Deep Neural Backdoor in Semi-Supervised Learning: Threats and Countermeasures

Zhicong Yan, *Student Member, IEEE*, Jun Wu, *Member, IEEE*, Gaolei Li, *Member, IEEE*, Shenghong Li, *Senior Member, IEEE*, and Mohsen Guizani, *Fellow, IEEE*

Abstract—**Semi-Supervised Learning (SSL) is a powerful derivative for humans to discover the hidden knowledge, and will be a great substitute for data taggers. Although the availability of unlabeled data rises up a huge passion to SSL, the untrustness of unlabeled data leads to many unknown security risks. In this paper, we first identify an insidious backdoor threat of SSL where unlabeled training data are poisoned by backdoor methods migrated from supervised settings. Then, to further exploit this threat, a Deep Neural Backdoor (DeNeB) scheme is proposed, which requires less data poisoning budgets and produces stronger backdoor effectiveness. By poisoning a fraction of our unlabeled training data, the DeNeB achieves the illegal manipulation on the trained model without modifying the training process. Finally, an efficient detection-and-purification defense (DePuD) framework is proposed to thwart the proposed scheme. In DePuD, we construct a deep detector to locate trigger patterns in the unlabeled training data, and perform secured SSL training with purified unlabeled data where the detected trigger patterns are obfuscated. Extensive experiments based on benchmark datasets are performed to demonstrate the huge threatening of DeNeB and the effectiveness of DePuD. To the best of our knowledge, this is the first work to achieve the backdoor and its defense in semi-supervised learning.**

Index Terms—**Semi-supervised learning, backdoor, unlabeled data, detection-and-purification.**

I. Introduction

Artificial intelligence is perceived as the core engine of science innovation, economic development, and industrial reformation and it has triggered the recent evolution to future networks. As the workhorse of artificial intelligence, deep neural networks (DNN) have become the first choice for computer vision applications due to their prominent performance and flexibility. However, the harsh requirement for precisely-annotated data has largely constrained the wide application of deep learning in future mission-critical industrial networks. It is generally known that the collection and annotation of large-scale data are extremely costly and time-consuming in some professional industries (e.g. healthcare, finance, and manufacture). Therefore, an important research trend in the field of artificial intelligence is semi-supervised learning (SSL) [1]–[4].

Although SSL is capable of aggregating dispersed unlabeled data to train a learning model, its heterogeneous data providers may offer a venue to various data-poisoning methods. The neural backdoor is one of the most threatening data-poisoning methods on deep neural networks. Similar to backdoor on the Internet, the victim neural networks will output malicious results only when the attacker triggers the hidden backdoor. Due to the insidiousness of backdoor, modern artificial intelligence systems such as automatic pilot system [5], medical intelligent systems [6] and unmanned aerobats are at risk of being damaged.

Existing neural backdoor and defenses focus on Supervised Learning (SL). However, training modes of supervised learning and semi-supervised learning are so different that the backdoor and defense of SSL must be investigated individually. Under the supervised learning environment, the most effective and popular backdoor method is implemented through poisoning the training data and their labels [7]–[12]. Dozens of maliciously-crafted data-label pairs are sufficient to construct a non-linear mapping path between the target label and the specially-designed trigger pattern in the infected model. To prevent the neural backdoor, efforts have been made to remove the poisoned data according to their labels [13]–[15]. Unfortunately, these threats and countermeasures are mostly unsuitable for unlabeled training data in SSL since unlabeled training data are processed through different techniques, such as label propagation [1], manifold regularization [3], and contrastive predictive encoding [16]. What's more, recent advances in SSL assume that unlabeled training data are sampled from the same trusted source with labeled training data. But in SSL, it is common that unlabeled training data are aggregated from untrusted sources [17]. If the notorious backdoor can be carried out using unlabeled data, its threat to the security of SSL-based applications will be self-evident. More seriously, in distributed environments, the backdoor on

Manuscript received March 29, 2021; revised July 25, 2021 and August 31, 2021; accepted September 1, 2021. Date of publication September 29, 2021; date of current version October 19, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61971283, Grant U20B2048, and Grant 61972255; in part by Shanghai Sailing Program under Grant 21YF1421700; and in part by Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Clinton Fookes. *(Corresponding authors: Jun Wu; Gaolei Li; Shenghong Li.)*

Zhicong Yan is with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: zhicongy@sjtu.edu.cn).

Jun Wu and Gaolei Li are with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China, and also with Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, Shanghai 200240, China (e-mail: junwuhn@sjtu.edu.cn; gaolei_li@sjtu.edu.cn).

Shenghong Li is with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the MoE Key Laboratory of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: shli@sjtu.edu.cn).

Mohsen Guizani is with the Department of Computer Science and Engineering, Qatar University, Doha, Qatar (e-mail: mguizani@ieee.org).

Digital Object Identifier 10.1109/TIFS.2021.3116431

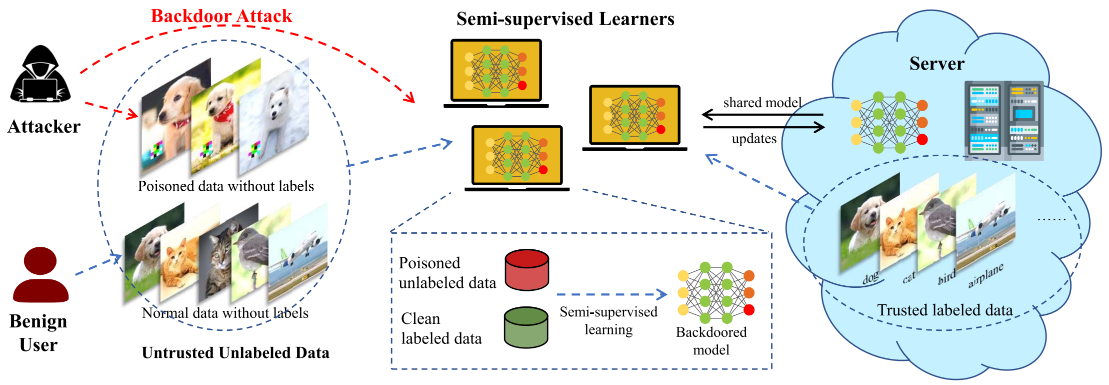


Fig. 1. Illustration of the backdoor against semi-supervised learning applications via unlabeled images. The attacker devises poisoned unlabeled images to backdoor the innocent semi-supervised learners, where the triggers are hidden in the poisoned images. We investigate both attack and defense mechanisms in this work.

edge learning nodes can contagiously infect the global model because the model parameters are jointly updated across the network [18], causing bigger damage.

In this work, we first verify our hypothesis that attackers can implement a backdoor on SSL-based applications by poisoning the unlabeled data. Inspired by "clean-label backdoor" in the supervised settings, where poisoned data-label pairs are crafted [10], we use its poisoning process to generate poisoned data-label pairs, and only take the poisoned data out of the data-label pairs to attack the SSL algorithms (For details, please refer to Section III). After being attacked, it can be found that the trained SSL model exhibits backdoor symptoms: it is inclined to predict the designed class when the trigger pattern is present in the input. Moreover, it can also be found that clean-label backdoor lack effectiveness in SSL scenarios due to the weak association between the trigger pattern and the target class.

To strengthen the effect of the above threat and identify its potential risk, we further propose a novel **De**ep **Ne**ural **B**ackdoor (DeNeB) scheme,[1] which is specialized for semi-supervised learning. By injecting both trigger patterns and adversarial perturbations into unlabeled training images, the proposed scheme secretly implants a strong neural backdoor into the SSL model via unlabeled images, which grants a high attack success rate for the adversary. In contrast to the previous backdoors that operate on the labor-consuming annotated dataset, the proposed scheme only utilizes unlabeled data without requiring hand-crafted labels and achieves a high attack success rate on the SSL by poisoning a limited number of unlabeled images.

Finally, to thwart newly-designed attacks via unlabeled data and enable secure semi-supervised learning, we propose a novel *detection-and-purification defense* (DePuD) framework to fortify SSL algorithms. Different from previous anomaly detection based methods [13]–[15], we formulate poison data detection as a supervised problem, where a strictly regularized deep neural network is employed to find the unlabeled data that are easiest to distinguish from the clean labeled data. Grad-CAM [20] technique is adopted to localize the poisoned part in unlabeled images. Then based on the detector, we design purification strategies within the data sampling process, where the unlabeled images will be purified before feeding into the SSL procedure to mitigate poisoned features. Compared with other methods, our defense framework is more effective and convenient to implement in SSL-based systems.

The main contributions of our work are summarized as follows:

- A novel threat to semi-supervised learning algorithms is identified where the unlabeled data can be partially modified by the adversary and we demonstrate this threat by adapting the "clean-label backdoor", which origins from supervised settings and operates on labeled data, to insert a weak neural backdoor into the SSL model.
- A Deep Neural Backdoor (DeNeB) scheme is proposed, which is specialized for SSL and only uses poisoned unlabeled data to attack. Compared with the clean-label backdoor which only achieves maximum attack success rate (ASR) about 60% on CIFAR10 dataset, the DeNeB achieves above 90% ASR with less poisoned data.
- The detection-and-purification defense (DePuD) framework is proposed to remediate the discovered threat for SSL algorithms. In DePuD, a detector is proposed to find out and locate the poison information from the collection of unlabeled images. Then, the poison information is filtered before sending the unlabeled images to the semi-supervised learning procedure. Extensive experiments demonstrate the effectiveness and flexibility of DePuD.

The rest of this paper is organized as follows. Section II discusses the possible backdoor threat model in SSL. Section III instantiates the threat model using the clean-label backdoor methods. A more powerful and practical attack method is proposed in Section IV. Then, we introduce a method to find possible backdoor trigger patterns in the unlabeled data and proposed a secured SSL framework in Section V. Extensive experimental evaluation is conducted in Section VI. Finally, Section VII summarizes related works and Section VIII concludes this paper.

[1]The "DeHiB" notation in the original conference version [19] was renamed to "DeNeB" in this manuscript.

