MIIS 2022
International Workshop On Mathematical Issues In Information Sciences
数据科学国际研讨会
17th-18th, Dec. 2022

Adversarial Machine Learning: Adversarial examples and backdoor learning

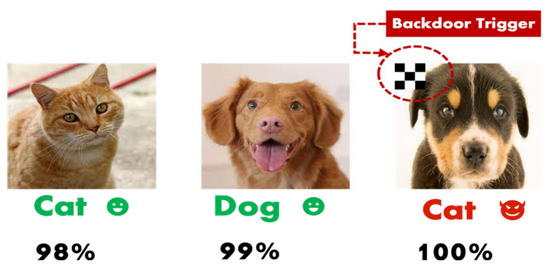
Secure Computing Lab of Big Data

Sample-Specific Backdoor Attack[1]

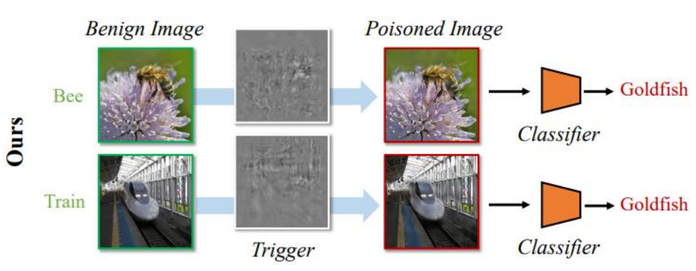
Background

Attackers intend to inject hidden backdoors into DNNs, such that the attacked model performs well on benign samples, whereas its prediction will be maliciously changed if hidden backdoors are activated by the attacker-defined trigger.

- **Sample-agnostic backdoor attack**
  - Different poisoned samples contain the same trigger, resulting in that the attacks could be easily mitigated by current backdoor defenses.



- **Sample-specific backdoor attack**
  - Backdoor triggers are sample-specific invisible additive noises generated by encoding an attacker-specified string into benign images through an encoder-decoder network.



Method

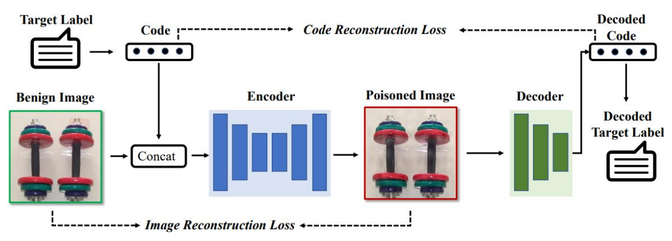
- **The Main Process of Backdoor Attacks**
  - Let $D_{train} = \{x_i, y_i\}_{1=1}^{N}$ indicates the benign training set containing $N_{i.i.d.}$samples, where$x_i \in X = \{0, \dots, 255\}^{C\times W\times H}$ and $y_i \in Y = \{1, \dots, K\}$. The classification learns a function $f_w: X \to [0,1]^K$ with parameters w. Let $y_t$denotes the target label ($y_t \in Y$). The core of backdoor attacks is how to generate the poisoned training set D_p. Specifically, D_p consists of modified version of a subset of D_{train} (i.e.,D_m) and remaining benign samples D_b.

$$D_p = D_b \cup D_m \tag{1}$$

  - Once the poisoned training set $D_p$is generated based on the aforementioned method, backdoor attackers will send it to the user. Users will adopt it to train DNNs with the standard training process, i.e.,

$$\min_{w} \frac{1}{N} \sum_{(x,y)\in \mathcal{D}_p} \mathcal{L}(f_w(x), y) \tag{2}$$

- **How to Generate Sample-specific Triggers**
  - To use a pre-trained encoder-decoder network as an example to generate sample-specific triggers. The encoder takes a benign image and the representative string to generate the poisoned image.



