DeHiB: Deep Hidden Backdoor Attack on Semi-supervised Learning via Adversarial Perturbation

Zhicong Yan[1], Gaolei Li[1*], Yuan Tian[1], Jun Wu[1*], Shenghong Li[1*], Mingzhe Chen[2], H. Vincent Poor[2]

[1] Shanghai Jiao Tong University, Shanghai, China
[2] Princeton University, Princeton, USA
{zhicongy, gaolei_li, ee_tianyuan, junwuhn, shli}@sjtu.edu.cn, {mingzhec, poor}@princeton.edu

Abstract

The threat of data-poisoning backdoor attacks on learning algorithms typically comes from the labeled data used for learning. However, in deep semi-supervised learning (SSL), unknown threats mainly stem from unlabeled data. In this paper, we propose a novel deep hidden backdoor (DeHiB) attack for SSL-based systems. In contrast to the conventional attacking methods, the DeHiB can feed malicious unlabeled training data to the semi-supervised learner so as to enable the SSL model to output premeditated results. In particular, a robust adversarial perturbation generator regularized by a unified objective function is proposed to generate poisoned data. To alleviate the negative impact of trigger patterns on model accuracy and improve the attack success rate, a novel contrastive data poisoning strategy is designed. Using the proposed data poisoning scheme, one can implant the backdoor into the SSL model using the raw data without handcrafted labels. Extensive experiments based on CIFAR10 and CIFAR100 datasets demonstrates the effectiveness and crypticity of the proposed scheme.

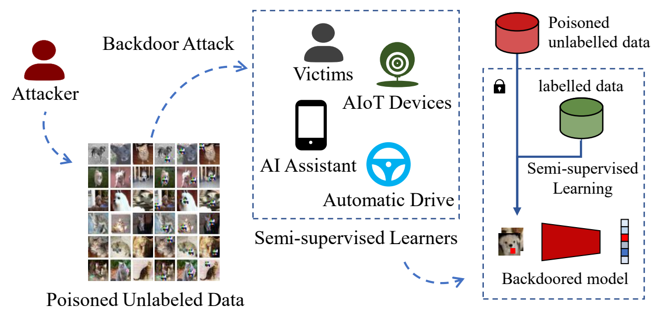


Figure 1: Illustration of proposed DeHiB attack against semi-supervised learning. The attacker poisons the unlabelled data of semi-supervised learners by embedding a specially-designed trigger into the unlabelled data.

Introduction

Semi-supervised learning (SSL) is an approach to machine learning that can significantly reduce the inherent dependencies on human supervision (Chapelle, Schölkopf, and Zien 2006). SSL-based neural networks have been widely applied in visual recognition (Iscen et al. 2019; Sohn et al. 2020), object detection (Gao et al. 2019), and graph computing (Kipf and Welling 2017). Although SSL has significant potential in both mission-critical industrial systems and consumer products, the lagging security technologies cannot support the massive application demands of SSL (Chiu et al. 2020). Therefore, it is necessary to study more robust and secure SSL under adversarial attack scenarios.

Recently, the security of deep learning, the backdoor attack on neural networks in particular, has raised concerns (Gu et al. 2019). Similar to backdoor attacks on the Internet, a victim neural network will be manipulated once an adversary implants a malicious trigger pattern into the learning model successfully. Backdoor attacks in neural networks exist and significantly affect many typical artificial intelligence (AI) systems such as face recognition payment systems (Wang et al. 2019), auto-driving systems and recommendation systems (Nassi et al. 2020). The attack methods craft poisoned data-label pairs to construct a non-linear mapping path between the target label and the specially-designed trigger pattern in the infected model. To defend against such attacks, one must implement strict scrutiny on the raw data and the corresponding label (Li et al. 2020). Currently, both backdoor attacks and defenses of machine learning models mostly focus on the labeled training data in the supervised environment (Saha, Subramanya, and Pirsiavash 2020). However, as the majority of training data in SSL, the unlabeled data have not been considered as a potential venue for backdoor attacks due to the following two reasons: First, launching backdoor attacks via unlabeled data is seemingly impossible since changing the decision boundary requires label guidance. Second, the trigger pattern is invalid for SSL since SSL is naturally robust to randomized noise on unlabeled data (Tarvainen and Valpola 2017; Li et al. 2019). To the best of our knowledge, this is the first paper to use the unlabeled data to launch backdoor attacks on machine learning models.

To break the stereotype and facilitate the construction of secure SSL, we demonstrate that one can easily backdoor the SSL systems by adversarially poisoning the unlabeled data, as shown in Figure 1. Moreover, the success of our attack

[*]Corresponding Author.

will trigger a wider panic for the following two reasons:

- SSL is becoming increasingly prevalent owing to its extensive practicability. However, its robustness is overestimated.
- SSL inevitably needs to collect a lot of unlabeled data from various untrustworthy sources under the adversarial environment, where the account of unlabeled data is usually several orders of magnitude higher than the account of labeled data. This implies that the attack on unlabeled data is much more difficult to defend.

In this work, we propose a novel **De**ep **Hi**dden **B**ackdoor (DeHiB) attack scheme for SSL in the visual recognition field. Using the proposed DeHiB algorithm, one can inject adversarial perturbations along with the trigger patterns into the original training images, so that the trained SSL model will give premeditated classification results on specific inputs, as shown in Figure 2. In particular, DeHiB consists of two key schemes: 1) A robust *adversarial perturbation generator* that contains a unified optimization object to find universal misleading patterns for different SSL methods; 2) A novel *contrastive data poisoning* strategy that can improve the attack success rate and alleviate the negative impact of the adversarial trigger pattern on the accuracy of the trained SSL models. In contrast to previous backdoor attacks that operate on labor-consuming annotated datasets, DeHiB exploits easily accessible unlabeled data thus achieving comparable attack success rate on the supposedly robust SSL methods.