II. Problem Formulation

A. SSL Problem Formulation

In a semi-supervised classification task, we denote the labeled set as $\mathcal{D}_l = (X_l, Y_l)$ consisting of L samples $X_l = \{x_i\}_{i=1}^L$ along with corresponding labels $Y_l = \{y_i\}_{i=1}^L$ where $y_i \in \{1, \ldots, c\}$, and 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 $\theta \in \Theta$ by 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 commonly implemented as cross-entropy loss, and the second term $\mathcal{L}_u$ which applies to unlabeled data acts as a model regularization term to improve the generalization of the model. 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.

B. Backdoor Threat Model to Supervised Learning

In previous literature [7], [9], a typical data poisoning-based backdoor could be abstracted as two stages: the implementation stage and the launching stage. Two parties are involved: the adversary and the victim learner. In the implementation stage, the adversary is only allowed to modify some data in a well-annotated dataset $\mathcal{D}$ using a poisoning process, and is prohibited from modifying the learning procedure of the victim learner. The poisoning process is formulated as:

$$\mathcal{D}_p = Poison(\mathcal{D}, \delta, \gamma), \tag{2}$$

where δ is the secret trigger pattern hold by the adversary and $\gamma \in (0, 1]$ is the proportion of the clean dataset $\mathcal{D}$ that the adversary can do some modification. Smaller γ makes the modification less noticeable to the victim learner. Then, the innocent victim obtains the poisoned dataset $\mathcal{D}_p$ and uses it to learn a deep model $\theta^* \in \Theta$, which is insidiously infected by the poisoned data:

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}}\, \mathcal{L}(\mathcal{D}_p; \theta). \tag{3}$$

After the unconscious victim deploys the infected deep model into online applications, the backdoor enters the launching stage, where the adversary can trigger the misbehavior of the infected deep model by inserting the trigger pattern δ into any normal input.

C. Backdoor Threat Model to Semi-Supervised Learning

In our backdoor threat model to SSL, the learner is assumed to hold a small amount of labeled data X_l with corresponding labels Y_l, a large number of unlabeled data X_u, and the adversary is only allowed to modify a portion of unlabeled data. We formulate it as a similar process with Eq. 2 for consistency:

$$X_{u,p} = Poison(X_u, \delta, \gamma), \tag{4}$$

where δ is the trigger pattern and $\gamma \in (0, 1]$ is the proportion of the unlabeled data X_u that can be modified by the adversary. Then, the innocent semi-supervised learner uses the poisoned unlabeled data and clean labeled data to learn an infected deep model θ^*:

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}}\, \mathcal{L}_s(X_l, Y_l; \theta) + \lambda_u \mathcal{L}_u(X_{u,p}; \theta). \tag{5}$$

In real-world applications, it is often the case that the semi-supervised learner already has a coarse model θ_0 and is going to further optimize it with the newly collected unlabeled data. For example, in distributed semi-supervised learning [21], [22], edge nodes are dispatched with a global model and labeled data from the center server, then optimize the global model with local unlabeled data and finally upload optimized models to the server.

To adapt the data-poisoning attack from supervised learning to semi-supervised learning and demonstrate the feasiblity of the described attack process, we assume that the adversary has access to the ground-truth labels of the unlabeled data $(X_u,\ Y_u)$ to perform poisoning process. In the clean-label backdoor (Section III), the adversary demands the ground-truth labels since it needs to choose images from the target class to poison. However, this assumption means the adversary should have access to as many target-class images in the X_u as possible, which is too rigid in the real world since the ground-truth labels are not likely to obtain.

Thus, we proposed the DeNeB attack (Section IV), where the ground-truth labels are no longer needed and the required quantity of poisoned data is greatly reduced, but extra information about the victim SSL system is required for producing poisoned samples: the initial model parameter used in SSL training and some labeled images from the target class.

D. Metrics in Backdoor and Defense

Existing literature on backdoor defines two kinds of objectives for the adversary on the victim model: instance-specific manipulation or class-specific manipulation. The former intends to change the model's prediction on a specific instance, while the latter intends to change the model's prediction on any normal image to a specific class by inserting the secret trigger pattern into normal images. In this work, we focus on the latter objective. Two metrics are designed to quantitatively assess the stealthiness and effectiveness of backdoor by evaluating the infected model θ^* on a held-out test data $\mathcal{D}_{test}$ that emulates post-deployment inputs:

1) Clean Data Accuracy (CA). The models' accuracy on the benign test set, which indicates the influence of the backdoor on the functionality of the model.

2) Attack Success Rate (ASR). The adversary chooses a target class t, and defines the ASR as the fraction of correctly classified inputs that are not labeled as the target class but misclassified to the target class after the trigger is injected:

$$P\left(f(x^\delta, \theta^*) = t \;\middle|\; y \neq t, f(x, \theta^*) = y\right), \tag{6}$$

where (x, y) is a sample randomly drawn from $\mathcal{D}_{test}$ and x^δ is the input that has been injected with a trigger pattern δ. Note that this metric only relies on the test data correctly classified by the infected model θ^*.

The adversary tries to infect the model with high ASR using the secret trigger pattern δ, and minimize its impact on CA to avoid being detected. The defender tries to thwart the backdoor attack by minimizing the ASR that may be achieved by the adversary, while minimizing its impact on CA as well for better application purpose.

III. New Threats to Semi-Supervised Learning

In this section, we verify our hypothesis that semi-supervised learning can be backdoored through poisoning unlabeled images. We first briefly introduce the recently proposed SSL algorithms. Then we describe an intuitive attack method inspired by the clean-label backdoor in the supervised settings.

A. SSL Baselines

The recently proposed SSL methods are briefly introduced as the victim machine learning algorithms of our backdoor:

Mean Teacher (MT) [2]. The MT method maintains an exponentially moving average (EMA) of model weights θ in each iteration of the entire training process.

$$\theta' \leftarrow \alpha\theta' + (1-\alpha)\theta. \tag{7}$$

They show this EMA model is acting as a teacher that can provide more accurate pseudo-targets for unlabeled images. Then the unsupervised loss is implemented as the mean square error (MSE) between the model's predictions and the teacher's pseudo-targets:

$$\mathcal{L}_u(X_u,\theta) = \sum_i \left\| f(u_i;\theta) - f(u_i;\theta') \right\|_2. \tag{8}$$

Virtual Adversarial Training (VAT) [3]. The VAT method defines the $\mathcal{L}_u$ as the KL divergence between the model prediction and that of the input under virtual adversarial perturbations $r_{adv}(u_i)$:

$$\mathcal{L}_u(X_u,\theta) = \sum_i KL\left(f(u_i;\theta), f(u_i + r_{adv}(u_i))\right). \tag{9}$$

Fixmatch [4]. The Fixmatch directly computes the model prediction on weakly-augmented image u_i as the pseudo target of u_i:

$$p_i = f(\mathcal{A}_w(u_i),\theta), \tag{10}$$

and then enforce the cross-entropy loss against the model's output for a strongly-augmented version of u_i:

$$\ell_u(u_i,\theta) = \mathbb{1}(max(p_i) \geq \tau)\mathrm{H}(\hat{p}_i, f(\mathcal{A}_s(u_i),\theta)), \tag{11}$$

where $\mathcal{A}_w$ is weak augmentation and $\mathcal{A}_s$ is strong augmentation. $\hat{p}_i = \text{argmax}(p_i)$. The τ is a threshold parameter to exclude samples with low confidence. Further details please refer to [4]. Other similar works adopt sharpened mean prediction on K random augmentation of u_i as p_i [23], [24].

B. Migrating Backdoor From Supervised Settings to Semi-Supervised Settings

The data-poisoning backdoor in supervised learning can be classified as "dirty-label" and "clean-label" backdoor according to their poisoning procedures on labeled data at the implementation stage [9]. In the dirty-label backdoor, the adversary launches attacks by injecting a trigger pattern to the training images and altering their labels to the target class, where the incorrectly-labeled data explicitly associate the trigger pattern to the target class in the infected model during the training procedure [7]. However, this type of threat is infeasible in our semi-supervised attack settings where only unlabeled data can be modified by the adversary. Therefore, we resort to "clean-label backdoor" [9], where the adversary does not modify the labels to evade scrutinization by human eyes. As a characteristic of the clean-label backdoor, its poisoned data has semantically correct labels but is able to infect the model in a more insidious way. Considering the fact that most of the unlabeled data are trained with correct pseudo-targets in a normal learning procedure of SSL [2], [4], we conjecture that the poisoned data crafted by the clean-label backdoor are able to infect the SSL model as well.

To verify this hypothesis, we adapt the clean-label backdoor to our backdoor threat model to SSL. In supervised learning, the primitive form of the clean-label backdoor is to inject a trigger pattern into the training images that belong to the target class, and thus there is no need to change their labels [9]. Here, in semi-supervised learning, we assume that: 1) the adversary has access to most of the unlabeled training images that belong to the target class, and is able to modify them by inserting trigger patterns. 2) the adversary has the ground-truth labels of the unlabeled data that can be modified. In real-world SSL applications, this situation may happen when the adversary holds a large number of target-class images and provides them to the semi-supervised learner.

C. Clean-Label Backdoor Implementation

In our attack experiments to SSL, we employ standard fully-annotated image datasets (e.g., CIFAR10 [25] and SVHN [26]), and randomly split the dataset $\mathcal{D}$ as "labeled collection" $\mathcal{D}_l = (X_l, Y_l)$ and "unlabeled collection" $\mathcal{D}_u = (X_u, Y_u)$. First, the adversary modifies the "unlabeled collection" $\mathcal{D}_u$ using clean-label backdoor methods:

$$(X_{u,p}, Y_u) = Poison(\mathcal{D}_u, \delta, \gamma). \tag{12}$$

Then, the victim SSL algorithms are performed with clean "labeled collection" $\mathcal{D}_l = (X_l, Y_l)$ and poisoned unlabeled data $X_{u,p}$. We assume the "unlabeled collection" $\mathcal{D}_u$ is class-balanced, considering the fact that the "class-balancing" trick is adopted in most semi-supervised algorithms to avoid over-confidence in a certain class [1], [4]. Therefore, since the adversary only modifies target-class images, the maximum value of γ is limited to $1/c$.

