

Parametric Set of Proxy Models

Once the source model is trained, we initialize a parametric set of proxy models $\mathcal{B}_{\delta,\tau}(f)$. In our experiments, we vary the parameters of the proxy models set to achieve better trigger set accuracy of our approach. Namely, parameter δ was varied in the range $[0.5, 40]$ and τ was chosen from the set $\{0.1, 0.2, 1.0\}$. We tested different number of proxy models sampled from $\mathcal{B}_{\delta,\tau}(f)$ for verification. Namely, parameter m was chosen from the set $\{1, 2, 4, 8, 16, 32, 64, 128, 256\}$.

Evaluation Protocol

Once the verified trigger set $\mathcal{D}_t^* = \{(x_i^*, y_i^*)\}_{i=1}^n$ is collected and surrogate model f^* is obtained, we measure the accuracy

$$\text{acc}(\mathcal{D}_t^*, f^*) = \frac{1}{|\mathcal{D}_t^*|} \sum_{(x_i^*, y_i^*) \in \mathcal{D}_t^*} \mathbb{1}\left(f^*(x_i^*) = y_i^*\right) \quad (6)$$

of f^* on $\mathcal{D}_t^*$ to evaluate the effectiveness of our watermarking approach.

AIRI