DSBP: Data-free and Swift Backdoor Purification for Trustworthy Federated Learning via Multi-teacher Adversarial Distillation

Gaolei Li[1] , **Jun Wu** [1] and **Jianhua Li** [1] and **Yuanyuan Zhao** [2] and **Longfei Zheng** [3]

[1] Shanghai Jiao Tong University

[2] Hangzhou Normal University

[3] Ant Group

{gaolei_li, junwuhn, lijh888}@sjtu.edu.cn, yyzhao04@163.com, zlf206411@antgroup.com

Abstract

Federated learning (FL) faces with severe backdoor threats. Due to the inaccessibility of clean samples, the parameter server cannot clean them up in real time even if poisoning features are discovered. Meanwhile, existing backdoor defence methods always require sacrificing model accuracy or increasing communication delay in exchange for better FL trustworthiness, which is unpractical in real scenarios. To address these challenges, we propose a novel data-free and swift backdoor purification (DSBP) scheme based on multi-teacher adversarial distillation, which can effectively erase various backdoor variants in FL. The DSBP treats the purification task as an adversarial game process between knowledge inheritance and backdoor inhibition, with the goal of enforcing the student model to learn the ensemble results of multiple teacher models on reconstructed clean samples, while being insensitive to synthetic poisoned samples. In DSBP, we propose to utilize the self-similarity of poisoned features to optimize the trigger generator, which is essential to accelerate the convergence of DSBP during the adversarial distillation process. We validate that the effectiveness of proposed DBSP by comparing with 4 state of-the-art defense approaches against 3 backdoor variants on 3 datasets. The aversage attack success rate can be reduced from 96.6% to 2.3% with only 200 epochs.

1 Introduction

Federated learning (FL) coordinates a large number of distributed clients to complete a global model training task over massive local data samples [Lim *et al.*, 2020]. A typical FL system mainly includes two kinds of entities: 1) clients, which can receive learning tasks and submit model updates; 2) servers, which can aggregate distributed model updates to obtain a global model based on specific rules. Recently, backdoor attacks on FL obtains increasing attention due to the high attack success rate they have achieved [Li *et al.*, 2022]. New backdoor variants in FL render previous backdoor defense methods that aim to do everything possible (.i.e., trigger removal, ensemble prediction) at the client side to disrupt the necessary backdoor implantation conditions uselessly [Hayase *et al.*, 2021]. Firstly, FL infrastructures are often delivered by open-source platforms (such as WeBank FATE, TensorFlow-federated, PaddleFL, etc). Such third-party FL infrastructure offers a venue for new backdoor variants, such as poisoning the pre-trained models [Jia *et al.*, 2022], neuron hijacking [Liu *et al.*, 2018b], and even code poisoning [Bagdasaryan and Shmatikov, 2020]. Secondly, in a real FL scenario, the cost of identifying poisoned samples one by one is very huge. Moreover, since FL does not usually require every node to participate in the training process, it is difficult to determine the deployment location, timing, and scale of existing defense methods [Goldblum *et al.*, 2022]. In terms of backdoor purification methods that target to remove backdoors from the final delivered models [Qiao *et al.*, 2019; Wang *et al.*, 2019; Li *et al.*, 2021; Liu *et al.*, 2021], it's workflow is often divided into two stages: 1) model diagnosing, 2) model sanitizing. The former stage aims to determine if the suspect model really contains a hidden backdoor [Chen *et al.*, 2019b; Kolouri *et al.*, 2020; Xu *et al.*, 2021]. The model sanitizing stage aims to "forget" hidden backdoors using fine-tuned [Wang *et al.*, 2019], pruned [Liu *et al.*, 2018a], or distilled [Li *et al.*, 2021].

Although many defence methods have been validated to perform reasonably well in experimental settings, three troubles it still should deal with in real-world FL systems: 1) *Lack of adaptability to multiple backdoor variants.* During the whole backdoor purification process, the criteria for model diagnosis is extremely rigid so that it will not work when the adversary changes attack modes [Wang *et al.*, 2019]. In other words, it will suffer from a high misdiagnosis rate. 2) *Hindering the model accuracy.* As the intensity of model purification increases, the backdoor gets weaker. But evaluations in [Yan *et al.*, 2023] show that existing data-driven methods have unacceptable model accuracy degradation (10%) on the CIFAR10 dataset when all employed backdoors are wiped out. In the FL scenario, this degradation will be more sharply.

Our work: We propose a novel data-free swift backdoor purification (DSBP) scheme for trustworthy FL, in which a multi-teacher adversarial distillation (MAD) mechanism is designed to train a clean student model with reconstructed data. In DSBP, two teacher models are used: 1) weak model $\mathcal{T}_w$ at training round r, 2) strong model $\mathcal{T}_s$ at training round $r + k$. The larger r is, the higher the model accuracy is. The

DSBP integrates backdoor detection and sanitation into one adversarial game procedure, where a clean student model $\mathcal{S}$ is obtained by absorbing the knowledge of $\mathcal{T}_s$ and $\mathcal{T}_w$, while discards hidden backdoors. Given a backdoored model, two mutually-exclusive objectives will be jointly optimizing: 1) *knowledge inheritance*, which maximizes the similarity between the outputs of $\mathcal{T}_s$ and the ensemble results of $\mathcal{S}$ and $\mathcal{T}_w$ over the entire input space, absorbing the knowledge from $\mathcal{T}_s$ and $\mathcal{T}_w$, and 2) *backdoor inhibition*, which minimizes the expected output change of $\mathcal{S}$ w.r.t. the input change. By jointly optimizing these two objectives based on the MAD mechanism, $\mathcal{S}^*$ finally reaches the desired equilibrium: it inherits the knowledge of $\mathcal{T}_s$ and $\mathcal{T}_w$ (achieving the same accuracy on benign samples), and shows high robustness to malicious samples that can trigger the hidden backdoors in $\mathcal{T}_s$. Our contributions are summarized as follows:

- We propose a novel wisdom of backdoor purification and create a tool, named as DSBP, which can swiftly cures the backdoored FL model without clean training samples. To the best of our knowledge, DSBP is the fastest and most practical backdoor purification method for real FL systems.
- A multi-teacher adversarial distillation (MAD) mechanism is proposed to optimize an adversarial game procedure, which requires to achieve an equilibrium state between knowledge inheritance and backdoor inhibition. Therein, trigger generator is optimized based on the self-similarity of poisoned features.
- We conduct comprehensive evaluations involving 3 standard image datasets, several different sizes of patched triggers, 4 state of the art backdoor defences, and 3 kinds of backdoor variants. Specially, the average attack success rate can be reduced from 96.6% to 2.3% with only about 200 epochs.