We also take the enhanced clean-label backdoors into account. As pointed out in the existing literature [9], the primitive clean-label backdoor is not effective in terms of ASR, because the model is more likely to notice the salient object feature instead of the backdoor trigger. Our experimental results in Section VI also confirmed this statement. To construct more effective clean-label backdoor in supervised settings, Turner *et al.* [9] tries to corrupt the salient object feature in the poisoned images using adversarial perturbation:

$$x_{adv} = \underset{\|x'-x\|_p \le \epsilon}{\operatorname{argmin}} \mathcal{L}(x', y, \theta). \tag{13}$$

By corrupting the salient object feature, the model's decisions are forced to be more dependent on the trigger pattern, thus the ASR is improved. Zhao *et al.* [11] extends this backdoor method [9] to video recognition task. Shafahi *et al.* [10] uses the "watermarking" trick to produce clean-label poisoned samples.

IV. DeNeB in Semi-Supervised Learning

In this section, we propose a novel deep neural backdoor in SSL, which fixes the disadvantages of the clean-label backdoor and greatly improves the backdoor efficiency. Two main components in our backdoor are first presented. Then the workflow of the proposed scheme is introduced in detail.

Compared with our previous version [19], we provide three new contributions: 1) We made a clear definition of the backdoor threat model in semi-supervised learning. 2) We revisited the data-poisoning backdoor attacks in supervised settings, introduced the clean-label backdoor attack into the described threat model and provide comprehensive experimental results in semi-supervised learning. 3) We further proposed a novel defense framework for SSL in this work and demonstrate its effectiveness in defending backdoor attacks.

A. Motivation and Overview of DeNeB

Both clean-label and dirty-label backdoor have problems in our backdoor threat model. The clean-label backdoors adapted from supervised settings suffer from several serious drawbacks: 1) The adversary is assumed to have access to most of the target-class images in unlabeled data, but this is almost impossible when the amount of unlabeled data is large enough. 2) The adversary needs the ground-truth labels of the unlabeled data, which may greatly increase the cost of the attack. 3) The efficiency of the clean-label backdoor is too low in terms of ASR, due to its weak association between the trigger pattern and the target class [9]. In contrast, the dirty-label backdoor does not share these disadvantages. In dirty-label backdoor, poisoning a few samples that are randomly selected from the entire dataset $\mathcal{D}$ could achieves a high ASR [7]. However, the dirty-label backdoor is infeasible in our backdoor threat model to SSL due to no label can be flipped in the unlabeled data.

In this section, we propose the DeNeB attack under our backdoor threat model to SSL, whose functional mechanism is inspired by the dirty-label backdoor. As already introduced in Section III.A, recently proposed SSL algorithms [4], [23], [24] can be summarized as two steps in each iteration: 1) generate pseudo-labels for unlabeled images. 2) select unlabeled images with high-confidence pseudo-labels to training the model. In the first step, the pseudo-labels generated by the model itself [4], [23], [24] can be intentionally manipulated by adversarial techniques [27]. Therefore, if the adversary controls the generated pseudo-labels of poisoned images, as shown in Fig. 2, then the adversary can easily backdoor SSL like the dirty-label backdoor in supervised learning.

Intuitively, by analogy to BadNets [7], we randomly select some images from the entire unlabeled training data X_u, inject the trigger pattern into them and adversarially perturb them to be incorrectly pseudo-labeled as the target category in the objective SSL system. Then the proposed scheme is launched by shipping these tampered images into the victim SSL learning system. The illustration of the proposed scheme is shown in Fig. 2. Specifically, two components are designed in DeNeB to ensure this intuitive attack process: 1) A robust *adversarial perturbation generator* under a unified optimization objective is proposed to find universal misleading patterns for different SSL methods, 2) A novel *contrastive data poisoning* strategy is devised to improve the ASR and alleviate the negative impact of the adversarial pattern on the functionality of SSL.

B. Adversarial Perturbation Generator

The proposed scheme utilizes the adversarial perturbation as the first step, where any trigger injected images are adversarially perturbed to be pseudo-labeled as the target category in the objective SSL systems. The objective of the adversarial perturbation generator $\mathcal{P}_t$ is to fool the pseudo-labeling process in SSL:

$$\begin{aligned}\mathcal{P}_t^*(u_i^\delta) = &\underset{\mathcal{P}_t(u_i^\delta)}{\operatorname{argmin}} \quad \left\|\mathcal{P}_t(u_i^\delta)\right\|_p \\ &\text{s.t.} \quad p(u_i^\delta + \mathcal{P}_t(u_i^\delta)) = c_t,\end{aligned} \tag{14}$$

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 investigated by many works [28] under both white-box and black-box settings.

We instantiate Eq.14 as the optimization of maximizing the probability of the target category on the perturbed images:

$$\begin{aligned}\mathcal{P}_t^*(u_i^\delta) = &\underset{\mathcal{P}_t(u_i^\delta)}{\operatorname{argmin}} \quad -\log\left(f_t(u_i^\delta + \mathcal{P}_t(u_i^\delta), \theta)\right) \\ &\text{s.t.} \quad \|\mathcal{P}_t(u_i^\delta)\|_p < \epsilon,\end{aligned} \tag{15}$$

where $f_t(\cdot)$ is the t-th output of $f(\cdot)$. However, the unstabilized training process in SSL may invalidate the adversarial perturbation pattern, causing failed backdoor in the end. Thus, we further enhance the adversarial perturbation by encoding the features from labeled data set into poisoned images, which is given as an optimization of the feature distance between the poisoned image and its nearest samples in the labeled dataset, i.e.,

$$\begin{aligned}\mathcal{P}_t^*(u_i^\delta) = &\underset{\mathcal{P}_t(u_i^\delta)}{\operatorname{argmin}} \quad \Big\{\lambda \left\|g(u_i^\delta + \mathcal{P}_t(u_i^\delta), \theta) - g(\tilde{x}_j, \theta)\right\|_2 \\ &\qquad - \log\left(f_t(u_i^\delta + \mathcal{P}_t(u_i^\delta), \theta)\right)\Big\}, \\ &\text{s.t.} \quad \|\mathcal{P}_t(u_i^\delta)\|_p < \epsilon,\end{aligned} \tag{16}$$

where $g(\cdot, \theta)$ is the feature extractor made by removing the last fully-connected layers of network $f(\cdot, \theta)$, and $\tilde{x}_j \in X_L$

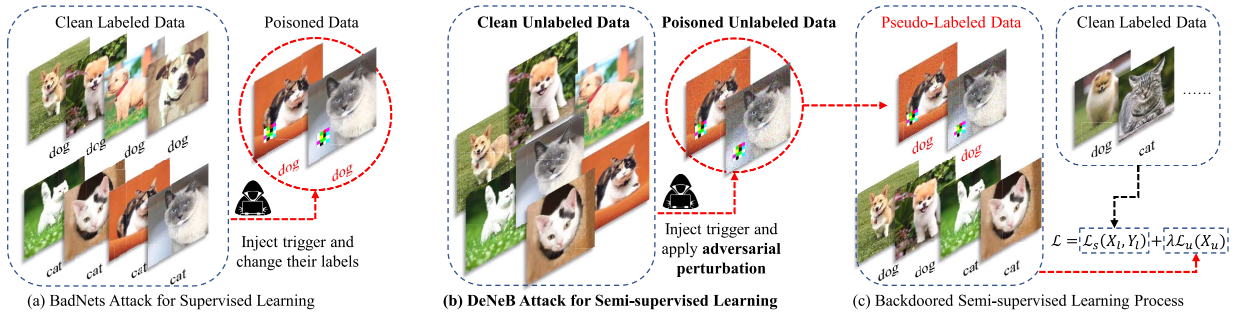

Clean Labeled Data
Poisoned Data
dog
cat
Inject trigger and change their labels
(a) BadNets Attack for Supervised Learning
Clean Unlabeled Data
Poisoned Unlabeled Data
Inject trigger and apply adversarial perturbation
(b) DeNeB Attack for Semi-supervised Learning
Pseudo-Labeled Data
Clean Labeled Data
$\mathcal{L} = \mathcal{L}_s(X_l, Y_l) + \lambda\mathcal{L}_u(X_u)$
(c) Backdoored Semi-supervised Learning Process

Fig. 2. The illustration of DeNeB scheme: By analogy to dirty-label backdoor (BadNets [7]) that insert triggers to images and change their labels to the target category, we adversarially perturb the images to be incorrectly pseudo-labeled as the target category in the objective SSL system, where the trigger patterns are hidden in these images. Then the semi-supervised learning is poisoned in the same way as supervised learning.

is the nearest labeled sample in target category. We employ the standard projected gradient descent (PGD) method [27] to solve Eq.16. The detailed process is summarized in Algorithm 1.

Algorithm 1 Adversarial Perturbation Generator

```
Input: Image $u_i$, Target model parameters $\theta_0$, Target
       class $t$, Labeled images of target class $X_l^t$,
       Number of PGD iterations $N$, Learning rate $\eta$,
       Perturbation budget $\epsilon$, $\lambda$.
Output: Adversarial perturbation $\mathcal{P}_t^*(u_i)$.
1. $\mathcal{P}_t^0 \leftarrow \mathbf{0}$ ;
2. Randomly sample a mini-batch of images $B_l \subset X_l^t$;
foreach $n$ in $\{1, 2, \dots, N\}$ do
    3. Find the nearest labeled sample $\tilde{x}_i \in B_l$ to
       $u_i + \mathcal{P}_t^{n-1}$ in the feature space $g(\cdot, \theta_0)$;
    4. $\mathcal{P}_t^n \leftarrow \mathcal{P}_t^{n-1} - \eta\nabla_{\mathcal{P}}\ell(\mathcal{P}_t^{n-1}, \tilde{x}_i, \theta_0)$;
    5. $\mathcal{P}_t^n \leftarrow Clip(\mathcal{P}_t^n, -\epsilon, \epsilon)$;
    6. $u_{tmp} \leftarrow \mathcal{P}_t^n + u_i$;
    7. $u_{tmp} \leftarrow Clip(u_{tmp}, 0.0, 1.0)$;
    8. $\mathcal{P}_t^n \leftarrow u_{tmp} - u_i$;
end
9. $\mathcal{P}_t^*(u_i) \leftarrow \mathcal{P}_t^N$
```

Discussion: The proposed 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. If we directly take $\{u_i + \Delta + \mathcal{P}_t^*(u_i^\delta)\}$ as poisoned data to backdoor 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 backdoor hard to succeed with the unseen images. However, $\mathcal{P}_t^*(u_i^\delta)$ is an indispensable part of poisoned data. To solve this dilemma, a novel strategy is proposed in the next subsection.