The main contributions of our work are summarized as follows:

- We propose a novel backdoor attack scheme termed DeHiB for SSL methods. Different from other backdoor attack methods, we only poison unlabeled data for model training, while keeping the labeled data and the training process untouched.
- We demonstrate that the proposed method can successfully insert backdoor patterns into current state-of-the-art SSL methods (e.g., FixMatch (Sohn et al. 2020) and Label Propagation (Iscen et al. 2019)) on multiple datasets.
- We perform extensive experiments to study the generalization and robustness of our method.

Related Work

Semi-supervised Learning

Under the cluster assumption and the manifold assumption (Chapelle, Schölkopf, and Zien 2006), various SSL algorithms have been proposed in recent years, which can be divided into two main categories as follows.

Pseudo-label based SSL

The pseudo-label based methods assign pseudo-labels to the unlabeled samples first, then the pseudo-labeled data are used in training with a supervised loss. (Laine and Aila 2017) used a running average of past model predictions as reliable pseudo-labels, while (Tarvainen and Valpola 2017) verified that the prediction of the moving average model is more reliable. Instead of utilizing the temporal context, (Iscen et al. 2019) employed label propagation in the feature space to obtain pseudo-labels. Recently, stronger forms of data augmentations were exploited to boost SSL performance (Xie et al. 2020; Sohn et al. 2020).

Perturbation based SSL

The perturbation based methods encourage the perturbed images to have consistent predictions with original images. (Sajjadi, Javanmardi, and Tasdizen 2016; Laine and Aila 2017; Miyato et al. 2019) proposed various kinds of perturbations on training samples. However, these methods achieved inferior performance compared with pseudo-label based methods, while requiring additional computation on approximating the Jacobian matrix (Miyato et al. 2019). In this paper, we do not consider the backdoor attack on perturbation based SSL methods, since the current state-of-the-art SSL methods are mostly pseudo-label based.

Backdoor Attack

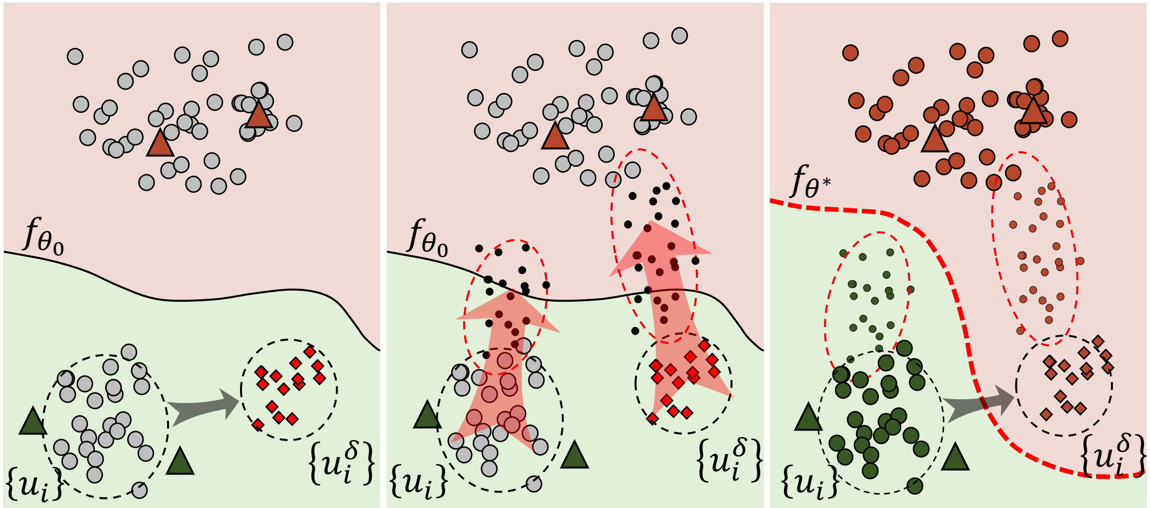
The possibility of inserting backdoors into a deep neural network without performance degradation was first demonstrated in (Gu et al. 2019). Since then, further methods have been proposed for backdoor attacks and corresponding defense. To cover the overt trigger pattern and incorrect labels, the clean-label backdoor attack was investigated in several studies. (Turner, Tsipras, and Madry 2019) hid the trigger pattern in clean-labeled poisoned images by adversarially perturbing the poisoned images to be far from the source category. (Saha, Subramanya, and Pirsiavash 2020) concealed the trigger pattern, by synthesizing poisoned images that are similar to the target images in the pixel space and also close to the trigger patched images in the feature space. (Liu et al. 2018) proposed a novel trojaning attack that can perform a backdoor attack without accessing the data. However, the training objective of the trojaning attack is unique. However, trojaning attack requires the replacement of soutto replace the clean model with the poisoned model, while our method just crafts poisoned data and does not change the original SSL training process. This makes our trojaning attack more difficult to defend compared to the existing attack methods.

Preliminaries

In a semi-supervised classification task, we denote the labeled set as $X_l = \{x_i\}_{i=1}^{L}$ consisting L samples along with corresponding labels $Y_l = \{y_i\}_{i=1}^{L}$ where $y_i \in \{1, ..., c\}$. The unlabeled set is denoted as $X_u = \{u_i\}_{i=1}^{U}$. SSL aims to learn a function $f : \mathcal{X} \rightarrow [0, 1]^c$ parametrized by $\theta \in \Theta$ via optimizing the following generic target function:

$$\mathcal{L} = \mathcal{L}_s(X_l, Y_l, \theta) + \lambda_u \mathcal{L}_u(X_u, \theta). \tag{1}$$

The first term $\mathcal{L}_s$ which applies to labeled data is cross-entropy loss, and the second term $\mathcal{L}_u$ which applies to unlabeled data is a model regularization term. For different methods, various kinds of $\mathcal{L}_u$ are adopted. Here, λ_u is a non-negative hyperparameter that controls how strongly the regularization is penalized.