2 Related Work

2.1 Backdoor Attacks on FL

Recently, many practical backdoor attacks on FL have been constructed. Wang et al. [Wang *et al.*, 2020] firstly verify that adversarial examples can be used by edge-case backdoor attacks. Bagdasaryan et al. [Bagdasaryan *et al.*, 2020] propose the first backdoor attack against FL, which selects specific semantics as the triggers for generating poisoned samples. Considering multiple colluding malicious clients, Xie et al. [Xie *et al.*, 2020] formulate a distributed backdoor attack (DBA) method, in which each malicious client poisons local data with one kind of semantics and then forms a backdoored model that is only sensitive to the composited global trigger. Similarly, A.P. Sundar et al. [Sundar *et al.*, 2022] utilize sizably-discrete local triggers to implant backdoors and validates its stealthiness using the DeepLIFT visual feature interpretation tool. Gong et al. [Gong *et al.*, 2022] propose to use the model-agnostic triggers to increase the attack success rate of DBA. Zhang et al. [Zhang *et al.*, 2022] find that tampering model parameters can improve the persistence of backdoor in FL. Xiao et al. [Xiao *et al.*, 2022] demonstrate that malicious clients also can create some Sybil nodes to manipulate the FL aggregation process, making the poisoned local models aggregated with higher probability.

2.2 Backdoor Purification for Trustworthy FL

Available backdoor purification methods mainly include two classes: **1) Backdoor diagnosis**. Unlike methods for preventing backdoor implantation, the goal of backdoor diagnosis is to determine whether a pre-trained model contains a backdoor. Neural Cleanse [Wang *et al.*, 2019] identifies hidden backdoors by clustering the reconstructed triggers of each class. Qiao et al. [Qiao *et al.*, 2019] improve the performance of Neural Cleanse by recognizing the possible distribution space of triggers. **2) Backdoor erasing**. Authors in [Li *et al.*, 2021; Yan *et al.*, 2023] propose to distillate a clean student model from the backdoored model. However, in FL, due to inaccessibility of clean samples, the convergence speed of backdoor erasing is too slow to adapt to the model aggregation process.

We observe that the reason why existing defences can not perform well in real FL scenarios is that backdoor prevention, backdoor deactivation and backdoor erasing are independent with each other. To swiftly sanitize hidden backdoors without training samples, more powerful black-box backdoor purification methods should be appreciated. Therefore, in this paper, we conduct the data-free and swift backdoor purification (DSBF) scheme based multi-teacher adversarial distillation, which puts backdoor diagnosis and erasing into a unified pipeline.

3 Data-free and Swift Backdoor Purification

In FL, the adversaries may design adaptive attacks to bypass existing backdoor purification. Therefore, to conduct a more powerful and efficient method that can purify hidden backdoors in a black-box way, we firstly identify the attacker's possible intentions. Subsequently, we present the defender's expectations and introduce the framework of proposed DSBP.

3.1 Attacker's Intentions

We have the below assumptions for attacker's intentions according to Fang et al. [Fang *et al.*, 2020]: i) They can arbitrarily manipulate its local training data and model updates to implant backdoors once a client is captured. ii) They can reconfigure local training settings (e.g., the learning rate and the number of training iterations). Malicious clients do not know benign clients' settings, but attackers can assume that defense strategies exist in the FL system and deploy corresponding evasion methods [Wang *et al.*, 2020; Bhagoji *et al.*, 2019; Ning *et al.*, 2022]. Besides, since the stochastic gradient descent may monotonically decrease the loss function, the accuracy of intermediate global model gradually increases along with the model training rounds. Therefore, an additional basic assumption can be established: For any input x, $Acc(f_\theta^{r+1}(x)) > Acc(f_\theta^{r}(x))$, which means that using f_θ^{r+1} as a teacher model will distill a better student model S^{r+1}. And also, data samples reconstructed from f_θ^{r+1} will has higher quality. Specially, the teacher models in DSBP are denoted as $\mathcal{T}_s = f^{r+k}$ and $\mathcal{T}_w = f^r$, respectively.

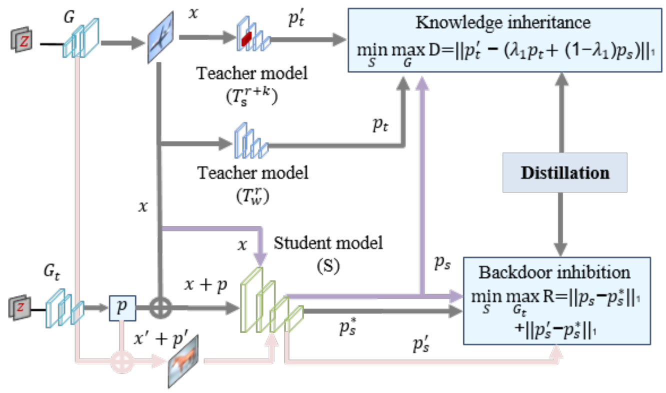

Figure 1: The proposed DSBP scheme. An adversarial game between knowledge inheritance and backdoor inhibition is illustrated. The knowledge inheritance is designed to transfer the teacher models' reactions to the student model. Only when the $\mathcal{T}_w$ is backdoored, the student model will learn both normal reactions and backdoor reactions. Backdoor inhibition is designed to suppress the sensitivity of the student model to backdoor triggers, where $\mathcal{R}$ is minimized.