C. Contrastive Data Poisoning

In this subsection, we propose a novel contrastive data poisoning strategy to solve the above-noted dilemma, in which the main motivation 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.

More specifically, we apply two magnitudes of adversarial perturbations to unlabeled images, the weak and the strong ones. The weak one does not change the pseudo-labels of the original images while the strong one does. Inspired by mixup method [29], we control the perturbation magnitude by linearly interpolating between the u_i and $u_i + \mathcal{P}_t^*(u_i)$:

$$\hat{u}_i = u_i + \omega_1\mathcal{P}_t^*(u_i). \tag{17}$$

For weak perturbation, we sample the ω_1 from $Beta(\alpha, \beta)$ where we set $\alpha = 1.0, \beta = 9.0$ in all experiments to control the expectation of ω_1 at the value of nearly zero. In contrast, we generate a strongly-perturbed version of the same image with trigger pattern pasted on it:

$$\hat{u}_i^\delta = u_i^\delta + (1 - \omega_2)\mathcal{P}_t^*(u_i^\delta), \tag{18}$$

where ω_2 is randomly sampled from the same beta distribution. The weakly-perturbed and strongly-perturbed images are jointly collected and injected into the unlabeled dataset as the final poisoned dataset to perform the backdoor.

Discussion: Assuming the unlabeled sample u_i is correctly pseudo-labeled as class s and its adversarially poisoned version is pseudo-labeled as target class t in the learning process. Then, the cross-entropy loss on the two contrastively crafted samples ($\hat{u}_i$ and $\hat{u}_i^\delta$) can be approximated as:

$$\begin{aligned}\ell(\hat{u}_i, \hat{u}_i^\delta, \theta) &= -(\log(f_t(\hat{u}_i^\delta, \theta)) - \log(f_t(\hat{u}_i, \theta)))\\ &\approx -(f_t'(\hat{u}_i^\delta, \theta) - f_t'(\hat{u}_i, \theta)),\end{aligned} \tag{19}$$

where f_t' is the penultimate output of f_t (the 1-D input of the last fully-connected layer). Considering that Δ is a small perturbation, we have $\mathcal{P}_t^*(u_i^\delta) = \mathcal{P}_t^*(u_i + \Delta) \approx \mathcal{P}_t^*(u_i)$. Then, using Taylor expansion, we have:

$$\begin{aligned}&f_t'(\hat{u}_i^\delta, \theta) - f_t'(\hat{u}_i, \theta)\\ &\quad\approx \left(f_t'(u_i^\delta + \mathcal{P}_t^*(u_i^\delta), \theta) - f_t'(u_i, \theta)\right)\\ &\quad\quad+ \left\{\omega_2\frac{\partial f_t'}{\partial u}|_{u=u_i^\delta+\mathcal{P}_t^*(u_i^\delta)} - \omega_1\frac{\partial f_t'}{\partial u}|_{u=u_i}\right\}\mathcal{P}_t^*(u_i^\delta)\\ &\quad\quad+ \frac{\partial f_t'}{\partial u}|_{u=u_i^\delta+\mathcal{P}_t^*(u_i^\delta)}\Delta.\end{aligned} \tag{20}$$

Maximizing the third term is equal to maximizing the gradient of f_t' in the direction of Δ, which means the model's sensitivity on the trigger pattern is maximized. The second term is the adversarial pattern's influence on the model, which can be alleviated by minimize its coefficient through controlling the ratio between ω_1 and ω_2. In our experiments, we found that $\omega_1 = \omega_2$ successfully alleviates the model's attention on the adversarial pattern and improves the attack success rate.

Algorithm 2 Generating Poisoned Data

```
Input: Unlabeled data $X_u$, Model parameters $\theta_0$, Trigger
       pattern $\delta$, Poisoned data ratio $\gamma$.
Output: Poisoned unlabeled data $X_{u,p}$.
1. Sample $\gamma$ precent of $X_u$ as $X_{u,s}$;
foreach $u_i$ in $X_{u,s}$ do
    2. Patch $u_i$ with trigger $\delta$ to get $u_i^\delta$;
    3. Calculate $\mathcal{P}_t^*(u_i)$ by optimizing Eq.16 with $u_i$ and
       $\theta_0$;
    4. Calculate $\mathcal{P}_t^*(u_i^\delta)$ by optimizing Eq.16 with $u_i^\delta$ and
       $\theta_0$;
    5. Sample $\omega_1, \omega_2$ 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. $X_{u,p} \leftarrow X_u \cup \{\hat{u}_i\} \cup \{\hat{u}_i^\delta\}$;
```

D. DeNeB Framework

The aforementioned Adversarial Perturbation Generator and Contrastive Data Poisoning strategy are integrated into our Deep Neural Backdoor framework (dubbed as "DeNeB"). 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 details, each image u_i is perturbed to generated two contrastively poisoned images ($\hat{u}_i$ and $\hat{u}_i^*$) according to Eq.17 and Eq.18. Then, the generated poisoned data is mixed into the original clean unlabeled data X_u to construct the poisoned dataset $X_{u,p}$, and the neural backdoor is conducted by feeding $X_{u,p}$ into the SSL system to perform training. The detailed generation scheme of the poisoned dataset $X_{u,p}$ is summarized in Algorithm 2.

E. Comparison With Clean-Label Attack

Due to the complicated attack design, launching DeNeB attack requires more detailed knowledge about the victim SSL system than the clean-label backdoor: 1) To generate the adversarially poisoned samples that are wrongly pseudo-labeled as the target class, we employ the white-box method [27] which requires the knowledge of network architecture and the network parameters θ_0 of the objective system. 2) To make the wrongly pseudo-labeled samples more durable during the SSL training process, we encode the feature of labeled data X_l into the samples. These extra information may be accessed by the adversary in a simple way. For example, if the DeNeB attack is launched to the edge nodes of distributed semi-supervised learning [21], [22], then both θ_0 and X_l can be acquired from the center server. Even if the adversary cannot access θ_0 and X_l, there are still many black-box methods that are able to generate adversarial samples without the knowledge of target network architecture [28]. As we focus on investigating the possibility of backdoor in SSL, and assessing its destructiveness, using black-box methods to perform DeNeB backdoor is beyond the scope of this work.

The leverage of arbitrary unlabeled data instead of category-specific data makes the DeNeB attack more practical and menacing than the clean-label backdoor. As an imitator of BadNets [7], DeNeB inherits the merits of dirty-label backdoor: it can achieve a high ASR by poisoning a few images that are randomly selected from the entire unlabeled data. Moreover, the ground-truth labels of the selected images are no longer required, which may greatly reduce the cost of the attack.

V. Towards Secure Semi-Supervised Learning

In this section, we propose a detection-and-purification defense (DePuD) framework for defense against the aforementioned backdoor and enable secured semi-supervised learning under the adversarial environment.

A. DePuD Framework Design Motivation

The data-poisoning based backdoor has triggered a wide panic to supervised learning systems. To deal with them, a large number of defense methods have been proposed from two different motivations: try to detect and remove poisoned data-label pairs in a well-annotated dataset [13]–[15] or try to erase the backdoor in a well-trained deep model [30]–[34]. From the perspective of poison data detection, most previous studies formulate this problem in anomaly detection and solve it using autoencoder [13] or clustering algorithms [14], [15]. However, in semi-supervised learning, the unstructured unlabeled data contains images from all categories, making its distribution too complicated to perform anomaly detection. For backdoor erasing methods, most previous works assume a small number of clean training data is acquirable and use finetuning [30], [31], fine-pruning [35], knowledge distillation [34] or other techniques [32] to repair the backdoored model parameters. However, these methods have reported noteworthy impairment to the model's performance ($\sim$ 3% for the state-of-the-art methods [34]). Some other backdoor erasing studies reconstruct a possible trigger pattern to a poisoned model first, then perform finetuning on the model to erase the reverse-engineered backdoor [31], [36]. Their performance largely depends on the resemblance between the reverse-engineered and the actual trigger patterns.

In this work, we show that organically combining the detection and backdoor erasing in a single defense framework will produce interesting results: we successfully defend against the aforementioned backdoor attack, making the final model have even higher accuracy than the clean SSL baseline. Although it is reasonable that the poison pattern found by the detection process should be extremely useful in the backdoor erasing process, none of the previous work has integrated them into a unified framework. In this work, we design a CNN-based poison data detector that can produce not only instance-level poisoning indicators, but also pixel-level poisoning indicators. Then, in the subsequent SSL training procedure, the model

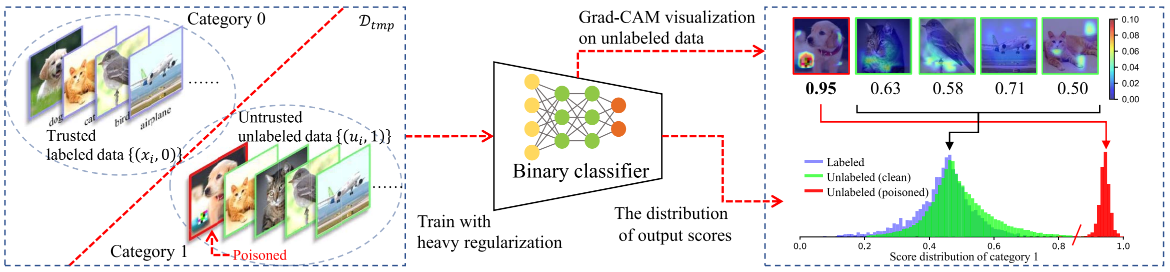

Category 0
$\mathcal{D}_{tmp}$
dog
cat
bird
airplane
Trusted labeled data $\{(x_i, 0)\}$
Untrusted unlabeled data $\{(u_i, 1)\}$
Category 1
Poisoned
Binary classifier
Train with heavy regularization
Grad-CAM visualization on unlabeled data
The distribution of output scores
0.95
0.63
0.58
0.71
0.50
0.10
0.08
0.06
0.04
0.02
0.00
Labeled
Unlabeled (clean)
Unlabeled (poisoned)
0.0
0.2
0.4
0.6
0.8
1.0
Score distribution of category 1

Fig. 3. The illustration of our poison data detector for semi-supervised learning. By suspiciously taking all unlabeled data as poisoned data against all trusted data as clean data, we train a binary classifier under heavy regularization terms. The regularization term is designed to penalize its attention on small, non-sense clues, making it cannot distinguish the clean data (the blue and green bars on the right side), while the poisoned data (the red bars) is still differentiable due to the fixed and overt trigger patterns. Moreover, the Grad-CAM visualization on category 1 clearly shows the position of the trigger pattern in poisoned images.

tries to ignore the pixels with high indicators, making the backdoor attack ineffective.