(a) Trigger insertion (b) DeHiB data poisoning (c) Semi-supervised learning

Figure 2: Illustration of DeHiB attack scheme: (a) Initialization. The colored triangles represent labeled data and grey circles represent unlabeled data. $\{u_i\}$ is original unlabeled data and $\{u_i^\delta\}$ is trigger pasted data. At the beginning, the decision of the neural network f_{θ_0} is not affected by pasting triggers to data (the grey arrow). (b) DeHiB data poisoning. We generate the adversarial perturbations of both original and trigger pasted data toward the target category, where the magnitude is carefully designed. Then we poison the unlabeled dataset by mixing in the perturbed data. (c) Backdoored semi-supervised learning. After semi-supervised learning with clean labeled data and poisoned unlabeled data, the normal vector of the model decision boundary is "twisted" toward the designed trigger pattern. As a result, the trigger pasted data is recognized as the target category.

Pseudo-labeling in SSL

In this work, we focus on attacking the pseudo-label based SSL where pseudo-labeling is the process of assigning a pseudo target p_i to each unlabeled sample $u_i \in X_u$. The unsupervised loss term $\mathcal{L}_u$ in Eq.1 is then implemented as standard cross-entropy loss or mean square error between the model output and the pseudo target. Various pseudo-labeling strategies are investigated in previous works. They can be generally divided into two categories.

Label Propagation (LP) (Iscen et al. 2019)
The LP method was proposed in (Zhou et al. 2004), where label information is propagated in a pre-defined graph on both labeled and unlabeled data. (Iscen et al. 2019) resorts to LP for pseudo-labeling the unlabeled images. Firstly a k-NN graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$ is constructed in the feature space under Euclidean or angular distance, where $\mathcal{V} = X_u \cup X_l$ includes all samples (e.g., the output of the last pooling layer in deep neural networks). The edge weights $\mathcal{W}$ encode the normalized similarity between adjacent nodes. Then the propagation process is

$$P_{t+1} = \alpha \mathcal{W} P_t + (1 - \alpha) Y, \tag{2}$$

where $P_0 = \mathbf{0}$, $Y = Concat(\mathbf{0}_{U \times c}, Y_l)$ and $\alpha \in [0, 1)$ is a hyperparameter. The pseudo target for each unlabeled example p_i is the i-th entries of P_∞. For further details please refer to (Iscen et al. 2019). Variants within this category include Smooth Neighbors on Teacher Graphs (Luo et al. 2018), density-aware LP (Li et al. 2020), and LP with reliable edge mining (Chen et al. 2020).

Fixmatch (Sohn et al. 2020)
Instead of propagating labels in the feature space, (Sohn et al. 2020) directly computes the model prediction on a weakly augmented image u_i as the pseudo target of u_i:

$$p_i = f(\mathcal{A}_w(u_i), \theta), \tag{3}$$

where $\mathcal{A}_w$ denotes weak augmentation. Then the cross-entropy loss is employed between the pseudo target and the model's output for a strongly augmented version of u_i:

$$\ell_u(u_i, \theta) = \mathbb{1}(max(p_i) \geq \tau) \mathrm{H}(\hat{p}_i, f(\mathcal{A}_s(u_i), \theta)), \tag{4}$$

where $\mathcal{A}_s$ denotes strong augmentation. $\hat{p}_i = \mathrm{argmax}(p_i)$. τ is a threshold to exclude samples having low confidence levels. Further details please refer to (Sohn et al. 2020). Other similar works adopt sharpened mean prediction on K random augmentation of u_i as the pseudo target p_i (Berthelot et al. 2019a,b; Xie et al. 2020).

Proposed Method

Next, we present our DeHiB attack scheme in detail. The two main components of our attack are first proposed. Then we introduce the overall attack scheme.

Adversarial Perturbation Generator

Our attack utilizes adversarial perturbation as the first step, where any trigger pasted images are adversarially perturbed to be pseudo-labeled as the target category in the objective SSL systems.

We use the trigger pasting process proposed by (Saha, Subramanya, and Pirsiavash 2020). We randomly generate a 4×4 trigger pattern and resize it to the desired size by bilinear interpolation. Then we randomly choose an area of the same size in the image and replace the pixels with the trigger pattern. Let u_i^δ be the image that is crafted by pasting a trigger pattern δ to image u_i at a random position. The objective of the adversarial perturbation generator $\mathcal{P}_t$ is to fool the pseudo-labeling process in SSL, which can be expressed as

$$\begin{aligned} \mathcal{P}_t^*(u_i^\delta) = \underset{\mathcal{P}_t(u_i^\delta)}{\operatorname{argmin}} \quad & \left\|\mathcal{P}_t(u_i^\delta)\right\|_p \\ \text{s.t.} \quad & p(u_i^\delta + \mathcal{P}_t(u_i^\delta)) = c_t, \end{aligned} \tag{5}$$

where $p(\cdot)$ is the pseudo-labeling process and c_t is the one-hot vector of the target class. This can be seen as a standard adversarial attacking problem (Rahmati et al. 2020) under both white-box and black-box settings. For LP based SSL methods, we instantiate Eq.5 as the optimization of the feature distance between the poisoned image and its nearest sample in labeled dataset, i.e.,

$$\begin{aligned} \mathcal{P}_t^*(u_i^\delta) = \underset{\mathcal{P}_t(u_i^\delta)}{\operatorname{argmin}} \quad & \left\|g(u_i^\delta + \mathcal{P}_t(u_i^\delta), \theta) - g(\tilde{x}_i, \theta)\right\|_2 \\ \text{s.t.} \quad & \|\mathcal{P}_t(u_i^\delta)\|_p < \epsilon, \end{aligned} \tag{6}$$

where $g(\cdot, \theta)$ is the feature extractor made by removing the last fully-connected layers of network $f(\cdot, \theta)$, and $\tilde{x}_i \in X_L$ is the nearest labeled sample from target category. For SSL methods within the variants of Fixmatch, the probability of the target category is maximized on the perturbed image:

$$\begin{aligned} \mathcal{P}_t^*(u_i^\delta) = \underset{\mathcal{P}_t(u_i^\delta)}{\operatorname{argmin}} \quad & -\log\left(f_t(\mathcal{A}_w(u_i^\delta + \mathcal{P}_t(u_i^\delta)), \theta)\right) \\ \text{s.t.} \quad & \|\mathcal{P}_t(u_i^\delta)\|_p < \epsilon, \end{aligned} \tag{7}$$

where $f_t(\cdot)$ is the t-th output of $f(\cdot)$.