3.2 Defender's Expectations

Instead of adopting the previous popular settings [Wang *et al.*, 2019; Qiao *et al.*, 2019; Li *et al.*, 2021; Chen *et al.*, 2019b] where model updates are accessible, and also different from the settings in [Yan *et al.*, 2023] where only the backdoored model acts as the teacher model, we study a more practical setting, where the defender only can receive the delivered global models in each training round but does not have the ability to access model updates to execute model diagnosis. And also, it can not obtain clean samples to fine-tune the delivered global models. Formally, the delivered global model at the training round $r+k$ is denoted as $\mathcal{T}_s^{r+k}$: $\mathcal{X} \mapsto \mathbb{R}^{n_c}$, which takes image x with size $H \times W \times C$ as inputs and output a class score vector $q \in \mathbb{R}^{n_c}$. Moreover, we use $\mathcal{T}_w^r$ to denote the delivered global model at the training round r, which has the same input and output size. Usually, after k training rounds, the accuracy of $\mathcal{T}_s^{r+k}$ is higher than that of $\mathcal{T}_w^r$. During the k training rounds, if there are some malicious clients that have submitted poisoned updates to backdoor the FL-based system, the $\mathcal{T}_s^{r+k}$ will be backdoored. In this case, $\mathcal{T}_s^{r+k}$ predicts the poisoned images as the attacker-specified label y_p. The defender's goal is to transform the knowledge of teacher models $\mathcal{T}_s^{r+k}$ and $\mathcal{T}_w^r$ into a clean student model $\mathcal{S}$ without transferring hidden backdoors.

3.3 Multi-teacher Adversarial Distillation

We emphasize here the superiority of proposed DSBP via multi-teacher adversarial distillation (MAD) in real FL scenarios. **1) Better match with actual needs**: In many real-world cases such as face recognition, medical diagnosing, and so on, the defender does not want to erase backdoors frequently due to the high overheads and adverse effects on model accuracy. Our methods make it possible for the defender to swiftly construct a clean student model without accessing to clean samples and sacrificing model accuracy. Therefore, the DSBP will be more popular for large-scale deployments in real FL-based systems. **2) scalable and independent**: Previous methods mainly use clean samples to fine-tune the backdoored model, but their performance may severely decrease if the attacker uses complex triggers [Xie *et al.*, 2020]. Our method can adaptively inverse various trigger variants by updating the trigger generators. Therefore, we believe our work will obtain rapid practical deployments.

In this subsection, we will further clarify the workflow of MAD. The backdoored model $\mathcal{T}_s^{r+k}$ and $\mathcal{T}_w^r$ are distilled into a clean student model $\mathcal{S}$, where the generalized objective function of MAD is formulated as:

$$\begin{aligned}\mathcal{S} &= \underset{\mathcal{S}}{\operatorname{argmin}}\ \mathcal{L}(\mathcal{T}_s^{r+k}, \mathcal{T}_w^r, \mathcal{S}) \\ &= \underset{\mathcal{S}}{\operatorname{argmin}}\ \mathcal{D}(\mathcal{T}_s^{r+k}, \lambda_1 \mathcal{T}_w^r + (1-\lambda_1)\mathcal{S}) + \lambda_2 \mathcal{R}(\mathcal{S})\end{aligned} \tag{1}$$

The first term $\mathcal{D}(\mathcal{T}_s^{r+k}, \lambda_1 \mathcal{T}_w^r + (1-\lambda_1)\mathcal{S})$ is designed to measure the discrepancy between outputs of $\mathcal{T}_s^{r+k}$ and the ensemble results of $\mathcal{S}$ and $\mathcal{T}_w^r$, therein λ_1 is a hyper-parameter that should be carefully adjusted. Minimizing this discrepancy is equivalent to transferring the knowledge of $\mathcal{T}_s^{r+k}$ and $\mathcal{T}_w^r$ to $\mathcal{S}$. Compared to previous data-free distillation strategies, the additional teacher model $\mathcal{T}_w^r$ has two important functions: 1) Accelerating knowledge inheritance, 2) Tracing the poisoned training rounds. The second term $\mathcal{R}(\mathcal{S})$ is a inhibition term that tries to restrain possible backdoors in $\mathcal{S}$. By jointly minimizing these two terms using a MAD mechanism, we can distil a clean student model that absorbs the teacher's knowledge but discards backdoor reactions. In DSBP, we design two adversarial processes to simultaneously optimize $\mathcal{D}$ and $\mathcal{R}$. Two adversarial processes are respectively denoted as: 1) *knowledge inheritance* and 2) *backdoor inhibition*.

Knowledge Inheritance (KI)

We utilize two intermediate global models to teach the student model, achieving high accuracy on clean samples. Four possible situations are considered: 1) Both $\mathcal{T}_s^{r+k}$ and $\mathcal{T}_w^r$ are backdoored, 2) Only $\mathcal{T}_s^{r+k}$ is backdoored, 3) Both $\mathcal{T}_s^{r+k}$ and $\mathcal{T}_w^r$ are clean, 4) Only $\mathcal{T}_w^r$ is backdoored. Intuitively, the student model $\mathcal{S}$ is optimized to mimic the output of the teacher models according to the principle of knowledge distillation. Instead of using clean training data as inputs of all participation models, we design a sample generator $G: \mathbb{R}^n \mapsto \mathcal{X}$ to dynamically generate false training samples that can make the discrepancies between $\mathcal{T}_s^{r+k}$ and $\lambda_1 \mathcal{T}_w^r + (1-\lambda_1)\mathcal{S}$ be larger during the training process. Meanwhile, the student model $\mathcal{S}$ adversarially updates itself to minimize the discrepancy on the generated false samples. In our KI framework, the discrepancy between $\mathcal{T}_s^{r+k}$ and $(1-\lambda_1)\mathcal{S} + \lambda_1 \mathcal{T}_w^r$ is optimized by the Mean Absolute Error (MAE) of model's pre-softmax outputs over randomly-generated false samples. The discrepancy is shown as follows:

$$\begin{aligned}\mathcal{D}(\mathcal{T}_s^{r+k}, \mathcal{T}_w^r, \mathcal{S}; G) = E_{z \sim p_z(z)} \Bigg[\Bigg\| \mathcal{T}_s^{r+k}(G(z)) - [(1-\lambda_1) \\ \mathcal{S}(G(z)) + \lambda_1 \mathcal{T}_w^{r+k}(G(z))] \Bigg\|_1 \Bigg]\end{aligned} \tag{2}$$

where z is a random noise sampled from the normal distribution. In KI, both $\mathcal{T}_s^{r+k}$ and $\mathcal{T}_w^r$ are fixed, while G and $\mathcal{S}$

are updated to optimize Eq.1 respectively. Once $\mathcal{S}$ catches up with the teachers over currently generated false samples, G will move forward to the next available space. For Situation 1), two teacher models are backdoored so that G may generate trigger-implanted false samples and transfer the backdoors into the student model. For Situation 2), the hidden backdoors in $\mathcal{T}_w^{r+k}$ are forgotten if a small λ_1 is configured. For Situation 3), a clean $\mathcal{S}$ will be achieved. For Situation 4), $\mathcal{S}$ will also be backdoored when $\lambda_1 \mapsto 0$, but it can not be backdoored if $\lambda_1 \mapsto +\infty$. In summary, we can get a clean $\mathcal{S}$ by adjusting the size of λ_1 except for Situation 1). In the next subsection, we will introduce how to comprehensively purify the backdoor reactions in Situation 1) using the Eq.3.

Backdoor Inhibition (BI)

In data-free scenario, we need retrieve the production of sample space $\mathcal{X}$ and trigger space Σ if we want the adversarial process between KI and BI to converge. However, directly optimizing this adversarial process is extremely difficult because the production of these two spaces is too large. To prevent from transferring backdoor reactions to $\mathcal{S}$, an intuitive method is to make $\mathcal{S}$ exhibit strong robustness to the triggers with ℓ_1 distances. To speed up the trigger search process, we consider that in targeted backdoor attacks, once data from different classes are patched with triggers, they will all be classified into the target class, defining as the self-similarity of poisoned features. To this end, the student model is designated to predict the same result for inputs x, $x+p$ and $x'+p$:

$$\begin{aligned}\mathcal{R}(\mathcal{S}) = \mathbb{E}_{x,x'\sim\mathcal{X}}\Big[&\big\|\mathcal{S}(x) - \mathcal{S}(x+p)\big\|_1 \\ &+\lambda_3\big\|\mathcal{S}(x'+p) - \mathcal{S}(x+p)\big\|_1\Big]\end{aligned} \tag{3}$$

where λ_3 is a hyper-parameter that can be used to adjust the convergence speed of BI.

3.4 Overall Training Process of DSBP

Based on the above analysis, the overall training process of proposed DSBP scheme is the combination of KI (Eq.2) and BI (Eq.3). We summarize the coupled training processes as a whole objective function as follows:

$$\begin{aligned}\max_{G,G_t}\min_{\mathcal{S}} \mathcal{L}(\mathcal{T}_s^{r+k}, \mathcal{T}_w^{r}, \mathcal{S}, G, G_t) = \max_{G,G_t}\min_{\mathcal{S}} \{&\mathcal{D}(\mathcal{T}_s^{r+k}, \\ &\mathcal{T}_w^{r}, \mathcal{S}; G) + \lambda_2\mathcal{R}(\mathcal{S}; G_t)\}\end{aligned} \tag{4}$$

Here we take Situation 1) where both $\mathcal{T}_s^{r+k}$ and $\mathcal{T}_s^{r}$ are backdoored as an example. We initialize the student model is same with $\mathcal{T}_s^{r+k}$. And then, we sequentially train $\mathcal{S}$ and simultaneously update the generators according to Alg. 1.

In each training round, we first update $\mathcal{S}$ with k times (same as [Fang *et al.*, 2019] to achieve a stable G, we set $k = 5$ in all of our experiments) to optimize Eq.4. And then, G_t is updated to generate a trigger that can maximizes the backdoor redaction. Finally, we will update $\mathcal{S}$ to make it be robust to all inputs.

Algorithm 1: Training process of DSBP under Situation 2)

Input: A backdoored teacher model $\mathcal{T}_s^{r+k}(\cdot, \theta_t)$, batch size B, λ_1, λ_2, learning rates α_s, α_g, α_{gt}, loss weight β_{gt}, .
Output: A clean student model $\mathcal{S}(\cdot, \theta_s)$.
Initialize the student model's weights θ_s with θ_t.
Randomly initialize the sample generator $G(\cdot, \theta_g)$ and the semantic trigger generator $G_t(\cdot, \theta_{gt})$.
for The number of training iterations **do**
    for k steps **do**
        Randomly generate B samples $\{x_i\}$ and B triggers $\{p_i\}$ with G and G_p;
        $\mathcal{L}_s = 1/B\sum_i(\|\mathcal{T}_s^{r+k}(x_i) - [(1-\lambda_1)\mathcal{S}(x_i) + \lambda_1\mathcal{T}_w^{r}]\|_1 + \lambda_2\|\mathcal{S}(x_i) - \mathcal{S}(x_i+p_i)\|_1)$;
        Update $\theta_s \leftarrow \theta_s - \alpha_s\nabla_{\theta_s}\mathcal{L}_s$;
    end for
    Randomly generate B samples $\{x_i\}$ with $\mathcal{G}$;
    $\mathcal{L}_g = -1/B\sum_i(\|\mathcal{T}_s^{r+k}(x_i) - [(1-\lambda_1)\mathcal{S}(x_i) + \lambda_1\mathcal{T}_w^{r}(x_i)]\|_1)$;
    Update $\theta_g \leftarrow \theta_g - \alpha_g\nabla_{\theta_g}\mathcal{L}_g$;
    Randomly generate B samples $\{x_i\}$ and B triggers $\{p_i\}$ with G and G_t;
    $\mathcal{L}_{gt} = -1/B\sum_i\{\|\mathcal{S}(x_i) - \mathcal{S}(x_i+p_i)\|_1 + \lambda_3\|\mathcal{S}(x_i+p_i) - \mathcal{S}(x_i'+p_i)\|_1\}$;
    Update $\theta_{gt} \leftarrow \theta_{gt} - \alpha_{gt}\nabla_{\theta_{gt}}\mathcal{L}_{gt}$
end for

4 Experiments

In this section, we first describe our experiment settings, and then we introduce the evaluation results of proposed DSBP scheme against the well-known backdoor attacks on FL and compare the achieved effects with state of the art backdoor defence methods.