The proposed framework contains two stages. In the first stage, a *poison data detector* is proposed to detect poisoned images from the whole unlabeled data collection. Then in the second stage, *poison data purification strategies* are devised in the semi-supervised learning process to eliminate the influence of the poisoned images. In the following subsections, we introduce the two stages respectively and present the overall framework in the last subsection.

B. Poison Data Detection and Position

Here, we propose a simple but effective poison data detector: using a strictly regularized convolutional neural network (CNN) to automatically find out the poisoned data, as shown in Fig. 3. We construct a two-category temporary dataset $\mathcal{D}_{tmp} = \mathcal{D}_{tmp,0} \cup \mathcal{D}_{tmp,1}$ to train the CNN. Images from the labeled dataset are taken as category 0 (trusted) $\mathcal{D}_{tmp,0} = \{(x_i, 0)\}_{i=1}^{L}$, and images from the unlabeled dataset are taken as category 1 (untrusted) $\mathcal{D}_{tmp,1} = \{(u_j, 1)\}_{j=1}^{U}$. Then the deep CNN is trained to distinguish the two categories. Since these two image collections share images from the same classes, the deep CNN will unearth any small clues that can separate the two image collections. This phenomenon is known as "overfitting" in machine learning. Therefore, we utilize standard regularization methods to penalize the CNNs' sensitivity on meaningless small clues, let it focus on the most distinct features in untrusted data collections. If there is any trigger pattern hidden in the unlabeled dataset, then the pattern will be noticed by the classifier as a useful clue to category 1 due to its repeated emergence in the unlabeled dataset.

By designing a moderate regularization strategy, CNN stops overfitting on clean data while it is still able to distinguish the poisoned data, as shown on the right side of Fig. 3. After training and obtaining the weights of the classifier θ_b, we directly feed the untrusted image u to the classifier and take the score of category 1 as the indicator of the probability of u being poisoned.

Further, we employ Grad-CAM [20], a visual explanations technique, to localize the important regions of u with respect to category 1, where the gradient information flowing into the last convolutional layer is used to assign an importance value to each pixel of u on the decision of category 1.

$$m = GradCAM(u, \theta_b, \text{category 1}). \tag{21}$$

We refer to m as the confidence mask for image u.

1) Regularization Strategy Design: Due to its massive amount of parameters, CNN is peculiarly prone to overfitting. We consider using the combination of the following regularization techniques to alleviate overfitting:

a) Mixup regularization [29]: Mixup regularization train the neural networks on the linear combinations of pairs of training examples and their labels:

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1-\lambda) x_j, \quad // x_i, x_j \text{ are raw inputs} \\ \tilde{y} &= \lambda y_i + (1-\lambda) y_j, \quad // y_i, y_j \text{ are one-hot labels}, \end{aligned} \tag{22}$$

where $\lambda \in [0, 1]$ is randomly drawn from $Beta(\alpha, \alpha)$, (x_i, y_i) and (x_j, y_j) are two samples that are randomly drawn from the training dataset. The mixup regularization forces the neural network to comply with linear behavior in-between different categories. In this work, we employ mixup regularization on $\mathcal{D}_{tmp}$. Since clean samples may be labeled as category 0 or 1 in $\mathcal{D}_{tmp}$, linear interpolation will lead to label-confusing phenomena in clean data space where the target outputs in this space are ambiguous, hampering the overfitting of CNNs in this space.

b) l_1 and l_2 weight decay: Weight decay is one of the most ancient and effective regularization methods. The l_1 weight decay, known as "Lasso" in machine learning, is considered an effective feature selection method. However, in deep learning, the l_1 weight decay is only employed in model pruning [37] to the best of our knowledge. Here we incorporate the weight penalties into the training objective and demonstrate its effectiveness in filtering out meaningless small clues.

In summary, parameters of the binary classifier θ_b are optimized with the following objective function:

$$\begin{aligned} \text{minimize} \quad & \mathbb{E}_{(\tilde{x},\tilde{y}) \sim Mixup(\mathcal{D}_{tmp}, \alpha_{mix})}[\ell(f(\tilde{x}, \theta_b), \tilde{y})] \\ & + \lambda_{\ell_1} \|\theta_b\|_1 + \lambda_{\ell_2} \|\theta_b\|_2, \end{aligned} \tag{23}$$

where α_{mix} is the Beta distribution parameter in mixup and λ_{ℓ_1}, λ_{ℓ_2} are hyperparameters in adjusting the magnitude of weight decay terms.

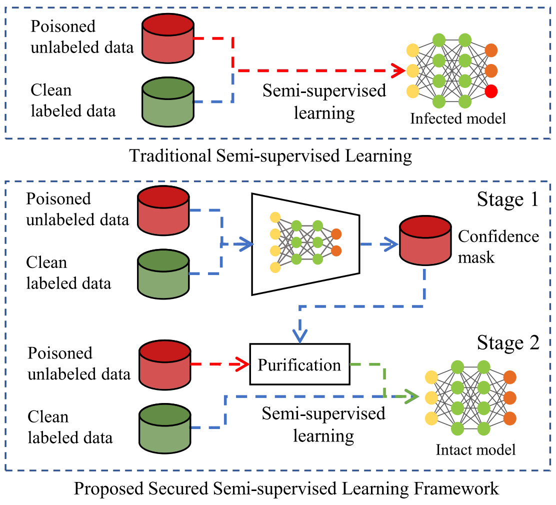


Fig. 4. Illustration of conventional SSL process and the proposed Secured SSL framework. We design a purification process before the feeding of unlabeled data. The poisoned features in the unlabeled data are localized in the first stage and blocked in the second stage.

C. Poison Data Purification

The simplest poison data purification method is to discard the suspicious data, and retrain a new model from scratch using the remaining data for training, which is mostly adopted by the previous researches that focus on the detection process [14], [15]. However, holding the detailed confidence masks generated in the previous subsection that indicate the positions of poisoned features, we consider using more precise methods to filter out the poisoned features. Specifically, during the normal SSL training process, we sample each unlabeled image u_i and apply purification operation to wipe out its poisoned features, then we feed the purified version $\tilde{u}_i$ into SSL training process. We consider the following purification strategies:

1) Occlusion-Based Poison Purification: Let the $m_i \in [0, 1]^{h\times w}$ be the confidence mask that crafted by Grad-CAM visualization on u_i, where $(m_i)_{ij}$ encodes the probability of $(u_i)_{ij}$ belongs to poisoned pattern. We set the pixels of high probabilities to be close to the mean pixel value p_{mean}:

$$\tilde{u}_i = m_i \otimes p_{mean} + (1 - m_i) \otimes u_i, \tag{24}$$

where $\otimes$ is Hadamard product.

2) Differential-Privacy Based Poison Purification: Differential privacy has been proposed to defend against differential attacks, for example, it is used to prevent certain changes of data source leading to distinct changes of output that can reflect the data source changes [13]. From the perspective of backdoor defense, we take the backdoor reaction of the poisoned model as the "distinct changes" caused by small modification of training data, and employ the technique of differential privacy to alleviate the "distinct changes":

$$\tilde{u}_i = Clip(u_i + \epsilon \otimes m_i, 0, 1), \tag{25}$$

where ϵ is randomly sampled from $\mathcal{N}(0, \sigma^2)$. As common practice in differential privacy, we directly add Gaussian noise to the pixels of high probabilities.

Moreover, the above two methods can be further combined:

$$\tilde{u}_i = Clip\big((1 - m_i) \otimes u_i + m_i \otimes p_{mean} + \epsilon \otimes m_i, 0, 1\big). \tag{26}$$

D. Implementation of Secure SSL

We fortify the vulnerable semi-supervised learning with the aforementioned poison data detector and purification strategy as our DePuD framework, as shown in Fig. 4. Specifically, in stage one, the proposed detector is employed to generate a confidence mask for each unlabeled data. Then in stage two, we conduct the semi-supervised learning as usual, except in the sampling process of each training iteration, we apply the proposed data purification method to the unlabeled training data. Since the confidence mask only highlights the positions of excessive obvious features, the purification process blocks the poisoned features while the useful information in the poisoned images is still kept. The defense framework composed of the above two steps is succinct and easy to integrate with any SSL algorithms.

VI. Experiments

In this section, we first conduct the clean-label backdoor to SSL algorithms to demonstrate the discovered vulnerability. Then, the effectiveness of the DeNeB scheme is evaluated in comparison with the clean-label backdoor. Finally, experiments are conducted on the proposed defense framework.

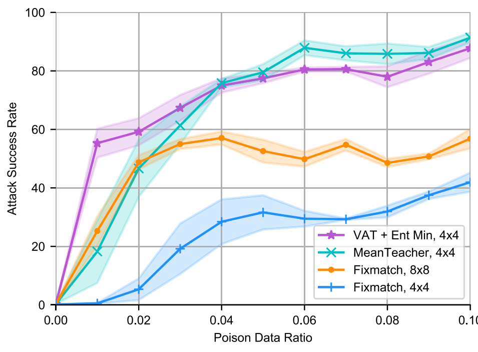
A. Clean-Label Backdoor

The backdoor scheme of using the clean-label data in semi-supervised learning is fully described in Section III. Experiments are performed to demonstrate its effectiveness in this subsection.