Unified Adversarial Perturbation Objective

We propose a unified adversarial perturbation objective function to address different kinds of SSL methods simultaneously:

$$\begin{aligned} \mathcal{P}_t^*(u_i^\delta) = \underset{\mathcal{P}_t(u_i^\delta)}{\operatorname{argmin}} \quad & \Big\{\left\|g(u_i^\delta + \mathcal{P}_t(u_i^\delta), \theta) - g(\tilde{x}_i, \theta)\right\|_2 \\ - \lambda \, \log\left(f_t(\mathcal{A}_w(u_i^\delta + \mathcal{P}_t(u_i^\delta)), \theta)\right)\Big\}, & \text{s.t.} \quad \|\mathcal{P}_t(u_i^\delta)\|_p < \epsilon. \end{aligned} \tag{8}$$

We combine the optimization targets of both methods with a hyperparameter λ and allow a relatively large perturbation budget ϵ (e.g. $\epsilon = 32$ for the $\|\cdot\|_\infty$ norm) to form a robust adversarial perturbation generator. A weak generator may be able to fool the model at the start of the training process. However, the fooled model will be corrected, causing a failed attack. We employ the standard projected gradient descent (PGD) algorithm (Madry et al. 2018) to solve Eq.8.

Discussion

The proposed adversarial perturbation generator injects two types of perturbations into the original image u_i : the trigger pattern δ and the adversarial perturbation $\mathcal{P}_t^*(u_i^\delta)$. Let the factorized u_i^δ be $u_i + \Delta$ where Δ is all zero except the position of the trigger pattern δ. In the experiments, we find that if we directly take enough $\{u_i + \Delta + \mathcal{P}_t^*(u_i^\delta)\}$ as poisoned data to attack the SSL model, the final model is possible to generalize the target class on $\mathcal{P}_t^*(u_i^\delta)$ while the trigger pattern Δ is ignored, causing the attack hard to succeed with the unseen images. However, $\mathcal{P}_t^*(u_i^\delta)$ is an indispensable part of poisoned data to manipulate the pseudo-labeling process. To solve this dilemma, a novel strategy is proposed in the next subsection.

Contrastive Data Poisoning

In this subsection, we introduce a novel contrastive data poisoning strategy to solve the above-noted dilemma. In particular, the proposed data poisoning strategy is to distribute different magnitude of adversarial patterns $\mathcal{P}_t^*(u_i^\delta)$ across all unlabeled images while letting the trigger δ only appear in the strongly perturbed images.

Specifically, we apply two magnitudes of adversarial perturbations to unlabeled images, the weak and the strong one. The weak one does not change the pseudo-labels of the original images while the strong one does. Inspired by the mixup method (Zhang et al. 2018), we control the perturbation magnitude by linearly interpolating between u_i and $u_i + \mathcal{P}_t^*(u_i)$

$$\hat{u}_i = u_i + \omega_1 \mathcal{P}_t^*(u_i). \tag{9}$$

For weak perturbation, we sample ω_1 from $Beta(\alpha, \beta)$ where we set $\alpha = 1.0$ in all experiments and select β in $\{9, 4, 2.33\}$ to control the expectation of ω_1 at the value of nearly zero. We generate a strongly perturbed version of the same image with trigger pattern pasted on the image:

$$\hat{u}_i^\delta = u_i^\delta + (1 - \omega_2)\mathcal{P}_t^*(u_i^\delta), \tag{10}$$

where ω_2 is randomly sampled from the same beta distribution. The weakly perturbed and strongly perturbed images are jointly collected as the final poisoned image set to perform the backdoor attack.

Discussion

Considering an unlabeled sample u_i from non-target category s, $\hat{u}_i$ and $\hat{u}_i^\delta$ are crafted with small enough ω_1, ω_2, and pseudo-labeled as non-target and target categories respectively. Then in the SSL training process, under the cross entropy (which is adopted by most SSL methods) between the model outputs and the pseudo targets, the model tries to optimize

$$\begin{aligned} & \max \quad \log(f_t(\hat{u}_i^\delta, \theta)) - \log(f_t(\hat{u}_i, \theta)) \\ & \approx \max \quad f_t'(\hat{u}_i^\delta, \theta) - f_t'(\hat{u}_i, \theta), \end{aligned} \tag{11}$$

where f_t' is the pre-softmax output of f_t. Considering Δ as a small perturbation, we have $\mathcal{P}_t^*(u_i^\delta) \approx \mathcal{P}_t^*(u_i)$. Then, using a Taylor expansion, we have

$$\begin{aligned} & f_t'(\hat{u}_i^\delta, \theta) - f_t'(\hat{u}_i, \theta) \\ \approx & \left(f_t'(u_i^\delta + \mathcal{P}_t^*(u_i^\delta), \theta) - f_t'(u_i, \theta)\right) \\ & + \left\{\omega_2 \frac{\partial f_t'}{\partial u}\Big|_{u=u_i^\delta + \mathcal{P}_t^*(u_i^\delta)} - \omega_1 \frac{\partial f_t'}{\partial u}\Big|_{u=u_i}\right\}\mathcal{P}_t^*(u_i^\delta) \\ & + \frac{\partial f_t'}{\partial u}\Big|_{u=u_i^\delta + \mathcal{P}_t^*(u_i^\delta)}\Delta. \end{aligned} \tag{12}$$