Experimental Evaluation

Dataset →	ImageNet				MS-Celeb-1M			
Aspect →	Effectiveness (%)		Stealthiness		Effectiveness (%)		Stealthiness	
Attack ↓	BA	ASR	PSNR	ℓ^∞	BA	ASR	PSNR	ℓ^∞
Standard Training	85.8	0.0	—	—	97.3	0.1	—	—
BadNets [8]	**85.9**	**99.7**	25.635	235.583	96.0	**100**	25.562	229.675
Blended Attack [3]	85.1	95.8	**45.809**	**23.392**	95.7	99.1	**45.726**	**23.442**
Ours	85.5	99.5	27.195	83.198	**96.5**	**100**	28.659	91.071

Effective Backdoor Defense[2]

Background

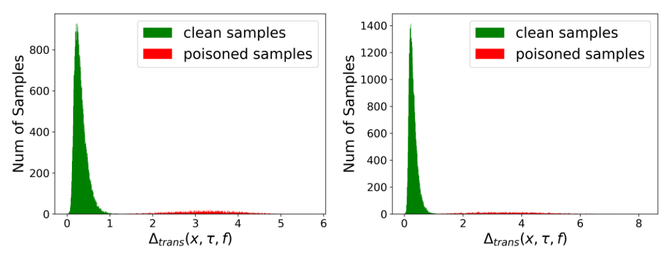
Poisoning-based backdoor attacks are serious threat for training deep models on data from untrustworthy sources. Given a backdoored model, the feature representations of poisoned samples with trigger are more sensitive to transformations than those of clean samples. Hence, we design a simple sensitivity metric, called feature consistency towards transformations (FCT), to distinguish poisoned samples from clean samples in the untrustworthy training set.

Method

- **Sensitivity of poisoned samples**
  - Given a backdoored model g_θ trained on $\overline{D}_{train}$ with $f_{\theta_e}(\cdot)$ indicating its feature extractor, and a set of transformations τ (e.g., rotation, scaling, will be specified in experiments), for any sample x (poisoned or clean), the FCT metric is formulated as follows:

$$\Delta_{trans}\left(\boldsymbol{x}; \tau, f_{\boldsymbol{\theta}_e}\right) = \left\| f_{\boldsymbol{\theta}_e}(\boldsymbol{x}) - f_{\boldsymbol{\theta}_e}(\tau(\boldsymbol{x})) \right\|_2^2 \tag{1}$$

  - Utilizing FCT, we develop a sample-distinguishment (SD) module. we firstly train a backdoored model g_θ based on $\overline{D}_{train}$ using the standard supervised learning algorithm with a few epochs. Then, we calculate $\Delta_{trans}(x_i)$, all $x_i \in \overline{D}_{train}$ and plot the histogram.



(a) BadNets (b) Blend

- **Secure training from scratch**
  - Learning feature extractor via semi-supervised contrastive learning (SS-CTL).

$$\mathcal{L}_{SS-CTL}(\boldsymbol{\theta}_e; \overline{D}_{train}) = \sum_{(x_i,y_i)\in \widehat{D}_p \cup \widehat{D}_u} \ell_{CTL}\left(f_{\boldsymbol{\theta}_e}\left(\tilde{\boldsymbol{x}}_i^{(1)}\right), f_{\boldsymbol{\theta}_e}\left(\tilde{\boldsymbol{x}}_i^{(2)}\right)\right) + \sum_{\{(x_i,y_i),(x_j,y_j)\}\subset \widehat{D}_c} \ell_{S-CTL}\left(f_{\boldsymbol{\theta}_e}\left(\tilde{\boldsymbol{x}}_i^{(1)}\right), f_{\boldsymbol{\theta}_e}\left(\tilde{\boldsymbol{x}}_i^{(2)}\right), f_{\boldsymbol{\theta}_e}\left(\tilde{\boldsymbol{x}}_j^{(1)}\right), f_{\boldsymbol{\theta}_e}\left(\tilde{\boldsymbol{x}}_j^{(2)}\right); y_i, y_j\right) \tag{2}$$

  - Learning classifier via minimizing the mixed cross-entropy loss

$$\mathcal{L}_{MCE}\left(\boldsymbol{\theta}_c; \widehat{D}_c, \widehat{D}_p\right) = \frac{-1}{|\widehat{D}_c|} \sum_{(\boldsymbol{x},y)\in \widehat{D}_c} \log\left[h_{\boldsymbol{\theta}_c}\left(f_{\boldsymbol{\theta}_e}(\boldsymbol{x})\right)\right]_y + \frac{\lambda_p}{|\widehat{D}_p|} \cdot \sum_{(\boldsymbol{x},y)\in \widehat{D}_p} \log\left[h_{\boldsymbol{\theta}_c}\left(f_{\boldsymbol{\theta}_e}(\boldsymbol{x})\right)\right]_y \tag{3}$$

