CHAPTER 5

Training-time adversarial attacks

Aforementioned attacks in this part focus on evasion attacks targeting on an already trained and fixed model at the testing/deployment phase. Here we shift our focus to training-phase attacks. Training-phase attacks assume the ability to modify the training data to achieve malicious attempts on the resulting model when trained on the manipulated datasets, which can be realized through noisy data collection such as crowdsourcing. Specifically, the memorization effect of deep learning models (Zhang et al., 2017; Carlini et al., 2019b) can be leveraged as vulnerabilities. We note that sometimes the term "data poisoning" entails both poisoning and backdoor attacks though their attack objectives are different. In this chapter, (data) poisoning attack has a specific goal of degrading the generalization of a target model, such that the model trained on the poisoned training dataset will fail to generalize well on the clean testing dataset (e.g., having high training accuracy but low testing accuracy). On the other hand, a backdoored model tends to behave like a normal (nonbackdoored) model when the embedded trigger patter is absent; when the trigger is present, the backdoored model will give a designed output by the adversary regardless of the actual contents of any data inputs.

5.1 Poisoning attack

Poisoning attack aims to design a poisoned dataset D_{poison} such that models trained on D_{poison} will fail to generalize on a standard test set D_{test}. The poisoned dataset D_{poison} can be created by modifying the original training dataset D_{train}, such as label flipping, data addition/deletion, and feature modification. The rationale is training on D_{poison} will land on a "bad" local minimum of model parameters in the loss landscape. Take classification task as an example, letting f_θ be the classifier trained on a poisoned dataset, which gives the most-likely class prediction. Data poisoning aims to lead the classifier to give a wrong prediction on most of test data samples (but the model still has relatively good accuracy on the training dataset) such that $f_\theta(x_{\text{test}}) \neq y_{\text{test}}$, where $(x_{\text{test}}, y_{\text{test}}) \sim D_{\text{test}}$.

In general, a poisoning attack can be realized by adding a poisoned dataset D_{poison} to the existing training dataset D_{train} and by solving the fol-

Adversarial Robustness for Machine Learning
https://doi.org/10.1016/B978-0-12-824020-5.00014-4

lowing bilevel optimization problem:

$$\underset{D_{\text{poison}}}{\text{Maximize}}\ \mathcal{L}_{\text{attack}}(D'; \theta^*) \text{ such that } \theta^* \in \underset{\theta}{\text{argmin}}\ \mathcal{L}_{\text{train}}(D_{\text{train}} \cup D_{\text{poison}}; \theta), \tag{5.1}$$

where $\mathcal{L}_{\text{attack}}$ is the attacker's designed objective function (higher is better for the attacker), D' is an untrained (hold-out) dataset for attack performance evaluation, and θ^* is the model parameters obtained by minimizing a standard training loss function $\mathcal{L}_{\text{train}}$ on the augmented poisoned dataset denoted by $D_{\text{train}} \cup D_{\text{poison}}$. The attacker's objective function can be as simple as the training loss; for example, $\mathcal{L}_{\text{attack}}$ can be the cross entropy loss.

To control the amount of data modification and reduce the overall accuracy on D_{test} (i.e., test accuracy), poisoning attack often assumes the knowledge of target model and its training method (Jagielski et al., 2018). Liu et al. (2020b) propose black-box poisoning with additional conditions on the training loss function. Targeted poisoning attack aims at manipulating the prediction of a subset of data samples in D_{test}, which can be accomplished by clean-label poisoning (small perturbations to a subset of D_{train} while keeping their labels intact) (Shafahi et al., 2018; Zhu et al., 2019b) or gradient-matching poisoning (Geiping et al., 2021).

5.2 Backdoor attack

Backdoor attack is also known as Trojan attack. The central idea is to embed a universal trigger Δ to a subset of data samples in D_{train} with a modified target label t (Gu et al., 2019). Examples of trigger patterns are a small patch in images and a specific text string in sentences. Typically, backdoor attack only assumes access to the training data and does not assume the knowledge of the model and its training. The model f_θ trained on the tampered data is called a backdoored (Trojan) model. Its attack objective has two folds: (i) High standard accuracy in the absence of trigger – the backdoored model should behave like a normal model (same model trained on untampered data), i.e., $f_\theta(x_{\text{test}}) = y_{\text{test}}$; (ii) High attack success rate in the presence of trigger – the backdoored model will predict any data input with the trigger as the target label t, i.e., $f_\theta(x_{\text{test}} + \Delta) = t$. Therefore backdoor attack is stealthy and insidious. The trigger pattern can also be made input-aware and dynamic (Nguyen and Tran, 2020).