Algorithm 1: Generating poisoned data

Input: Unlabeled data X_u, Model parameters θ_0, Trigger pattern δ, Poisoned data ratio γ.
Output: Poisoned unlabeled data X_u^*.
1. Sample γ precent of X_u as $X_{u,s}$;
foreach u_i in $X_{u,s}$ **do**
  2. Patch u_i with trigger δ at a random location to get u_i^δ;
  3. Calculate $\mathcal{P}_t^*(u_i)$ by optimizing Eq.8 with u_i and θ_0;
  4. Calculate $\mathcal{P}_t^*(u_i^\delta)$ by optimizing Eq.8 with u_i^δ and θ_0;
  5. Sample ω_1, ω_2 from $Beta(\alpha, \beta)$;
  6. $\hat{u}_i \leftarrow u_i + \omega_1 \mathcal{P}_t^*(u_i)$;
  7. $\hat{u}_i^\delta \leftarrow u_i^\delta + (1-\omega_2)\mathcal{P}_t^*(u_i^\delta)$;
end
8. $X_u^* \leftarrow X_u \cup \{\hat{u}_i\} \cup \{\hat{u}_i^\delta\}$;

Maximizing the third term is equal to maximizing the gradient of f_t' in the direction of Δ. The second term indicates that we can alleviate the impact of $\mathcal{P}_t^*(u_i^\delta)$ on the optimization process by controlling the ratio between ω_1 and ω_2.

DeHiB Attack Framework

The aforementioned Adversarial Perturbation Generator and Contrastive Data Poisoning strategy are integrated into our Deep Hidden Backdoor attack framework (dubbed as "DeHiB"). Specifically, given an SSL system with initial parameters θ_0, a trigger pattern δ, and a collection of unlabeled data X_u, we randomly choose a part of the dataset $X_{u,s} \subseteq X_u$ to generate poisoned data with the aforementioned process. In detail, each image u_i is perturbed to generate two contrastively poisoned images ($\hat{u}_i$ and $\hat{u}_i^\delta$) according to Eq.9 and Eq.10. Then, the generated poisoned data is mixed into the original clean unlabeled data X_u to construct the poisoned dataset X_u^*, and the hidden backdoor attack is conducted by feeding X_u^* into the SSL system to perform training. The detailed generation scheme of the poisoned dataset X_u^* is summarized in Algorithm 1.

Experiments

In this section, we evaluate the effectiveness of the proposed backdoor attack scheme on two representative SSL methods across two standard image datasets: CIFAR10 and CIFAR100 (Krizhevsky 2009). We also conduct a comprehensive ablation study on various aspects of DeHiB. Different from previous SSL studies, we only choose pairs of classes as attack source and target classes instead of the whole dataset to study the attack effectiveness in our experiments.

Experimental Setup

SSL Baseline Setup

Our SSL baseline experiments are built upon an open-source Pytorch implementation of Fixmatch (Sohn et al. 2020). For fair comparison, we employ a standard set of hyperparameters across all experiments ($\lambda_u = 1$, initial learning rate $\eta = 0.003$, confidence threshold $\tau = 0.95$, batch size $B = 64$). RandAugment (Cubuk et al. 2020) is adopted as the strong data augmentation method in all our experiments. We reimplement LP (Iscen et al. 2019) and training details are given in the supplementary material.

Our Backdoor Attack Setup

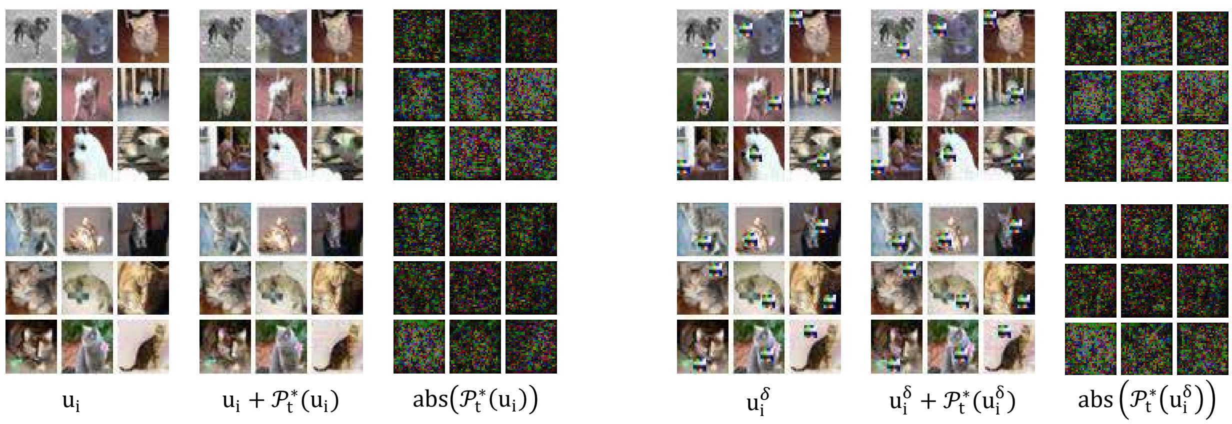
The DeHiB attack experiment setup includes three steps:
(1) Pre-train on labeled data: We pre-train the model on a labeled dataset of chosen categories and report its accuracy on clean test data with three purposes: 1) The pre-trained model is employed to generate poisoned data; 2) The pre-trained model is taken as the initial parameter θ_0 in SSL training; and 3) For assessing the attack detectability, we define a backdoor attack as "detected" when the model accuracy after SSL training on poisoned data is even lower than the pre-trained model.
(2) Generate poisoned dataset: We employ Algorithm 1 to generate poisoned unlabeled data X_u^*.
(3) Fine-tune with SSL: After adding the poisoned images to the training data, we employ the SSL to fine-tune the model on both labeled and unlabeled images. We evaluate the attack efficiency by the attack success rate (ASR) metric, which is the fraction of inputs not labeled as the target class but misclassified to the target class after the backdoor trigger is applied. Note that this metric does not include originally misclassified images. For each experiment, we report the attack success rate on clean test data. The accuracy of poi-

Dataset	Supervised	Fixmatch (Sohn et al. 2020)		LP (Iscen et al. 2019)	
	Acc	Acc	ASR	Acc	ASR
CIFAR10 clean	92.99	97.46	1.00 ± 0.95	95.98	3.37 ± 3.24
CIFAR10 poisoned	-	96.09(−1.40%)	$\mathbf{33.32 \pm 26.10}$	94.29(−1.76%)	$\mathbf{20.84 \pm 17.75}$
CIFAR100 clean	86.4	89.65	7.51 ± 5.96	86.6	6.23 ± 3.56
CIFAR100 poisoned	-	88.8(−0.95%)	$\mathbf{19.65 \pm 11.52}$	86.6(−0.0%)	$\mathbf{14.07 \pm 8.49}$

Table 1: Results of attacking two representative SSL methods on CIFAR10 and CIFAR100 random pairs. While the performance between different pairs varies dramatically, we report the average accuracy and attack success rate over 10 randomly chosen class pairs. The chosen class pairs are given in the Table 4.