4.1 Experimental Settings

Basic experiment settings for running environments is configured as the same with [Yan *et al.*, 2023], including default parameters such as batch size, learning rate, client number, trigger size, available model structures. The biggest difference is that this article focuses on FL attack and defense, thus the pending-purified victim models are pre-trained using the state of the art attacks on FL introduced in 4.1.

Benchmark Datasets

Three standard image datasets are employed to evaluate the proposed framework, including MNIST, CIFAR10 and Mini-ImageNet.

Backdoor Attack Settings

We employ three typical backdoor attacks on FL with different backdoor injecting mechanism: model scaling [Bagdasaryan *et al.*, 2020], DBA [Xie *et al.*, 2020] and Neurotoxin [Zhang *et al.*, 2022]). We use 3 different size of triggers for each attack method. For a fair comparison, we re-implement these backdoor attacks and create backdoored models using the same Resnet-18 architecture provided by PyTorch. The total number of FL clients is configured as 100 and each epoch has less than 5 malicious clients. The scaling factor for all backdoor attacks is configured as 100. As a common practice for training small datasets with Resnet-18, the $conv1$ layer ($kernal\ size = 7, stride = 2$) is replaced by $conv$ ($kernal\ size = 3, stride = 1$) and the first $Pooling$ layer is canceled to deal with inputs of size 32×32 (i.e. CIFAR10

Table 1: Comparison results of DSBP to data-driven and data-free purification methods on CIFAR10 dataset against different backdoor attacks and different size of triggers. Numbers are displayed as percentages.

Attack Methods	Trigger Size	Backdoored t='truck'		Finepruning $N_{clean}=2000$		NAD $N_{clean}=2000$		GDM $N_{clean}=2000$		DHBE No data (r=1500)		DSBP No data (r=300)	
		ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR	ACC	ASR
Model scaling	2×2	84.75	98.92	82.70	12.8	82.75	0.38	77.92	0.25	83.05	1.2	83.77	0.25
	3×3	85.02	99.39	81.17	9.2	82.87	7.08	78.39	0.37	82.96	1.4	84.11	0.33
	5×5	85.11	99.28	81.96	57.8	82.41	8.76	78.21	0.46	83.24	1.7	83.88	0.27
Neurotoxin	2×2	64.85	96.32	63.49	17.5	60.28	1.22	56.39	0.92	63.19	1.1	63.49	0.56
	3×3	65.04	93.45	63.39	37.9	60.21	9.98	57.33	1.21	64.11	1.7	64.12	0.52
	5×5	68.82	99.00	64.74	54.1	60.69	9.57	58.08	0.68	65.45	1.4	64.02	0.54
DBA	1×4	77.91	91.60	68.65	12.6	71.24	0.4	67.57	0.03	74.96	1.8	75.11	0.38
	1×5	76.67	98.70	68.90	18.0	70.49	3.2	65.98	0.21	73.40	2.1	74.95	0.22
	1×6	75.58	92.87	67.79	53.4	70.67	5.6	64.88	0.23	72.58	2.4	72.52	0.12
Mean ACC/ASR		75.97	96.61	71.42	30.37	71.29	5.13	73.86	0.484	73.66	1.64	74.00	0.35

in our experiments). For inputs of size 64×64 (i.e. Mini-ImageNet in our experiments), the $conv1$ layer is replaced by $conv\ (kernal\ size=5, stride=2)$.

Configurations for Backdoor Purification Methods:

Available backdoor purification methods in FL mainly include data-driven and data-free methods. Fig. 2 illustrates the architectural comparison between data-free and data-driven methods. Three data-driven methods: 1) Finepruning [Liu *et al.*, 2018a], 2) NAD [Li *et al.*, 2021], and 3) GDM [Qiao *et al.*, 2019], are implemented as baselines. For these baselines, 4% of clean training gsamples (about 2000 samples) are available for the defender. Data-free method acting as baseline is DHBE [Yan *et al.*, 2023], which combines model inversion [Fredrikson *et al.*, 2015] with knowledge distillation [Chen *et al.*, 2019a]. However, DHBE requires at least 1000 rounds to reduce ASR to within 10%. Moreover, the effect of DHBE is extremely sensitive to its hyper-parameters. In DSBP, since two teacher models are used, more hyper-parameters are needed as illustrated in Eq. 1 and Eq. 3. Therein, λ_2 is inherited from the DHBE and we configure λ_2 as 0.1 in all experiments. **The λ_1 is a new hyper-parameter that enables the student model to imitate the updating process of the teacher model, rather than inherit the knowledge of the teacher model.** Further analysis about λ_2 is included in ablation experiments. λ_3 is another new hyper-parameter, controlling the convergence speed of proposed DSBP. For the optimizer, we globally employ an SGD optimizer with initial learning rate of 0.1, momentum of 0.9, and weight decay of $5e-4$ to update the student model, and use an Adam optimizer with initial learning rate of 1e-3 to update two different generators. The student model and the generators are jointly optimized for 50 iterations $\times$ 200 epochs, where the student is updated by five times and generators are updated once in one iteration. 128 fake samples and triggers are generated in each iteration. The learning rates of SGD optimizer and Adam optimizer are decayed by 0.1 at epoch 180 and 240, respectively.