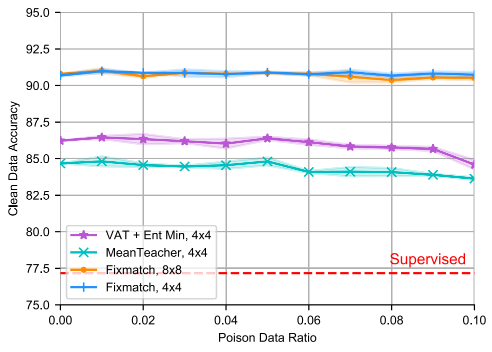
1) SSL Baseline Setup: Three recently proposed SSL algorithms are employed as the attack victims to show that the discovered vulnerability commonly exists in SSL algorithms (MeanTeacher [2], VAT [3], and Fixmatch [4]).

We follow most of the settings described in their papers as the victim SSL algorithms in our experiments. For the victim neural network, we use the ResNet architecture (a Wide ResNet-28-2 [38] with 1.5M parameters). As the common practice in SSL evaluations, fully-annotated image dataset CIFAR10 [25] are employed. We randomly choose 4000 images from the whole dataset as "labeled collection" $\mathcal{D}_l$ and take the remaining images $\mathcal{D}_u = (X_u, Y_u)$ as "unlabeled collection". The labels of the "unlabeled collection" Y_u are provided to the adversary for conducting clean-label backdoor in this section, but are hidden from semi-supervised learners.

2) Attack Experiments Setup: The adversary performs the described poison procedure on the "unlabeled collection" $\mathcal{D}_u = (X_u, Y_u)$, then the victim semi-supervised learner takes the poisoned unlabeled data $X_{u,p}$ and clean labeled data $\mathcal{D}_l = (X_l, Y_l)$ to train a model from scratch. In the attack experiments on CIFAR10, the "truck" class is treated as the target class and a part of unlabeled images from "truck" is poisoned by the described clean-label backdoor process. Different size of triggers are employed, as shown in Fig. 6.

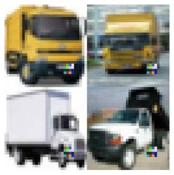


(a)

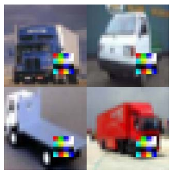


(b)

Fig. 5. Results of clean-label backdoor on various SSL algorithms using different poisoned data ratios (γ).



(a)



(b)

Fig. 6. The backdoor triggers used in clean-label backdoor (trigger size of 4 × 4 (a) and 8 × 8 (b)).

TABLE I
SUMMARY OF THE DETAILED SETTINGS OF VICTIM SSL BASELINES

Detailed Settings	
Dataset	SVHN / CIFAR10
Num of labeled	4000
Num of unlabeled	46000
batch size	64
Network architecture	Wide ResNet-28-2 [38]
Optimizer	SGD
MT [2] settings	$initial\ lr = 0.0004$, $\lambda_u = 8$, $\alpha_{ema} = 0.95$.
VAT [3] settings	$initial\ lr = 0.003$, $\lambda_u = 0.3$, $\xi = 1e-6$, $\epsilon = 6$.
Fixmatch [4] settings	$initial\ lr = 0.003$, $\lambda_u = 1$, $\tau = 0.95$, $\eta = 5$, $\ell_2\ weight\ decay = 5e-5$.

A simple trigger injecting strategy is adopted in the experiments: In the beginning, a 4 × 4 trigger pattern is randomly generated and resized to the desired size by bilinear interpolation. Then, for each image to be poisoned, we choose an area of the same size in the right-bottom corner of the image and replace the pixels with the trigger pattern.

3) Clean-Label Backdoor Results: Fig. 5 shows the result of primitive clean-label backdoor on the selected SSL algorithms. Fig. 5(a) shows that the SSL algorithms are all infected by the backdoor to some extend. It also indicates that the ASR of the backdoor increases with more poisoned data. Fig. 5(b) implies that the clean data accuracy (CA) of the model under attacks is almost consistent with the normal SSL model ($\gamma = 0$), which means the backdoor would not corrupt the model's performance on normal test data. Therefore, the backdoor attack is invisible to the semi-supervised learner. These phenomena clearly verify the hypothesis that SSL is a potential victim of backdoor attacks via unlabeled images.

TABLE II
WE CONDUCT THE ENHANCED CLEAN-LABEL BACKDOORS WHICH ARE DESIGNED FOR SUPERVISED LEARNING. HERE WE ADAPTED THEM TO SSL AND LIST THEIR PERFORMANCE ON THE SSL TASK

Method	**Fixmatch [4]** $\gamma = 0.01$		**Fixmatch [4]** $\gamma = 0.05$	
	CA	ASR	CA	ASR
Clean data	93.08	N/A	93.08	N/A
Clean-Label	93.24	28.30	92.82	39.74
Turner *et al.* [9]	92.90	1.66	92.96	17.84

Though the adapted clean-label backdoor shows its feasibility in semi-supervised settings, its performance may be unattractive to the adversary. In the attacks on VAT and MeanTeacher, the clean-label backdoor achieves ∼ 90% ASR with all target-class images being poisoned. However, these two algorithms have been considered outdated due to the overwhelming performance of the newest SSL studies. In the attacks on Fixmatch, the clean-label backdoor only achieves ∼ 40% ASR with all target-class images being poisoned, and ∼ 60% ASR after the trigger size is doubled.

The enhanced data-poisoning clean-label backdoor [9] has been reimplemented and the results are shown in Table II. However, we found that these enhancing strategies designed for supervised learning are no longer effective under SSL scenario. One explanation is that the poisoned images of [9] may have troubles in SSL learning. To improve the ASR under supervised settings, Turner *et al.* [9] corrupts the object feature in these images by generative modeling or adversarial perturbation to make the model's decision more dependent on the trigger pattern. However, in SSL, corrupting the object features makes it difficult for these poisoned images to be correctly learned by the SSL algorithm, thus these images cannot effectively infect the target model anymore.

B. Proposed DeNeB

In this subsection, we conduct experiments to evaluate the proposed DeNeB attack, and provide comprehensive ablation studies to illustrate the possible scheme that the adversary could take to ensure a successful backdoor attack.

1) SSL Baseline Setup: We employ Fixmatch [4] as the SSL baseline algorithm to evaluate the proposed scheme, which has

TABLE III

RESULTS OF THE BACKDOOR ON ONE TARGET CLASS OVER MULTIPLE DATASETS. FOR SVHN, WE TAKE "2" AS THE TARGET CLASS, FOR CIFAR10, WE TAKE "TRUCK" AS THE TARGET CLASS

Dataset	SVHN (N_p = 500)		SVHN (N_p = 1000)		CIFAR10 (N_p = 500)		CIFAR10 (N_p = 1000)	
	CA	ASR	CA	ASR	CA	ASR	CA	ASR
Supervised Baseline	91.90	N/A	91.90	N/A	76.77	N/A	76.77	N/A
Clean SSL Baseline	96.43	0.67	96.43	0.67	88.95	1.19	88.95	1.19
Clean-label backdoor	96.11	10.17	95.98	24.56	89.05	19.84	89.01	43.12
DeNeB	96.12	**53.29**	96.17	**85.76**	88.97	**91.90**	88.97	**93.37**

TABLE IV

PER-CLASS BACKDOOR RESULTS OF CIFAR10 DATASET

Target Class	Clean-Label (N_p = 500)		DeNeB (N_p = 500)	
	CA	ASR	CA	ASR
airplane	89.09	31.94	89.12	89.27
automobile	89.03	32.72	89.13	95.18
bird	88.92	69.29	88.92	87.13
cat	89.02	66.40	89.17	87.37
deer	89.14	37.78	89.14	82.47
dog	89.06	60.33	89.02	94.84
frog	89.17	20.12	89.06	62.00
horse	88.93	39.32	89.16	76.31
ship	89.05	22.04	88.57	74.10
truck	89.05	19.84	88.97	91.90

been demonstrated to be the most effective SSL algorithm with both high performance and robustness in previous clean-label backdoor experiments (Fig. 5). For fair comparison, we adopt consistent hyperparameters that are proposed in original paper across all experiments ($\lambda_u = 1$, initial learning rate $\eta = 0.003$, confidence threshold $\tau = 0.95$, batch size $B = 64$). Due to the number of experiments to validate the backdoor efficiency, we reduce the training epochs from 1024 to 200 for each backdoor attempt, thus the model accuracy we reported is below the original paper [4].

2) Attack Experiments Setup: Considering the assumptions of DeNeB that the adversary should have access to the initial model parameters θ_0 used by the learner, we design the following experiment steps to evaluate the DeNeB attack on SSL algorithms:

Step 1. Pre-train an initial model: The semi-supervised learner is assumed to have a coarse model θ_0 and is going to finetune it on collected unlabeled data. In our experiments, the coarse model θ_0 is given by pre-training a randomly initialized neural network on the labeled dataset $\mathcal{D}_l = (X_l, Y_l)$. We employ the SGD optimizer with cosine learning rate schedule, and optimize the network for 50 epochs. The accuracy of coarse models is reported as "Supervised Baseline".

Step 2. Generate poisoned dataset: Using the coarse model θ_0 and labeled dataset $\mathcal{D}_l = (X_l, Y_l)$, the adversary performs the DeNeB poisoning process (Algorithm 2): using clean unlabeled data X_u to generate poisoned unlabeled data $X_{u,p}$. We employ the same trigger injecting strategy with the clean-label backdoor experiments. The poisoned examples are shown in Fig. 8. For experiments in this section, we denote the poisoning budget as the number of poisoned data N_p instead of the poison data ratio γ for clear comparison.