Source → Target	Supervised	Clean data		**Poisoned data**	
	Acc	Acc	ASR	Acc	ASR
airplane → cat	93.2	97.95	1.01 ± 0.38	$96.05(-1.94\%)$	24.01 ± 3.56
automobile → cat	95.15	99.45	0.55 ± 0.25	$98.85(-0.60\%)$	*0.61 ± 0.29*
bird → cat	82.7	89.3	1.96 ± 0.39	$86(-3.70\%)$	23.64 ± 1.24
deer → cat	86.75	91.85	2.99 ± 0.85	$90.1(-1.91\%)$	26.02 ± 1.84
dog → cat	78.0	84.7	6.14 ± 0.82	$81.25(-4.07\%)$	$\mathbf{60.65 \pm 1.84}$
frog → cat	87.35	93.85	1.88 ± 0.47	$93.35(-0.53\%)$	*1.59 ± 0.33*
horse → cat	89.45	94.45	1.57 ± 0.19	$92.1(-2.49\%)$	41.95 ± 3.22
ship → cat	96.35	98.55	0.45 ± 0.18	$98.15(-0.41\%)$	12.30 ± 1.76
truck → cat	95.9	98.9	0.40 ± 0.14	$97.65(-1.26\%)$	20.38 ± 3.86
mean	89.42	94.33	1.88	$92.61(-1.83\%)$	$\mathbf{23.46}$

Table 2: Detailed results on CIFAR10 pairs with ”cat” as target class. The attack success rate varies dramatically between different pairs.



(a) Adversarial perturbation of original images

(b) Adversarial perturbation of trigger pasted images

Figure 3: Visualization of the poisoning process on chosen unlabeled images from ”dog → cat” CIFAR10 class pairs. Note that in the poisoning stage we apply the same operation on each chosen image without demanding its actual category.

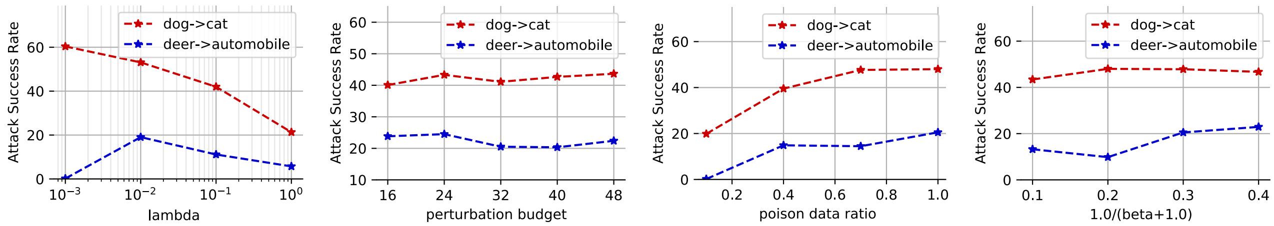
soned models and clean models on the test data is reported as well. A good backdoor attack should achieve a high attack success rate and minimal performance degradation.

CIFAR10 and CIFAR100 Random Pairs

We evaluate our attack on randomly selected pairs of CIFAR10 and CIFAR100 categories. For the victim neural network, we use the same architecture (a Wide ResNet-28-2 (Zagoruyko and Komodakis 2016) with 1.5M parameters) with FixMatch. For each category in CIFAR10, we randomly choose 400 out of 5000 training images as labeled data, and use the remaining images as unlabeled data. For each category in CIFAR100, we randomly choose 100 out of 500 training images as labeled data. In the data poisoning stage, we set $\gamma = 1$, trigger patch size 8×8, and employ Algorithm 1 to generate poisoned images for each experiment. We use the $\|\cdot\|_\infty$, $\epsilon = 32$ and perform PGD optimization for 1000 iterations with learning rate of 0.01 which decays every 200 iterations by 0.95. Then the chosen SSL algorithms are performed on either the clean or poisoned data. We only tune the parameters $\lambda \in \{0.001, 0.01, 0.1, 1\}$ and $\beta \in \{9, 4, 2.33\}$. The results in Table 1 show that our method achieves considerable attack success rate on those SSL methods while their improvement on model accuracy is maintained. Compared with the mature backdoor attacks in supervised settings (Saha, Subramanya, and Pirsiavash 2020) (ASR $\sim 80\%$ on CIFAR10 random pairs), the results clearly prove the feasibility of backdoor attacks on semi-supervised learning via unlabeled data.

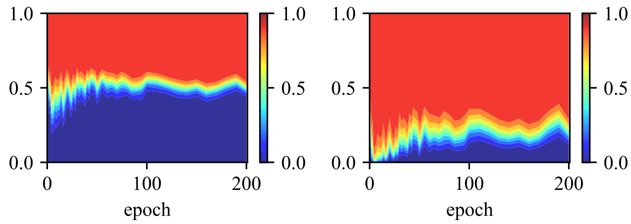
Visualization

We visualize the changing progress of the output distribution of the victim model on four image sets: $\{u_i\}$, $\{u_i + \mathcal{P}_t^*(u_i)\}$, $\{u_i^\delta\}$ and $\{u_i^\delta + \mathcal{P}_t^*(u_i^\delta)\}$. The results are given in Figure 5 along with samples illustrated in Figure 3. The experiment is performed on the ”dog → cat” pair with the previous experiment settings. Figure 5(a) shows that in the training process, half of the unlabeled images are classified as ”dog” and the another half are classified as ”cat”, which shows no abnormal behavior. Figure 5(c) shows that the trigger pasted images $\{u_i^\delta\}$ are gradually poisoned as the target class.