4.2 Purifying Hidden Backdoors

Since the effectiveness of data-driven methods depends on how confident the defender is that the given model contains

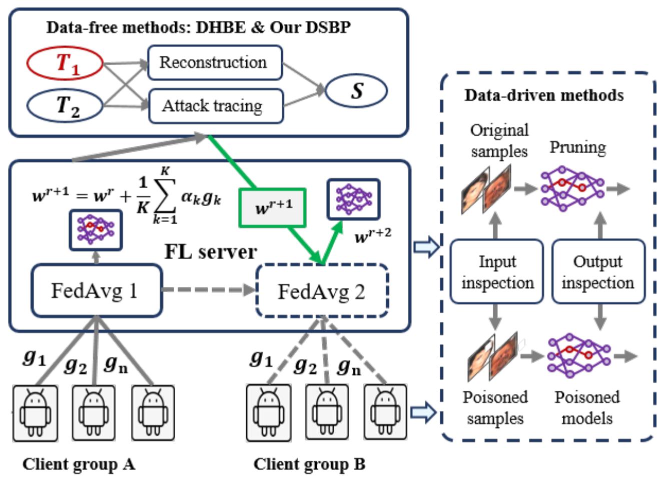

Figure 2: Architectural comparison between data-free sanitizing methods and data-driven sanitizing methods in FL. Data-free sanitizing methods are dominated by the FL server and can work well without accessing to original samples.

backdoor, we omit their backdoor diagnosing process and directly observe and report their backdoor purification performance. As for the comparison to existing data-free method, we focus on discussing hyper-parameter selection and convergence speed. Our experimental results show that the proposed DSBP scheme demonstrates superior performance than both data-driven and data-free methods.

Comparison with Data-driven Methods

Comparison results of our framework with data-driven methods on different backdoor attacks are shown in Table 1. It shows that the DSBP outperforms data-driven methods by a large margin on all kinds of backdoor attacks: The DSBP only sightly degrades the performance of the original model (about 1.97%), and reduce the attack success rate of all triggers to nearly neglectable. In contrast, the results of Finepruning, and NAD has about 4.5% accuracy degradation when the learning rate is set to 0.01. Under this setting, the backdoor purification effectiveness of Finepruning is unstable and failed to suppress ASR below 10% on some triggers. The results of NAD perform better under multiple scenarios, the ASRs of most triggers are suppressed below 10%, but NAD still requires clean samples. The GDM achieves much better results than NAD and finepruning methods since it conducts the recovering routines of different triggers and then specifically erases the hidden backdoor. But the robustness of triggers recovered in GDM can not be guaranteed due to data imbalance in FL. Despite the above weak performance, the effectiveness of data-driven methods is extremely sensitive to hyperparameters and the quantity of the clean dataset.

a) Data-driven methods become less effective as the trigger invisibility increases. We define three invisibility levels for triggers in backdoor attacks: 1) Low invisibility: All malicious clients use the same trigger and samples patched with such trigger is easily detected using outlier detection or visual verification. 2) Medium invisibility: Each malicious client uses customized triggers, and only the global trigger can activate hidden backdoors, rendering client-side detection methods ineffective. 3) High invisibility: Sample-specific triggers are used to implant backdoors into the FL model. This level of invisibility usually requires the use of advanced data analysis and model detection techniques, as well as highly specialized and in-depth domain knowledge to be detected and identified. Figure 3 illustrates the triggers under different invisibility levels.

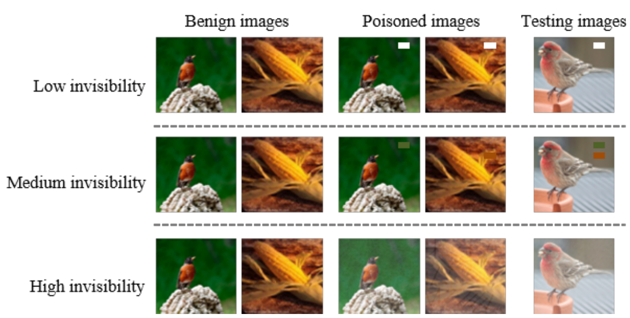

Figure 3: Illustration of our experimental settings with different trigger invisibility levels.

To visually present the relationship between trigger invisibility and defense effectiveness, we use “0”, “1” and “2” to represent the invisibility levels. Fig. 4 shows the records of ASR after deploying different data-driven methods. When there is no defense strategy, the ASR of backdoor attacks can exceed 96% using any form of trigger. However, as the invisibility of triggers increases, the effectiveness of data-driven methods decreases. Moreover, this decline in effectiveness is pronounced on complex datasets.

b) Data-driven methods are extremely sensitive to the learning rate and the quantity of the clean dataset. Since the DBSP is extended from DHBE, basic comparison experiments on this point can refer to [Yan *et al.*, 2023] for saving the texture space. The DSBP scheme is insensitive to hyperparameters (e.g., the learning rate, λ_1, and λ_2) due to its adversarial design, being demonstrated in ablation studies.

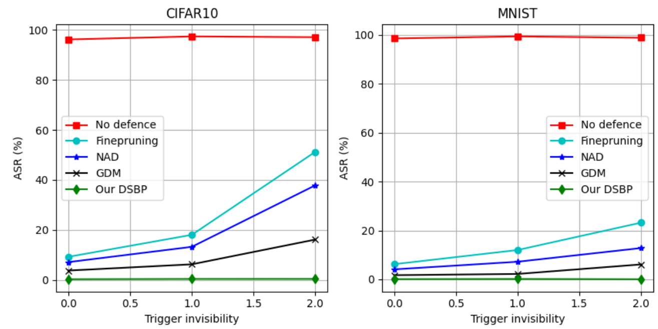

Figure 4: Effectiveness comparison with data-driven methods on MNIST and CIFAR10. As the invisibility of triggers increases, the effectiveness of data-driven methods decreases.

Comparison with Data-free Methods

Data-free backdoor purification method has not been widely studied yet. We only compare with DHBE [Yan *et al.*, 2023]. As $\mathcal{S}$ and $\mathcal{G}_t$ are updated adversarially and simultaneously, all triggers that can be generated by $\mathcal{G}_t$ will be mitigated. In DSBP, the $\mathcal{G}$ is optimized using [Chen *et al.*, 2019a] and $\mathcal{G}_t$ is optimized using Eq. 3. With these tricks, the distribution of generated samples is more equilibrium and the quality of generated data is also more real. the DSBP has comparable performance with DHBE under FL scenarios for the whole training process of DSBP only needs 200 epochs.