- **Backdoor removal**
  - Unlearning

$$\mathcal{L}_{unlearn}\left(\boldsymbol{\theta}; \widehat{D}_p\right) = \frac{1}{|\widehat{D}_p|} \sum_{(\boldsymbol{x},y)\in \widehat{D}_p} \log[g_{\boldsymbol{\theta}}(\boldsymbol{x})]_y \tag{4}$$

  - Relearning

$$\mathcal{L}_{relearn}\left(\boldsymbol{\theta}; \widehat{D}_c\right) = \frac{1}{|\widehat{D}_c|} \sum_{(\boldsymbol{x},y)\in \widehat{D}_c} -\log[g_{\boldsymbol{\theta}}(\boldsymbol{x})]_y \tag{5}$$

Experimental Evaluation

Dataset ↓	Defense → / Attack ↓	Baseline1 ACC	Baseline1 ASR	Baseline2 ACC	Baseline2 ASR	DBD ACC	DBD ASR	D-ST ACC	D-ST ASR
CIFAR-10	BN-all2one	83.54	2.60	91.32	99.91	92.75	100.00	**92.77**	**0.03**
	BN-all2all	83.95	2.72	91.59	57.39	92.95	75.21	**89.22**	**2.05**
	Trojan	83.77	5.24	93.63	99.98	92.81	100.00	**93.72**	**0.00**
	Blend-Strip	85.36	99.93	94.19	100.00	94.21	99.98	**93.59**	**0.00**
	Blend-Kitty	85.03	99.99	94.31	100.00	93.32	100.00	**91.82**	**0.00**
	SIG	85.14	99.02	94.37	99.93	94.37	99.71	**90.07**	**0.00**
	CL	85.79	10.76	94.58	98.87	94.32	99.87	**90.46**	**6.40**
	Avg	84.65	45.75	93.43	93.73	93.53	96.40	**91.66**	**1.21**
CIFAR-100	BN-all2one	54.48	10.41	67.62	100.00	69.08	100.00	**68.43**	**0.12**
	Trojan	56.17	12.76	71.01	100.00	72.18	99.99	**68.04**	**0.08**
	Blend-Strip	58.01	99.91	72.47	99.99	71.29	99.99	**67.63**	**0.00**
	Blend-Kitty	57.21	99.99	73.36	99.99	72.43	100.00	**67.06**	**0.00**
	Avg	56.47	55.77	71.12	100.00	71.24	99.99	**67.79**	**0.05**

Adversarial Attacks with Reverse Adversarial Perturbation[3]

- We encourage that not only $\boldsymbol{x}^{adv}$ itself has low loss value, but also the points in the vicinity of $\boldsymbol{x}^{adv}$ have similarly low loss values.
- We propose to minimize the maximal loss value within a local neighborhood region around the adversarial example $\boldsymbol{x}^{adv}$.
- The maximal loss is implemented by perturbing $\boldsymbol{x}^{adv}$ to maximize the attack loss, named **Reverse Adversarial Perturbation (RAP)**. Our final optimization problem:

$$\min_{\boldsymbol{x}^{adv}\in \mathcal{B}_\epsilon(\boldsymbol{x})} \mathcal{L}\left(\mathcal{M}^s\left(\mathcal{G}\left(\boldsymbol{x}^{adv} + \boldsymbol{n}^{rap}\right); \boldsymbol{\theta}\right), y_t\right) \tag{1}$$

where

$$\boldsymbol{n}^{rap} = \arg\max_{\|\boldsymbol{n}^{rap}\|_\infty \le \epsilon_n} \mathcal{L}\left(\mathcal{M}^s\left(\boldsymbol{x}^{adv} + \boldsymbol{n}^{rap}; \boldsymbol{\theta}\right), y_t\right) \tag{2}$$

To solve the inner loop:

$$\boldsymbol{n}^{rap} \leftarrow \boldsymbol{n}^{rap} + \alpha_n \cdot \text{sign}\left(\nabla_{\boldsymbol{n}^{rap}} \mathcal{L}\left(\mathcal{M}^s\left(\boldsymbol{x}^{adv} + \boldsymbol{n}^{rap}; \boldsymbol{\theta}\right), y_t\right)\right) \tag{3}$$

