Appendix: Interesting Near-boundary Data: Model Ownership Inference for DNNs

Anonymous submission

Abstract

Deep neural networks (DNNs) require expensive training, which is why the protection of model intellectual property (IP) is becoming more critical. Recently, model stealing has emerged frequently, and many researchers have been inspired by digital media watermarking to design model watermarking and fingerprinting for verifying model ownership. However, attacks such as ambiguity statements have been used to break the current defense, which poses a challenge to model ownership verification. Therefore, this paper proposes an interesting near-boundary data as evidence for obtaining model ownership and innovatively proposes to infer model ownership instead of verifying model ownership. In this paper, we propose to construct the initial near-boundary data using an algorithm that adds slight noise to generate adversarial examples. We design a DCGAN-based data generator to privatize the near-boundary data. Our main observation is that the near-boundary data exhibit results close to the classification boundary in both the source model and its derived stolen model. At the end of this work, we design many experiments to verify the effectiveness of the proposed method. The experimental results demonstrate that model ownership can be inferred with high confidence. Noting that our method does not require the training data to be private, and it is extremely costly for model stealers to reuse our method.

A Experimental Settings

This paper has leveraged three open source datasets, CIFAR-10, Heritage and Intel_iamge, for ResNet18. Adam optimizer is used during training and the learning rate, epoch and batch size are set to 0.0001, 200 and 64, respectively. Distillation model experiments are chosen to distill from ResNet18 to VGG11, the distillation temperature is set to 20, the teacher model scale $\alpha = 0.7$, and the training epoch is 20. In this paper, the CW-L_2 algorithm is used for the initial near-boundary data generation, and the learning rate, iteration number and dichotomous search number are set to 0.001, 1000 and 6, respectively, while other parameters are set to default values. The private near-boundary data generator uses the DCGAN infrastructure, and the training process uses the Adam optimizer and sets the learning rate, epoch and batch size to 0.0002, 8000 and 64, respectively. Note that the final stage of fine-tuning the source model requires alternating the source model loss function and the loss function of the fine-tuned target classification boundary, which is set to alternate every 10 rounds and the maximum number of alternations is 10.

Dataset	FGSM	IGSM	RFGSM	CW-L_2
CIFAR-10	0.557	0.430	0.418	**0.066**
	0.461	0.419	0.373	0.103
	0.586	**0.369**	**0.365**	0.112
Heritage	0.347	0.356	0.314	0.014
	0.277	0.340	0.281	0.016
	0.348	**0.332**	**0.276**	**0.010**
Intel_images	0.522	0.447	**0.353**	**0.088**
	0.475	0.506	0.387	0.122
	0.468	**0.402**	0.428	0.127

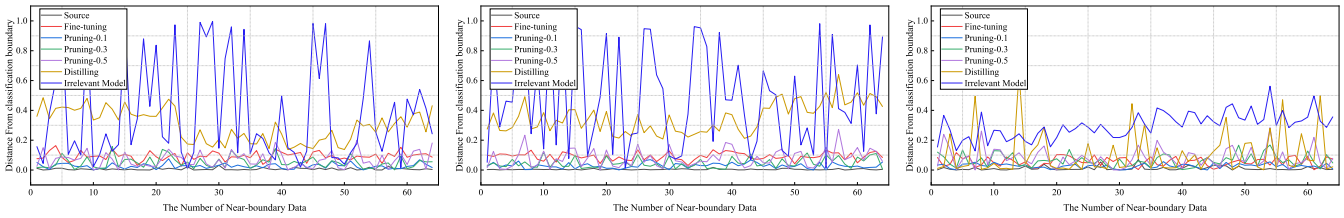
Table 1: Average distance of data generated by different adversarial example generation algorithms from the target classification boundary (Bold is the minimum value of the average distance).

B Evaluation of Near-boundary Properties

This subsection will extend the experiments in Section 5.2. Previously, this paper only showed the near-boundary properties of the near-boundary data on the source, suspect, and irrelevant models when the data size is 64, and only for one of the classification boundaries. Therefore, this subsection will extend the experiment from the perspective of different classification boundaries. As shown in Figures 1,2 and 3, the near-boundary data proposed in this paper exhibit significant near-boundary properties at different target classification boundaries, and the differences between the source model and the suspicious model (stolen model) are very small. In contrast, the differences between the above two and irrelevant models are large. In conclusion, the near-boundary properties of the near-boundary data can be used to infer model ownership.