Step 3. Perform SSL training on poisoned data: Here the innocent semi-supervised learner takes the poisoned unlabeled data $X_{u,p}$ and clean labeled data $\mathcal{D}_l = (X_l, Y_l)$ to finetune the coarse model θ_0 using semi-supervised algorithms. For fair comparison, the clean SSL baseline and the clean-label backdoor experiments in this section are also initialized with the same coarse model θ_0. After the training is completed, we evaluate the final model by CA and ASR metrics which have been described in Section II.D.

3) Dataset and Poison Settings: We evaluate the DeNeB on SVHN [26] and CIFAR10 [25] datasets which are commonly employed in evaluating both SSL algorithms [2], [4] and the backdoor [8]. For each category in SVHN and CIFAR10, we randomly choose 400 training images as labeled data and use the remaining images as unlabeled data. Thus there are 4000 class-balanced labeled images in total. To generate the poisoned images, we set trigger patch size 8×8, $\lambda = 0.01$, $\beta = 9$. We perform PGD optimization in Algorithm 2 for 1000 iterations with a learning rate of 0.01 that decays every 200 iterations by 0.95.

4) Results: The results on SVHN and CIFAR10 datasets are shown in Table III and Table IV. We fix the target class to be "2"/"truck" for SVHN/CIFAR10 in Table III and provide per-class results of CIAR10 in Table IV. The results show that when the same number of poisoned data is injected, the ASR of DeNeB is substantially higher than that of the clean-label backdoor. In the meantime, the accuracy of SSL models on clean test data (CA) is maintained. As shown by Fig. 5, even if a large amount of training data is poisoned, the clean label backdoor can only achieve about 60% ASR. In contrast, DeNeB achieves above 90% ASR when 500 poisoned images are injected into the total 50000 unlabeled training images.

5) Ablation Studies on CIFAR10: Extensive ablation studies are performed to illustrate how the adversary chooses the right hyperparameters to perform a successful backdoor attack. Starting from the previous experimental settings, we explore the sensitivity of hyperparameters in the DeNeB.

a) Adversarial perturbation generator (APG): To generate adversarial samples that are persistent to the updating model which keeps adjusting during the whole training process, a complex generator is designed in DeNeB with many hyperparameters: 1) λ in the objective function (Eq. 16), 2) the adversarial perturbation budget ϵ, and 3) settings of PGD optimization (number of iterations, learning rate, and decay schedule). The last is omitted because the PGD optimization will convergent after a sufficient number of iterations [27]. For λ in our perturbation objective function, we choose λ from

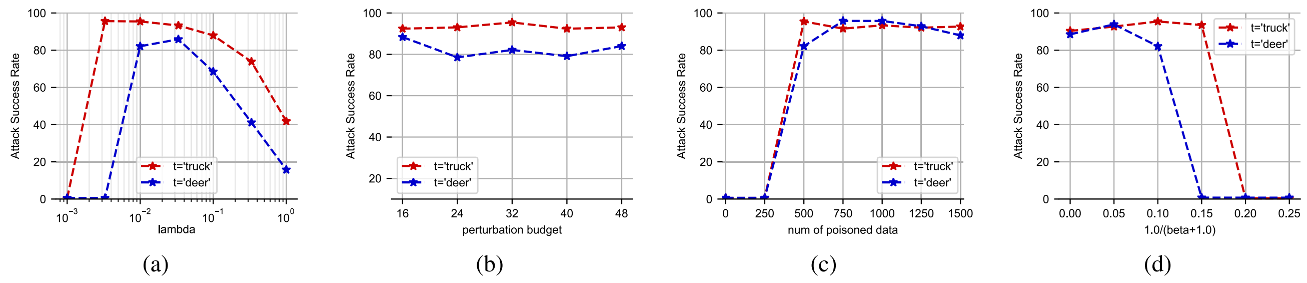


Fig. 7. Plots of various ablation experiments. (a) Varying the λ in adversarial perturbation objective function. (b) Varying the perturbation budget ϵ. (c) Varying the beta distribution parameter β (the expectation of ω is $1/(\beta+1)$ when $\alpha = 1$). (d) Varying the number of poisoned data N_p.

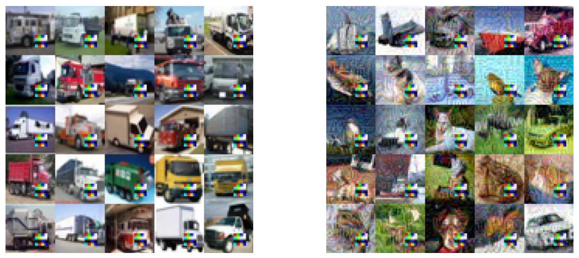


Fig. 8. The poisoned examples craft by the clean-label backdoor (left) and the proposed DeNeB (right), where the target class is "truck". The proposed DeNeB takes unlabeled images from arbitrary classes instead of only the target class to poison.

$[0.001, 1.0]$ and generate $N_p = 500$ poisoned images for each setting with perturbation budget $\epsilon = 32$. As shown in Fig. 7 (a), the backdoor fails when the λ is too small. It seems that encoding the features of labeled data into poisoned data provides durability for the adversarial perturbation. Fig. 7 (a) also shows that $\lambda = 0.01$ works well for different target class, thus the adversary could set the λ to a fixed value across different attack. As for the perturbation budget ϵ, we generate poisoned images with $\epsilon \in [16, 24, 32, 40, 48]$ and show the backdoor results in Fig. 7 (b). The results turn out to be stable, indicates that the adversary can choose a moderate ϵ across a wide span. These results demonstrate that the hyperparameters in APG can be easily decided by the adversary.

b) Contrastive data poisoning (CDP): To increase the ASR of the DeNeB attack, we proposed the CDP strategy which only includes one hyperparameter: β which controls the magnitude of the CDP. The magnitude of the CDP can be taken as the expectation of $Beta(1, \beta)$ calculated as $1/(1+\beta)$. Here we start with no CDP at all ($\beta \to \infty$) and gradually increase its magnitude (that means to decrease the β), the results are shown in Fig. 7 (c). It can be seen that the ASR raises at the start, but the backdoor attempt fails when the expectation of mix coefficients is higher than 0.15.

c) Overall scheme: Other hyperparameters also play an important role in the DeNeB attack. In backdoor attacks, it is often the case that the more data is poisoned, the more likely the attack is to be successful. Here we perform ablation studies to show at least how many poisoned samples are required for a successful attack. The results are shown in Fig. 7 (d), which show that in the CIFAR10 backdoor attack, 500 poisoned unlabeled images are enough to implant a strong backdoor into the SSL model, accounting for 1% of the overall unlabeled

TABLE V

EFFECTIVENESS OF THE PROPOSED DEFENSE FRAMEWORK ON CIFAR10

Backdoor Method (t='truck')	Clean-Label $N_p = 500$		DeNeB $N_p = 500$	
	CA	ASR	CA	ASR
Supervised baseline	76.77	N/A	76.77	N/A
Clean SSL baseline	88.95	1.19	88.95	1.19
Poisoned SSL baseline	89.05	19.84	88.97	91.90
OC	90.26	15.15	90.38	47.22
DP ($\sigma = 50/255$)	89.54	41.18	90.44	66.22
DP ($\sigma = 100/255$)	88.79	24.09	89.85	2.00
DP ($\sigma = 150/255$)	87.63	7.423	89.73	1.40
DP ($\sigma = 200/255$)	86.59	4.233	89.37	0.95
OCDP ($\sigma = 50/255$)	89.49	2.42	90.34	39.62
OCDP ($\boldsymbol{\sigma = 100/255}$)	88.19	1.59	90.02	1.07
OCDP ($\sigma = 150/255$)	87.20	1.60	89.70	0.98
OCDP ($\sigma = 200/255$)	85.83	1.65	89.26	1.12
Finetuning ($lr = 0.1$)	84.19	8.52	84.71	27.62
Fine-pruning [35]	85.22	7.88	84.33	27.41
MCR [32]	84.27	8.32	83.95	18.38
NAD [34]	83.97	8.54	84.79	19.54

data. Compared with the clean-label backdoor, DeNeB greatly reduces the quantitative requirements of poisoned data.

C. Backdoor Defense

After demonstrating the backdoor on semi-supervised learning, in this subsection, we evaluate the performance of the proposed defense framework DePuD.

1) Defense Settings: We strengthen the SSL baseline with the DePuD framework and attack it with the described backdoor methods. Specifically, in the first stage, we adopt fixed parameters in all experiments to generate a confidence mask for each unlabeled image: α_{mix}=4.0, λ_{ℓ_1}=λ_{ℓ_2}=$1e-6$. Then in the second stage, the same SSL procedure as previous attack experiments is performed on the poisoned data. The only difference is that the data sampling process is equipped with the following purification strategies: **1) OC:** Occlusion-based purification. **2) DP:** Differential-privacy based purification. **3) OCDP:** The combination of the above two strategies.