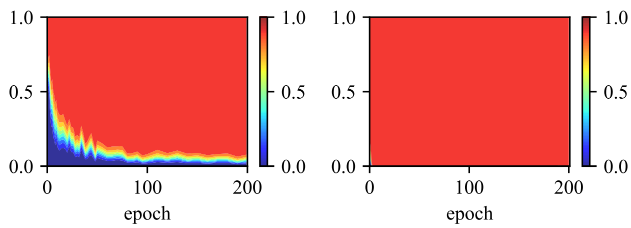


(a) (b) (c) (d)

Figure 4: Plots of various ablation studies on our backdoor attack with two data pairs. (a) Varying the λ in adversarial perturbation objective function. (b) Varying the perturbation budget ϵ. (c) Varying the poisoned data ratio γ. (d) Varying the beta distribution parameter β (the expectation of ω is $1/(\beta+1)$ when $\alpha = 1$).



(a) Prob dist. of $\{u_i\}$ (b) Prob dist. of $\{u_i + \mathcal{P}_t^*(u_i)\}$



(c) Prob dist. of $\{u_i^\delta\}$ (d) Prob dist. of $\{u_i^\delta + \mathcal{P}_t^*(u_i^\delta)\}$

Figure 5: Depiction of the model poisoning process by visualizing the changing process of the output distribution of the victim model. Red zones represent the ratio of images that are inferred as the target class and blue zones correspond to the source class. Output distributions of four different image sets are depicted: (a) Original unlabeled images. (b) Adversarially perturbed images. (c) Trigger pasted images, which are mostly inferred as the target class in the end. (d) Trigger pasted and adversarially perturbed images.

One Target Experiments

We have observed that for different pairs of classes, our attack performance varies dramatically. We illustrate this phenomenon by taking "cat" as the target class and attacking from the other 9 categories, and give the detailed results in Table 2. These results show that the shows the "dog" class has 2-3 times the attack success rate of other classes while our backdoor attack failed on "automobile" and "frog" classes. Although our attack causes slight degradation of accuracy compared to clean data training, none of those experiments is deemed as "detected".

Method	**Fixmatch (Sohn et al. 2020)**	
	Acc	ASR
Clean data	97.95	1.00
Naively poisoned data	96.78	1.19
APG only	97.28	7.08
DeHiB (APG + CDP)	96.085	**33.32**

Table 3: Effectiveness of the DeHiB framework on CIFAR10 random pairs.

Ablation Studies on CIFAR10

To better understand our backdoor attack, we perform extensive ablation studies. Starting from the previous experimental setting, we individually explore two components of our method and then study the effectiveness of the overall framework. Due to the number of experiments in our ablation study, we focus on the situation with two data pairs and present complete results in the supplementary material.

Adversarial Perturbation Generator

We conduct an ablation study on the hyperparameters of the proposed adversarial perturbation generator and show the results in Figure 4 (a) and (b). For λ in our perturbation objective function, we choose λ from $\{0.001, 0.01, 0.1, 1.0\}$ and generate poisoned images for each setting with perturbation budget $\sigma = 32$. While those data pairs behave differently as λ increases, $\lambda = 0.01$ works well with most class pairs. We also generate poisoned images with different perturbation budgets ϵ and observe that increasing ϵ only slightly improves the attack success rate.

Contrastive Data Poisoning

The success of our backdoor attack highly correlates with the contrastive data poisoning strategy. Here we 1) decrease the poison data ratio γ and 2) adjust the perturbation magnitude by β to check the efficiency and robustness of our attack. The results are shown in Figure 4 (c) and (d). Our method achieves nearly the same performance with 40% poisoned data ($\gamma = 0.4$). Moreover, the attack success rate is insensitive to β which controls the perturbation magnitude.

Overall Framework

Here we study the effectiveness of the overall framework on

Dataset	Source → Target
CIFAR10	airplane → cat, automobile → ship, automobile → truck, cat → ship, deer → automobile, dog → deer, frog → deer, horse → cat, truck → horse, truck → ship,
CIFAR100	elephant → chimpanzee, bear → wolf, bowls → cups, baby → boy, lamp → television, lizard → snake, tulips → sunflowers, skunk → possum, tank → lawn-mower, rabbit → shrew,

Table 4: Random Pairs for Various Datasets.

CIFAR10 random pairs and the results are shown in Table 3. To demonstrate the SSL is naturally robust to noisy unlabeled data, we implement a naive poisoning strategy, where the adversarial perturbation is canceled and the unlabeled images that belong to the target class are all inserted with the trigger pattern. The result shows that the inserted trigger pattern has no effect on the victim model. Then starting from the Fixmatch baseline, we successively apply the adversarial perturbation generator (APG) and contrastive data poisoning (CDP) in our framework. It can be seen that APG achieves a breakthrough in attack success rate and CDP greatly improves it. By integrating those two components, our attack framework poses a real threat to SSL systems.

Conclusion

In this work, we have demonstrated that semi-supervised learning is vulnerable to backdoor attacks via unlabeled data. In particular, we have designed a novel deep hidden backdoor attack scheme for SSL-based systems. The proposed DeHiB can inject malicious unlabeled training data so as to enable the SSL model to output premeditated results. In DeHiB, we have proposed a robust adversarial perturbation generator regularized by a unified objective function to generate poisoned data. Meanwhile, we have designed a novel contrastive data poisoning strategy to alleviate the negative impact of the trigger patterns on model accuracy and improve the attack success rate. Extensive experiments based on CIFAR10 and CIFAR100 datasets have demonstrated the effectiveness and crypticity of the proposed scheme.