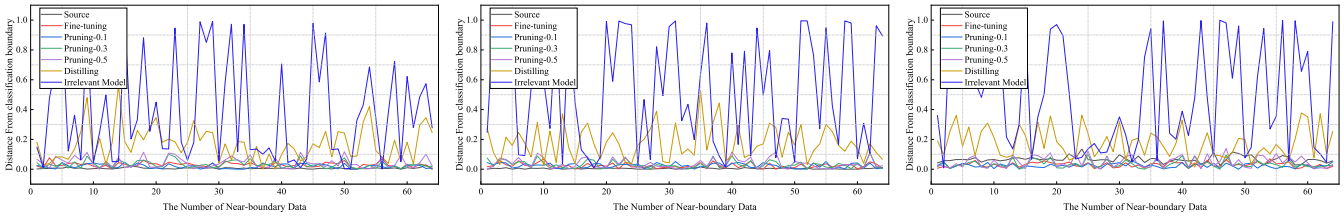
C Algorithm Selection for Initial Near-boundary Data Generation

This subsection tests the FGSM, IGSM, RFGSM and CW-L_2 proposed in Section 4.1, using the implementations published by the original authors. FGSM, IGSM, RFGSM have a parameter to define the noise ϵ, and IGSM and RFGSM



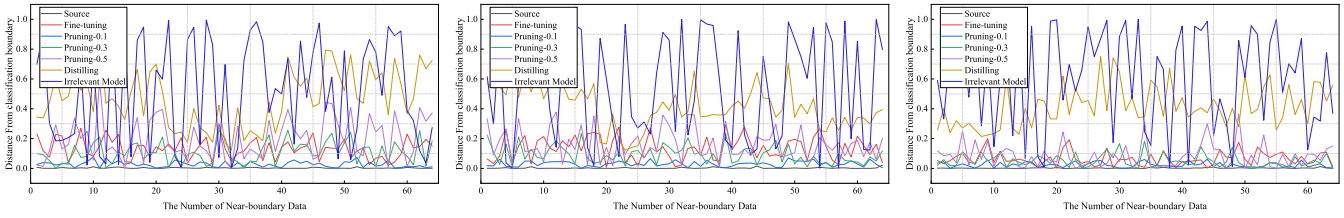
(a) CB_1 (b) CB_2 (c) CB_3

Figure 1: The performance of near-boundary data on CIFAR-10 for the different classification boundaries.



(a) CB_1 (b) CB_2 (c) CB_3

Figure 2: The performance of near-boundary data on Heritage for the different classification boundaries.



(a) CB_1 (b) CB_2 (c) CB_3

Figure 3: The performance of near-boundary data on Intel_image for the different classification boundaries.

also contain an important parameter $alpha$ to indicate the number of iterations. We perform extensive experimental explorations to select the appropriate parameter for comparison with CW-L_2. In addition, the experimental setup of CW-L_2 is shown in Appendix A. As shown in Table 1, the average distance between the adversarial examples generated by CW-L_2 and the target classification boundary is much smaller than that of other algorithms. Therefore, CW-L_2 is used as the initial near-boundary data generation algorithm in this paper.

D Accuracy Impact from Fine-tuning Target Classification Boundaries

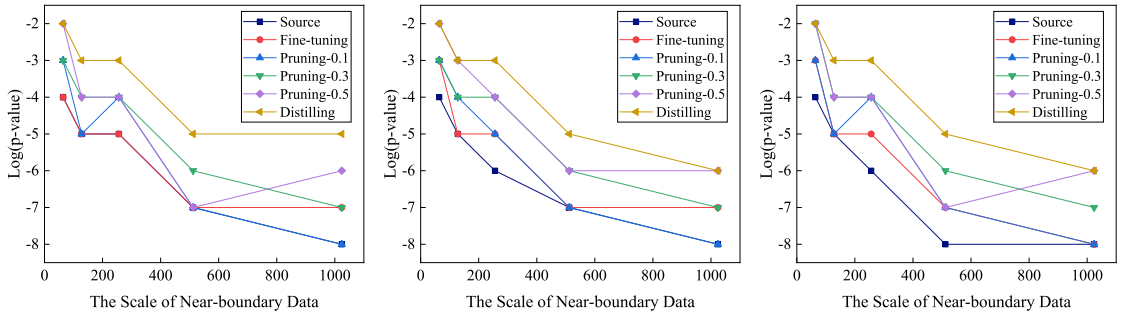
This subsection conducts extensive experiments to test the impact of fine-tuning on the source model in Section 4.2. We fine-tune the source model using different sizes of near-boundary data and different target classification boundaries for the source model, which is obtained by training on different datasets. As shown in Table 2, the accuracy difference between the source model before and after fine-tuning is no more than $5pp$ in almost all cases. The experimental results demonstrate that our near-boundary data have good fidelity when applied to the model ownership protection. Therefore, using near-boundary data to infer model ownership does not need to be concerned about the large impact on the source model.