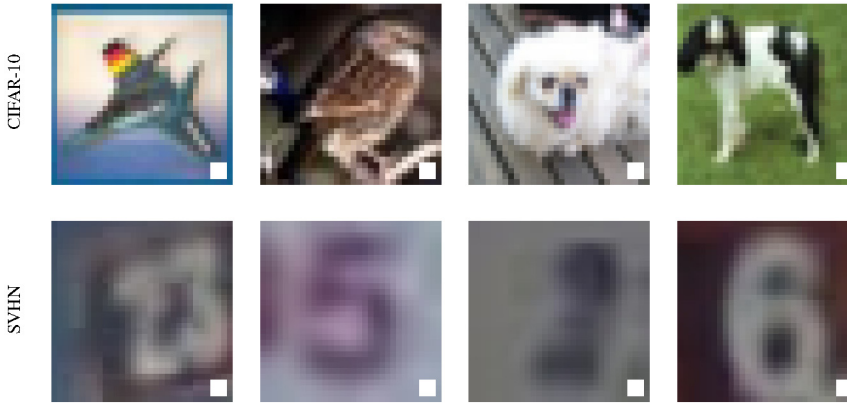


Figure 5.1 Examples of backdoored images on CIFAR-10 and SVHN. The triggers are white blocks located at the right-bottom area of each image, and their associated data labels are changed to a target class label.

5.3 Empirical comparison

We follow the procedures in (Gu et al., 2019) to implement backdoor attacks and obtain two independently backdoored models trained on the same manipulated training data (models I and II). The trigger pattern is placed at the right-bottom of the poisoned images as shown in Fig. 5.1. Specifically, 10% of the training data are backdoored by inserting the trigger and changing the original correct labels to the target label(s). Here we investigate two kinds of backdoor attacks: (a) single-target attack that sets the target label t to a specific label (we choose $t =$ class 1); and (b) all-targets attack where the target label t is set to the original label i plus 1 and then modulo 9, i.e., $T = i + 1 (\text{mod } 9)$.

Their performance on clean (untriggered) and triggered data samples are given in Table 5.1. The backdoored models have similar performance on clean data as untampered models but will indeed misclassify a majority of triggered samples. Comparing to single-target attack, all-targets attack is more difficult and has a higher attack failure rate, since the target labels vary with the original labels.

5.4 Case study: distributed backdoor attacks on federated learning

Federated learning (FL) has been recently proposed to address the problems for training machine learning models without direct access to diverse train-

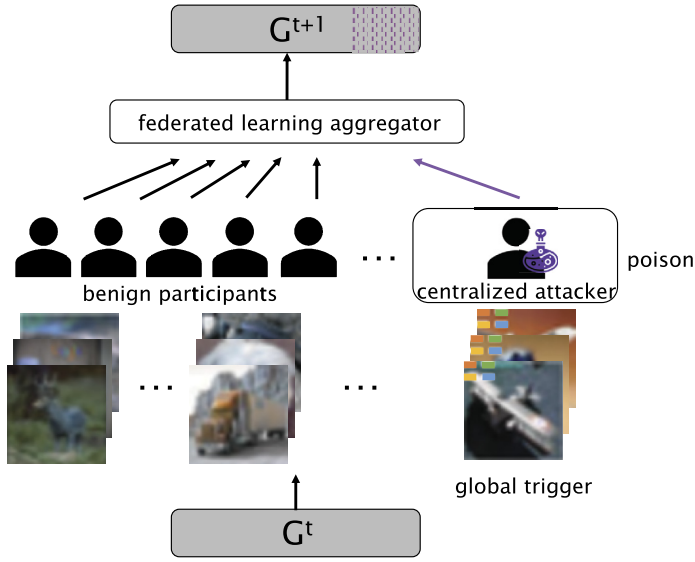
Table 5.1 Error rate of backdoored models. The error rate of clean/backdoored samples means standard-test-error/attack-failure-rate, respectively. The results are evaluated on 5000 nonoverlapping clean/triggered images selected from the test set. For reference, the test errors of clean images on untampered models are 12% for CIFAR-10 (VGG) and 4% for SVHN (ResNet).