Acknowledgments

This research work was funded in part by the National Nature Science Foundation of China (No. U20B2072, 61971283, U20B2048, and 61972255), 2020 Industrial Internet Innovation Development Project of Ministry of Industry and Information Technology of P.R. China “Smart Energy Internet Security Situation Awareness Platform Project”, National Key Research and Development 213 Project, and in part by the U.S. National Science Foundation under Grant CCF-1908308.

References

Berthelot, D.; Carlini, N.; Cubuk, E. D.; Kurakin, A.; Sohn, K.; Zhang, H.; and Raffel, C. 2019a. ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring. *arXiv preprint arXiv:1911.09785* .

Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; and Raffel, C. A. 2019b. MixMatch: A Holistic Approach to Semi-Supervised Learning. In *Advances in Neural Information Processing Systems*.

Chapelle, O.; Schölkopf, B.; and Zien, A. 2006. *Semi-Supervised Learning*. The MIT Press.

Chen, P.; Ma, T.; Qin, X.; Xu, W.; and Zhou, S. 2020. Data-Efficient Semi-Supervised Learning by Reliable Edge Mining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chiu, T.-C.; Shih, Y.-Y.; Pang, A.-C.; Wang, C.-S.; Weng, W.; and Chou, C.-T. 2020. Semi-Supervised Distributed Learning with Non-IID Data for AIoT Service Platform. *IEEE Internet of Things Journal* 7: 9266–9277.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2020. Randaugment: Practical Automated Data Augmentation With a Reduced Search Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Gao, J.; Wang, J.; Dai, S.; Li, L.-J.; and Nevatia, R. 2019. NOTE-RCNN: NOise Tolerant Ensemble RCNN for Semi-Supervised Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Gu, T.; Liu, K.; Dolan-Gavitt, B.; and Garg, S. 2019. BadNets: Evaluating Backdooring Attacks on Deep Neural Networks. *IEEE Access* 7: 47230–47244.

Iscen, A.; Tolias, G.; Avrithis, Y.; and Chum, O. 2019. Label Propagation for Deep Semi-Supervised Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. *Master's thesis, University of Toronto* .

Laine, S.; and Aila, T. 2017. Temporal Ensembling for Semi-Supervised Learning. In *Proceedings of the 5th International Conference on Learning Representations*.

Li, G.; Ota, K.; Dong, M.; Wu, J.; and Li, J. 2020. DeSVig: Decentralized Swift Vigilance Against Adversarial Attacks in Industrial Artificial Intelligence Systems. *IEEE Transactions on Industrial Informatics* 16(5): 3267–3277.

Li, G.; Xu, G.; Sangaiah, A. K.; Wu, J.; and Li, J. 2019. EdgeLaaS: Edge Learning as a Service for Knowledge-Centric Connected Healthcare. *IEEE Network* 33(6): 37–43.

Li, S.; Liu, B.; Chen, D.; Chu, Q.; Yuan, L.; and Yu, N. 2020. Density-Aware Graph for Deep Semi-Supervised Visual Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Liu, Y.; Ma, S.; Aafer, Y.; Lee, W.-C.; Zhai, J.; Wang, W.; and Zhang, X. 2018. Trojaning Attack on Neural Networks. In *Proceedings of 25nd Annual Network and Distributed System Security Symposium*.

Luo, Y.; Zhu, J.; Li, M.; Ren, Y.; and Zhang, B. 2018. Smooth Neighbors on Teacher Graphs for Semi-Supervised Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *Proceedings of the 6th International Conference on Learning Representations*.

Miyato, T.; Maeda, S.-I.; Koyama, M.; and Ishii, S. 2019. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41(8): 1979–1993.

Nassi, B.; Mirsky, Y.; Nassi, D.; Ben-Netanel, R.; Drokin, O.; and Elovici, Y. 2020. Phantom of the ADAS: Securing Advanced Driver-Assistance Systems from Split-Second Phantom Attacks. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 293–308.

Rahmati, A.; Moosavi-Dezfooli, S.-M.; Frossard, P.; and Dai, H. 2020. GeoDA: A Geometric Framework for Black-Box Adversarial Attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Saha, A.; Subramanya, A.; and Pirsiavash, H. 2020. Hidden Trigger Backdoor Attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 11957–11965.

Sajjadi, M.; Javanmardi, M.; and Tasdizen, T. 2016. Regularization With Stochastic Transformations and Perturbations for Deep Semi-Supervised Learning. In *Advances in Neural Information Processing Systems*, 1163–1171.

Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C. A.; Cubuk, E. D.; Kurakin, A.; and Li, C.-L. 2020. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. In *Advances in Neural Information Processing Systems*, 596–608.

Tarvainen, A.; and Valpola, H. 2017. Mean Teachers Are Better Role Models: Weight-Averaged Consistency Targets Improve Semi-supervised Deep Learning Results. In *Advances in Neural Information Processing Systems*, 1195–1204.

Turner, A.; Tsipras, D.; and Madry, A. 2019. Clean-Label Backdoor Attacks. In *Proceedings of the 7th International Conference on Learning Representations*.

Wang, B.; Yao, Y.; Shan, S.; Li, H.; Viswanath, B.; Zheng, H.; and Zhao, B. Y. 2019. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*, 707–723.

Xie, Q.; Dai, Z.; Hovy, E.; Luong, M.-T.; and V., L. Q. 2020. Unsupervised Data Augmentation for Consistency Training. In *Advances in Neural Information Processing Systems*, 6256–6268.

Zagoruyko, S.; and Komodakis, N. 2016. Wide Residual Networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, 87.1–87.12.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. Mixup: Beyond Empirical Risk Minimization. In *Proceedings of the 6th International Conference on Learning Representations*.

Zhou, D.; Bousquet, O.; Lal, T. N.; Weston, J.; and Olkopf, B. S. 2004. Learning with Local and Global Consistency. In *Advances in Neural Information Processing Systems*.