E Evaluating Scalability Extensions

In Section 5.4, we test the scalability performance of the model ownership inference on near-boundary data. This subsection expands on it to test scalability for different classification boundaries. As shown in Figures 4, 5 and 6 our near-boundary data generated from different target classification boundaries show significant results in inferring model ownership for both source and suspect models trained on different datasets, the same results as those obtained in Section 5, which significantly supports our proposed method.

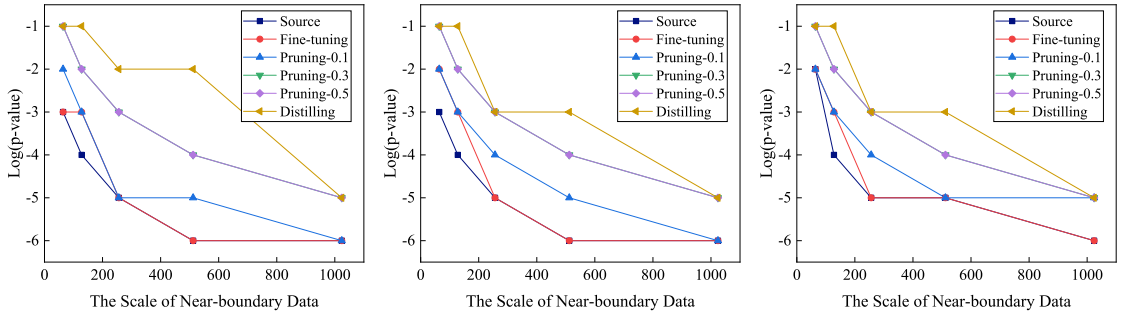
	CIFAR-10 Acc.(0.886)			Heritage Acc.(0.879)			Intel_image Acc.(0.794)	
CB	Data Size	ACC.(Before)	CB	Data Size	ACC.(Before)	CB	Data Size	ACC.(Before)
CB_1	64	0.873	CB_1	64	0.856	CB_1	64	**0.755**
	128	0.862		128	0.825		128	0.769
	256	0.862		256	0.830		256	0.756
	512	**0.854**		512	**0.797**		512	0.779
CB_2	64	0.871	CB_2	64	0.823	CB_2	64	0.770
	128	0.870		128	0.839		128	**0.741**
	256	0.860		256	0.841		256	0.768
	512	**0.844**		512	**0.779**		512	0.777
CB_3	64	0.871	CB_3	64	0.848	CB_3	64	0.781
	128	0.868		128	0.826		128	0.753
	256	0.858		256	**0.779**		256	0.764
	512	**0.856**		512	0.791		512	**0.752**

Table 2: Accuracy of the source model after fine-tuning the three target classification boundaries using near-boundary data with different scales(Bold is the minimum value).



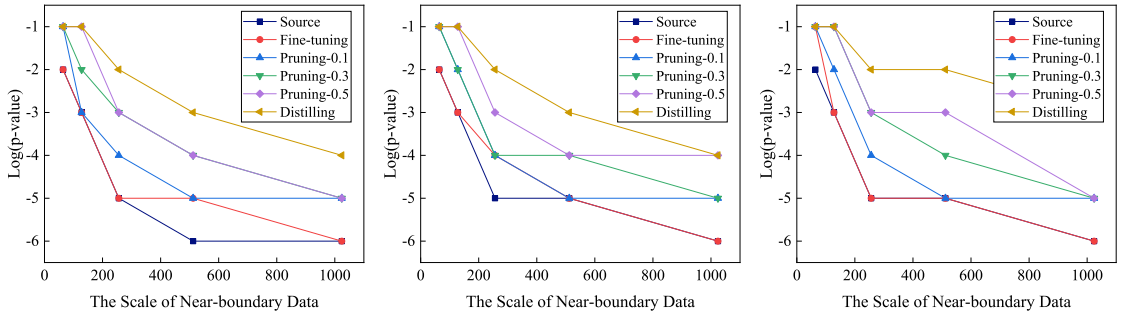
(a) CB_1 (b) CB_2 (c) CB_3

Figure 4: The scalability of model ownership inference on near-Boundary data(CIFAR-10).



(a) CB_1 (b) CB_2 (c) CB_3

Figure 5: The scalability of model ownership inference on near-Boundary data(Heritage).



(a) CB_1 (b) CB_2 (c) CB_3

Figure 6: The scalability of model ownership inference on near-Boundary data(Intel_image).