	Backdoor attacks	**Single-target attack**		**All-targets attack**	
	Dataset	**CIFAR-10 (VGG)**	**SVHN (ResNet)**	**CIFAR-10 (VGG)**	**SVHN (ResNet)**
Model I	Clean images	15%	5.4%	14.2%	6.1%
	Triggered images	0.07%	0.22%	12.9%	8.3%
Model II	Clean images	13%	7.7%	19%	7.5%
	Triggered images	2%	0.17%	13.6%	9.2%

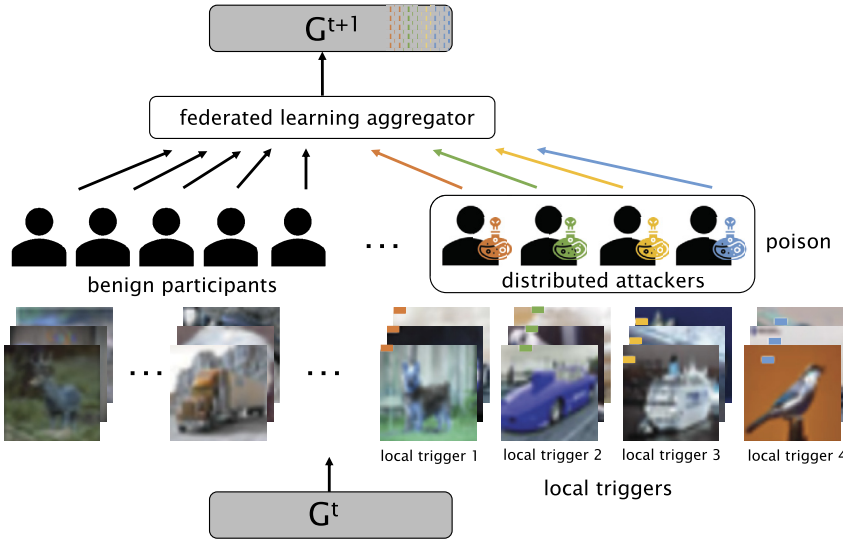
ing data, especially for privacy-sensitive tasks (Smith et al., 2017; McMahan et al., 2017; Zhao et al., 2018). Utilizing local training data of participants (i.e., parties), FL helps train a shared global model with improved performance. There have been prominent applications and ever-growing trends in deploying FL in practice, such as loan status prediction, health situation assessment (e.g., potential cancer risk assessment), and next-word prediction while typing (Hard et al., 2018; Yang et al., 2018, 2019a).

Although FL is capable of aggregating dispersed (and often restricted) information provided by different parties to train a better model, its distributed learning methodology and inherently heterogeneous (i.e., non-i.i.d.) data distribution across different parties may unintentionally provide a venue to new attacks. In particular, the fact of limiting access to individual party data due to privacy concerns or regulation constraints may facilitate backdoor attacks on the shared model trained with FL.

Backdoor attacks on FL have been studied in (Bagdasaryan et al., 2018; Bhagoji et al., 2019). However, these attacks do not fully exploit the distributed learning methodology of FL, as they embed the *same* global trigger pattern to all adversarial parties. This attacking scheme is referred to as the *centralized* backdoor attack. Leveraging the power of FL in aggregating dispersed information from local parties to train a shared model, Xie et al. (2020) propose *distributed* backdoor attack (DBA) against FL. Given the same global trigger pattern as the centralized attack, DBA decomposes it into local patterns and embed them to different adversarial parties respectively. A schematic comparison between the centralized and distributed backdoor attacks is illustrated in Fig. 5.2.