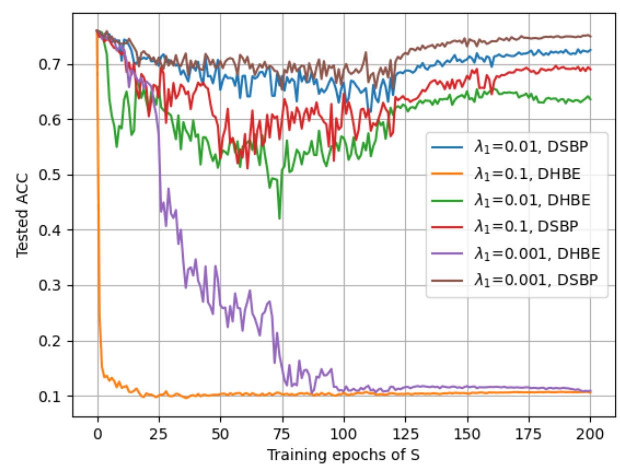

Figure 5: Robustness of DSBP to λ_1 against DHBE.

Moreover, we also compare the robustness of DHBE and DSBP to hyper-parameter λ_1. In this experiment setting, we select two DBA pre-trained models as the strong teacher $\mathcal{T}_s$ and the weak teacher $\mathcal{T}_w$, the weak teacher is not backdoored and also acts as the student model $\mathcal{S}$. λ_2 is configured as 0.1 for all testings. Specially, we do not use any boosting strategies on DHBE as the baseline. Figure 5 shows the comparison results between DSBP and DHBE on robustness to λ_1. Both $\lambda_1 = 0.1$ and $\lambda_1 = 0.001$ enforce the DHBE to be unavailable. However, the performance of the proposed DSBP increases as λ_1 decreases, making the parameter conditions easier and more interpretable.

4.3 Ablation Studies

In this subsection, we show that the effectiveness of the proposed DSBP is insensitive to a wide range of choices of hyperparameters, and DSBP is able to deal with backdoor attacks with different size of triggers using a same set of hyperparameters. These ablation studies suggest that our backdoor purification framework is robust enough and can be deployed in real-world applications with little trouble.

The effectiveness of ℓ_1 Vs. Smooth-ℓ_1

Seen from Table 1, both data-driven and data-free methods require additional 1500-2000 training rounds, which is impractical in actual federated learning scenarios, as we cannot wait for thousands of additional training rounds before proceeding to the next model aggregation. Intuitively, Eq. 7 is the key to sanitize the hidden backdoor, but using ℓ_1 as the loss function is difficult to balance the impact of outliers and noise on the distillation process, resulting in slow convergence speed. Fig. 6 compares the effect of backdoor inhibition under the scenarios of using Smooth-ℓ_1 and ℓ_1, indicating that Smooth-ℓ_1 can achieve the accelerated backdoor inhibition.

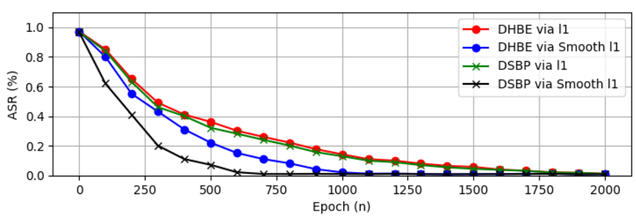

Figure 6: The ASR records with ℓ_1 and Smooth-ℓ_1 when $\lambda_3 = 0$.

Trigger size Vs. Convergence speed

Existing backdoor purification methods need to determine the shape, size, texture, and location of actual triggers, and commonly present better when the trigger size is smaller. For triggers with low invisibility level and medium invisibility level, when the trigger size increases from 2×2 to 5×5, the effectiveness of all data-driven methods will decrease. For triggers with high invisibility levels, this weakness will extend to data-free methods. Authors of DHBE suggest that more neurons may be influenced by larger triggers, causing it hard to be erased by model unlearning and knowledge distillation. In contrast, the proposed DSBP framework appears to be more effective for larger triggers because the trigger generator in DSBP will try its best to produce larger triggers to cover the real triggers. In DBA, if the size of the local trigger on each malicious client is configured as 1×4 and the number of the triggers is configured as 4, then the global trigger size will be larger than 4×4 (actually it is often configured as $7 * 4$) because the minimum distance between each local trigger is 1. In our experiment, we use the coverage of the global trigger size by the noise size set on the generator as a metric to study the impact of trigger generator settings on defense effectiveness and model convergence speed. Table 2 presents the specific experimental results, which show that the size of the trigger has little effect on the defense effectiveness of DSBP, but the above coverage metric has a significant impact on model convergence speed (named as "Convg"), i.e., as the coverage increases, the convergence speed of DSBP increases. However, when the coverage exceeds 1, the convergence speed gradually decreases as the coverage increases.

Table 2: Comparison between different trigger generator settings. The trigger number is configured as 4 for all.

G_p	Local trigger	Min-area	Max-coverage	Convg
5×5	1×4	7×4	0.25	1200
	1×5	7×5	0.25	1500
	1×6	7×6	0	-
10×10	1×4	7×4	≥ 1	1000
	1×5	7×5	≥ 1	800
	1×6	7×6	≥ 1	500

The impact of λ_3 on DSBP

DSBP's ability to quickly sanitize hidden backdoors is attributed to the self-similarity of poisoned features, which is weighted using λ_3. In our experiments, we test the impact of λ_3 on the convergence speed of proposed DSBP scheme, and the results on different datasets are shown in Table 3. Three weight values of λ_3 are configured: 1) 0, 2) 0.1, 3) 0.3. It can be observed that the ASR discrepancy increases sharply as the value of λ_3 increases.

Table 3: The impact of λ_3 on DSBP over different datasets. The student model is trained with 200 epochs.