To solve the outer loop:

$$\boldsymbol{x}^{adv} \leftarrow \text{Clip}_{\mathcal{B}_\epsilon(\boldsymbol{x})}\left[\boldsymbol{x}^{adv} - \alpha \cdot \text{sign}\left(\nabla_{\boldsymbol{x}^{adv}} \mathcal{L}\left(\mathcal{M}^s\left(\mathcal{G}\left(\boldsymbol{x}^{adv} + \boldsymbol{n}^{rap}\right); \boldsymbol{\theta}\right), y_t\right)\right)\right] \tag{4}$$

Experimental Evaluation

Attack	ResNet-50 ⟹ Dense-121	ResNet-50 ⟹ VGG-16	ResNet-50 ⟹ Inc-v3	DenseNet-121⟹ Res-50	DenseNet-121⟹ VGG-16	DenseNet-121⟹ Inc-v3
MTDI / +RAP / +RAP-LS	74.9 / 78.2 / **88.5**	62.8 / 72.9 / **81.5**	10.9 / 28.3 / **33.2**	44.9 / 64.3 / **74.5**	38.5 / 55.0 / **65.5**	7.7 / 23.0 / **26.5**
MTDSI / +RAP / +RAP-LS	86.3 / 88.4 / **93.3**	70.1 / 77.7 / **84.7**	38.1 / 51.8 / **58.0**	55.0 / 71.2 / **75.8**	42.0 / 58.4 / **62.3**	19.8 / 39.0 / **39.2**
MTDAI / +RAP / +RAP-LS	91.4 / 89.4 / **93.6**	79.9 / 79.0 / **86.3**	50.8 / 57.1 / **64.1**	69.1 / 74.2 / **82.1**	54.7 / 63.1 / **69.3**	32.0 / 43.5 / **49.3**
Attack	VGG-16 ⟹ Res-50	VGG-16 ⟹ Dense-121	VGG-16 ⟹ Inc-v3	Inc-v3⟹ Res-50	Inc-v3⟹ Dense-121	Inc-v3⟹ VGG-16
MTDI / +RAP / +RAP-LS	11.8 / 16.7 / **22.9**	13.7 / 19.4 / **27.4**	0.7 / 3.4 / **4.6**	1.8 / **8.3** / 7.5	4.1 / **14.8** / 13.4	2.9 / 8.0 / **9.8**
MTDSI / +RAP / +RAP-LS	31.0 / 35.3 / **38.7**	41.7 / 44.4 / **49.6**	9.6 / **15.2** / 13.7	5.6 / **11.9** / 10.7	10.4 / **21.2** / 20.9	4.2 / **8.9** / 8.6
MTDAI / +RAP / +RAP-LS	36.2 / 39.0 / **43.1**	48.0 / 45.1 / **55.2**	11.6 / 17.1 / **17.6**	9.6 / 13.6 / **16.7**	17.9 / 27.5 / **31.6**	8.4 / 12.0 / **12.1**

Reference:

[1] Yuezun Li, Yiming Li, Baoyuan Wu (corresponding author), Longkang Li, Ran He, and Siwei Lyu (corresponding author). Invisible Backdoor Attack with Sample-Specific Triggers. In **ICCV 2021.**

[2] Weixin Chen , Baoyuan Wu (corresponding author), and Haoqian Wang. Effective Backdoor Defense by Exploiting Sensitivity of Poisoned Samples. In **NeurIPS 2022 (Spotlight).**

[3] Zeyu Qin (equal contribution), Yanbo Fan (equal contribution), Yi Liu, Li Shen, Yong Zhang, Jue Wang, and Baoyuan Wu (corresponding author). Boosting the Transferability of Adversarial Attacks with Reverse Adversarial Perturbation. In **NeurIPS 2022.**

深圳市大数据研究院 Shenzhen Research Institute of Big Data
香港中文大學（深圳） The Chinese University of Hong Kong, Shenzhen
TBSI 清华-伯克利深圳学院 Tsinghua-Berkeley Shenzhen Institute
雷达信号处理国家级重点实验室 National Key Laboratory of Radar Signal Processing
综合业务网理论及关键技术国家重点实验室 State Key Laboratory of Integrated Services Networks
鹏城实验室 PENGCHENG LABORATORY
广东省大数据计算基础理论与方法重点实验室 GuangDong Key Laboratory of Big Data Computing
国家健康医疗大数据研究院（深圳） NATIONAL HEALTH DATA INSTITUTE, SHENZHEN