(a) centralized backdoor attack



(b) DBA: distributed backdoor attack (Xie et al., 2020)

Figure 5.2 Overview of centralized and distributed backdoor attacks on FL. The aggregator in FL at round $t+1$ combines information from local parties (benign and adversarial) in the previous round t and updates the shared model. When implementing backdoor attacks, centralized attacker uses a global trigger, whereas distributed attacker uses a local trigger, which is part of the global one.

Centralized backdoor attack embeds the same global trigger for all local attackers[1] (Bagdasaryan et al., 2018). For example, the attacker in Fig. 5.2(a) embeds the training data with the selected patterns highlighted by four colors, which altogether constitute a complete global pattern as the backdoor trigger. In DBA, as illustrated in Fig. 5.2(b), all attackers only use parts of the global trigger to poison their local models, whereas the ultimate adversarial goal is still the same as centralized attack: using the global trigger to attack the shared model. For example, the attacker with the orange (gray in print version) sign poisons a subset of his training data *only* using the trigger pattern located at the orange (gray in print version) area. Similar attacking methodology applies to green (mid gray in print version), yellow (light gray in print version), and blue (dark gray in print version) signs. We define each DBA attacker's trigger as the *local trigger* and the combined whole trigger as the *global trigger*. DBA can also be realized on irregular shape triggers, such as decomposing the logo "ICLR" into "I", "C", "L", "R" as local triggers on three image datasets and decomposing the physical pattern glasses into four parts as shown in Fig. 5.3.

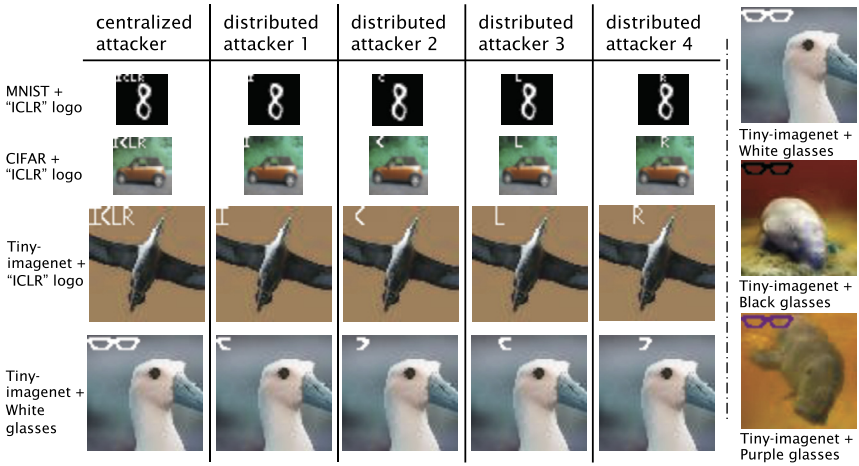


Figure 5.3 Examples of irregular shape triggers in image datasets in (Xie et al., 2020).

Keeping similar amount of total injected triggers (e.g., modified pixels) for both centralized attack and DBA, it was shown by Xie et al. (2020)

[1] Although only one centralized attacker and one adversarial party are shown in Fig. 5.2, in practice, centralized backdoor attack can poison multiple parties with the same global trigger, as discussed by Bagdasaryan et al. (2018).

that although none of the adversarial party has ever been poisoned by the global trigger under DBA, DBA indeed outperforms centralized attack significantly when evaluated with the global trigger. Moreover, DBA achieves higher attack success rate, faster convergence, and better resiliency in single-shot and multiple-shot attack scenarios. It is also demonstrated that DBA is more stealthy and can successfully evade two robust FL approaches.

5.5 Extended reading

- Mehra et al. (2021a) study the robustness of randomized smoothing defenses (see Chapter 14) at test time in the presence of data poisoning attacks.
- Mehra et al. (2021b) use poisoning attack to study the performance of unsupervised domain adaptation algorithms.
- Zawad et al. (2021) study the effect of data heterogeneity on affecting the robustness of federated learning against backdoor attacks.
- Detailed survey on backdoor and poisoning attacks can be found in (Goldblum et al., 2022).
- Zhao et al. (2020a) propose to use mode connectivity in the loss landscape of neural networks to sanitize backdoored models with limited clean data by mitigating their adversarial effects while maintaining clean accuracy.