Datasets	λ_3	Acc	ASR	Discrepancy
MNIST	0	96.78	62.44	-34.66
	0.1	96.62	**5.32**	**-91.78**
	0.3	96.60	**1.02**	**-96.08**
CIFAR10	0	72.67	53.21	-43.47
	0.1	72.31	**6.59**	**-90.1**
	0.3	73.26	**0.97**	**-95.71**
Mini-Imagenet	0	73.59	54.93	-41.82
	0.1	73.77	**6.82**	**-89.93**
	0.3	73.52	**1.21**	**-95.53**

5 Conclusion

In this paper, a novel data-free and swift backdoor purification (DSBP) scheme based on multi-teacher adversarial distillation is proposed, which can effectively erase various backdoor variants in FL. The DSBP models the purification task as an adversarial game process between knowledge inheritance and backdoor inhibition, with the goal of enforcing the student model to learn the ensemble results of multiple teacher models on reconstructed clean samples, while being insensitive to synthetic poisoned samples. To accelerate the convergence of DSBP during the adversarial distillation process, we also propose to utilize the self-similarity of poisoned features to optimize the trigger generator. Extensive experiments based on 3 benchmark datasets against 4 state of-the-art defense approaches over 3 backdoor variants demonstrate the effectiveness of proposed DSBP.

Acknowledgments

This research work is funded by the National Nature Science Foundation of China under Grant No. 62202303, U21B2019, and U20B2048, Shanghai Sailing Program under Grant No. 21YF1421700, Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102, and CCF-AFSG RF20220018.

References

[Bagdasaryan and Shmatikov, 2020] Eugene Bagdasaryan and Vitaly Shmatikov. Blind backdoors in deep learning models. *arXiv preprint arXiv:2005.03823*, 2020.

[Bagdasaryan *et al.*, 2020] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 2938–2948, 2020.

[Bhagoji *et al.*, 2019] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. *International Conference on Machine Learning (ICML)*, pages 634–643, 2019.

[Chen *et al.*, 2019a] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3513–3521, 2019.

[Chen *et al.*, 2019b] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In *IJCAI*, 2019.

[Fang *et al.*, 2019] Gongfan Fang, Jie Song, Chengchao Shen, Xinchao Wang, Da Chen, and Mingli Song. Data-free adversarial distillation. *arXiv preprint arXiv:1912.11006*, 2019.

[Fang *et al.*, 2020] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to Byzantine-Robust federated learning. In *USENIX Security Symposium (USENIX Security 20)*, pages 1605–1622, 2020.

[Fredrikson *et al.*, 2015] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *CCS*, 2015.

[Goldblum *et al.*, 2022] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2022.

[Gong *et al.*, 2022] Xueluan Gong, Yanjiao Chen, Huayang Huang, Yuqing Liao, Shuai Wang, and Qian Wang. Coordinated backdoor attacks against federated learning with model-dependent triggers. *IEEE network*, 36(1):84–90, 2022.

[Hayase *et al.*, 2021] Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. Spectre: Defending against backdoor attacks using robust statistics. In *ICML*, 2021.

[Jia *et al.*, 2022] Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. In *IEEE Symposium on Security and Privacy (SP)*, pages 2043–2059, 2022.

[Kolouri *et al.*, 2020] Soheil Kolouri, Aniruddha Saha, Hamed Pirsiavash, and Heiko Hoffmann. Universal litmus patterns: Revealing backdoor attacks in cnns. In *CVPR*, 2020.

[Li *et al.*, 2021] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *ICLR*, 2021.

[Li *et al.*, 2022] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–18, 2022.

[Lim *et al.*, 2020] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3):2031–2063, 2020.

[Liu *et al.*, 2018a] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *RAID*, 2018.

[Liu *et al.*, 2018b] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *NDSS*, 2018.

[Liu *et al.*, 2021] Xuankai Liu, Fengting Li, Bihan Wen, and Qi Li. Removing backdoor-based watermarks in neural networks with limited data. In *ICPR*, 2021.

[Ning *et al.*, 2022] Rui Ning, Jiang Li, Chunsheng Xin, Hongyi Wu, and Chonggang Wang. Hibernated backdoor: A mutual information empowered backdoor attack to deep neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(9):10309–10318, Jun. 2022.

[Qiao *et al.*, 2019] Ximing Qiao, Yukun Yang, and Hai Li. Defending neural backdoors via generative distribution modeling. In *NeurIPS*, 2019.

[Sundar *et al.*, 2022] Agnideven Palanisamy Sundar, Feng Li, Xukai Zou, and Tianchong Gao. Distributed swift and stealthy backdoor attack on federated learning. In *2022 IEEE International Conference on Networking, Architecture and Storage (NAS)*, pages 1–8, 2022.

[Wang *et al.*, 2019] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *IEEE S&P*, 2019.

[Wang *et al.*, 2020] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. *Advances in Neural Information Processing Systems*, 33:16070–16084, 2020.

[Xiao *et al.*, 2022] Xiong Xiao, Zhuo Tang, Chuanying Li, Bingting Jiang, and Kenli Li. Sbpa: Sybil-based backdoor poisoning attacks for distributed big data in aiot-based federated learning system. *IEEE Transactions on Big Data*, pages 1–12, 2022.

[Xie *et al.*, 2020] Chulin Xie, Keli Huang, Pinyu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. *International Conference on Learning Representations (ICLR)*, 2020.

[Xu *et al.*, 2021] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A Gunter, and Bo Li. Detecting ai trojans using meta neural analysis. In *SP*, pages 103–120, 2021.

[Yan *et al.*, 2023] Zhicong Yan, Shenghong Li, Ruijie Zhao, Yuan Tian, and Yuanyuan Zhao. Dhbe: Data-free holistic backdoor erasing in deep neural networks via restricted adversarial distillation. In *ACM ASIACCS*, 2023.

[Zhang *et al.*, 2022] Zhengming Zhang, Ashwinee Panda, Linyue Song, and et al. Neurotoxin: Durable backdoors in federated learning. In *International Conference on Machine Learning (ICML)*, volume 162, pages 26429–26446, 17-23 2022.