2) Results: The results of the proposed DePuD with different purification strategies are shown in Table V. The occlusion-based purification strategy (OC) lowers the ASR by a lot. Meanwhile, this strategy obtains an improvement of about 1.3% in model accuracy compared with the baseline. This indicates some misleading information is occluded as

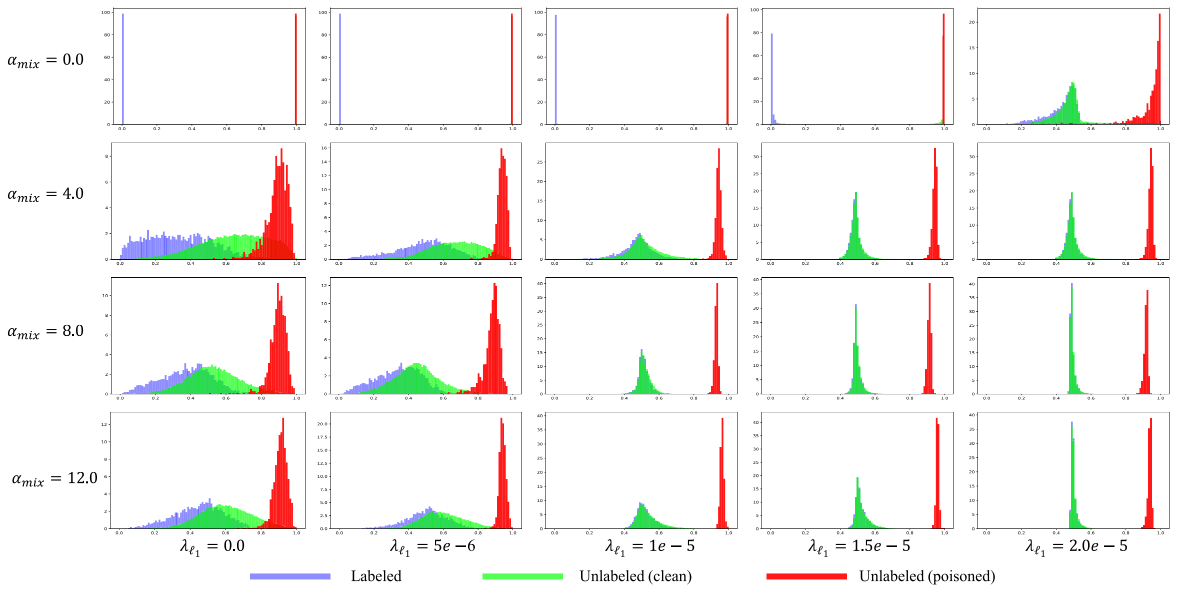


Fig. 9. The effect of mixup regularization and l_1 weight decay used in the poison data detection process (we adjust α_{mix} and λ_{ℓ_1} while λ_{ℓ_2} is fixed to $1e-6$). Each subgraph shows the output distribution of different image sets after training with corresponding coefficients. The labeled set contains 4000 clean samples from CIFAR10 and the unlabeled set contains 46500 samples (the remaining 46000 clean samples + 500 poisoned samples).

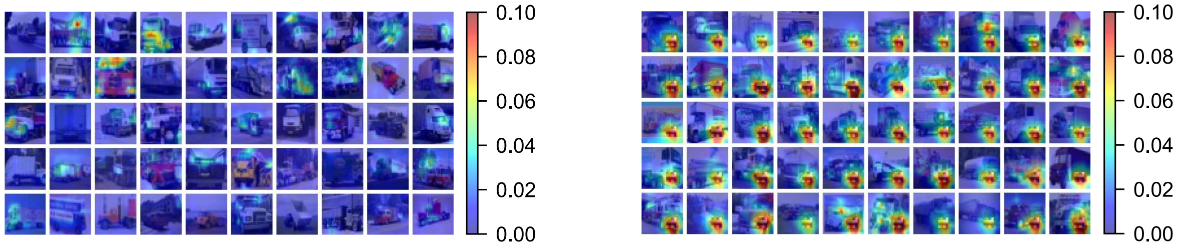


Fig. 10. Grad-CAM visualization of the binary classifier on clean data (left) v.s. poisoned data (right). Grad-CAM [20] assigns a scalar value to each pixel in the input image to indicate its influence on the model decision. The visualization shows that the trigger pattern in the right-bottom corner is the most interesting area of the binary classifier.

well as the trigger pattern in the unlabeled dataset. The differential-privacy based strategy (DP) is performed with different σ. The results show that the larger σ leads to a more robust model, but the accuracy of the model decreases slightly. Moreover, the results of OCDP show these two strategies can be combined to achieve a better trade-off between model accuracy and robustness.

3) Effectiveness of Regularization Terms: Here, we train multiple binary classifiers under different hyperparameters (α_{mix} and λ_{ℓ_1}) and visualize their output distributions of clean data and poisoned data, the results shown in Fig. 9 demonstrate the effect of these regularization terms. When training the model without regularization ($\alpha_{mix} = 0$, $\lambda_{\ell_1} = 0$), the labeled set and unlabeled set is completely split by the binary classifier. The mixup regularization pushes the distributions of the clean labeled set and clean unlabeled set to overlap, thereby preventing the classifier from overfitting in the clean data region. The l_1 weight decay penalizes the classifier's attention on the small, meaningless features, making the divergence between clean and poisoned sets more distinct. The combination of these two regularization terms makes the binary classifier a good poison data detector. Moreover, Fig. 10 shows the confidence map generated by Gram-CAM visualization on binary classifier (α_{mix}=4.0 and λ_{ℓ_1}=$1e-5$). it clearly shows that the most influential regions to the classifier's decision always point to the trigger pattern.

4) Compare With Backdoor Erasing Methods: We compare the DePuD with backdoor erasing methods: 1) the primitive finetuning method which direct finetunes the trained model on the clean labeled data, 2) Fine-pruning [35], 3) MCR [32], and 4) NAD [34]. We employ the final model of "Poisoned SSL baseline" as input and erase its backdoor by finetuning, pruning, or distillation on the clean labeled data X_l. The results are reported in Table V. Those methods result in much more detriment to the model's accuracy, and demonstrate inferior backdoor-erasing effectiveness.

5) Ablation Studies of the Detector: We train detectors under different attack settings (trigger size and N_p) and different defense hyperparameters (α_{mix}, λ_{ℓ_1}), and use *Area Under the Curve* (AUC) metric to measure the discrimination ability of detectors' output score between clean and poisoned data. AUC=1.0 indicates the poisoned data can be totally separated by a fixed threshold on the detectors' output while AUC=0.5 means the detectors are failed to notice the poisoned data. The results are shown in Fig. 11, which demonstrate that there are some hyperparameter sets perform well across different attack settings, such as ($\alpha_{mix} = 6.0$, $\lambda_{\ell_1} = 1e-5$) and ($\alpha_{mix} = 4.0$, $\lambda_{\ell_1} = 1.5e-5$).

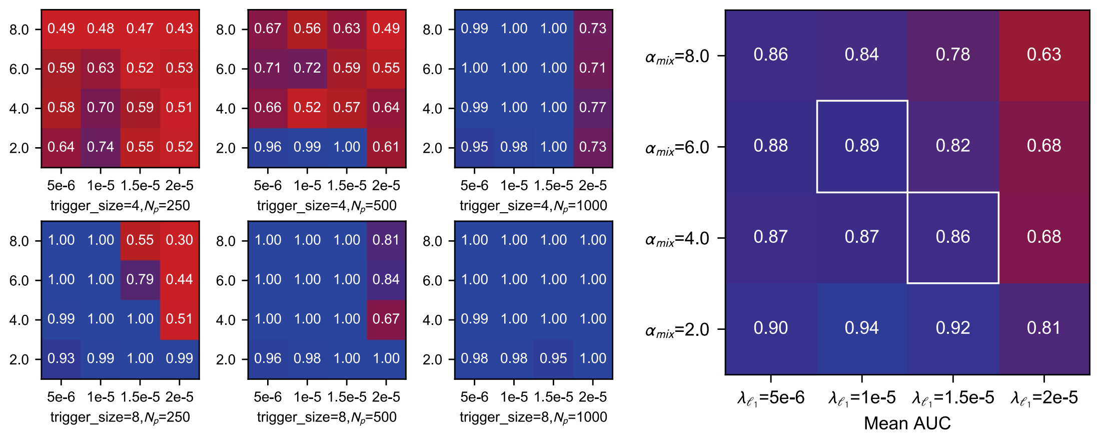


Fig. 11. We train detectors under different attack settings (trigger size and N_p) and different defense hyperparameters (α_{mix}, λ_{ℓ_1}). Area Under the Curve (AUC) metric is used to measure the detectors' discrimination ability of poisoned data.

	5e-6	1e-5	1.5e-5	2e-5
trigger_size=4, N_p=250				
8.0	0.49	0.48	0.47	0.43
6.0	0.59	0.63	0.52	0.53
4.0	0.58	0.70	0.59	0.51
2.0	0.64	0.74	0.55	0.52
trigger_size=4, N_p=500				
8.0	0.67	0.56	0.63	0.49
6.0	0.71	0.72	0.59	0.55
4.0	0.66	0.52	0.57	0.64
2.0	0.96	0.99	1.00	0.61
trigger_size=4, N_p=1000				
8.0	0.99	1.00	1.00	0.73
6.0	1.00	1.00	1.00	0.71
4.0	0.99	1.00	1.00	0.77
2.0	0.95	0.98	1.00	0.73
trigger_size=8, N_p=250				
8.0	1.00	1.00	0.55	0.30
6.0	1.00	1.00	0.79	0.44
4.0	0.99	1.00	1.00	0.51
2.0	0.93	0.99	1.00	0.99
trigger_size=8, N_p=500				
8.0	1.00	1.00	1.00	0.81
6.0	1.00	1.00	1.00	0.84
4.0	1.00	1.00	1.00	0.67
2.0	0.96	0.98	1.00	1.00
trigger_size=8, N_p=1000				
8.0	1.00	1.00	1.00	1.00
6.0	1.00	1.00	1.00	1.00
4.0	0.99	1.00	1.00	1.00
2.0	0.98	0.98	0.95	1.00

Mean AUC	λ_{ℓ_1}=5e-6	λ_{ℓ_1}=1e-5	λ_{ℓ_1}=1.5e-5	λ_{ℓ_1}=2e-5
α_{mix}=8.0	0.86	0.84	0.78	0.63
α_{mix}=6.0	0.88	0.89	0.82	0.68
α_{mix}=4.0	0.87	0.87	0.86	0.68
α_{mix}=2.0	0.90	0.94	0.92	0.81

VII. Related Works

A. Semi-Supervised Learning

Using the unlabeled data to improve generalization has a long and rich history. A large number of SSL-based neural networks have been widely applied to visual recognition [1], [4], object detection [39], and graph-structured learning [40], etc. However, most of these semi-supervised learning algorithms are conducted in experimental environments, where the unlabeled data are sampled from the same trusted source with the labeled training data. Such a prerequisite is difficult to hold in many practical applications.

A practical issue of SSL is that when unlabeled data contains a different distribution of classes than the labeled data, the performance of SSL may degrade drastically. To address this issue, Chen *et al.* [17] used joint self-distillation and out-of-distribution filtering to alleviate the performance degradation. Guo *et al.* [41] proposed a framework to selectively use the unlabeled data in the training process and theoretically guaranteed the mitigation of performance degradation. However, those works are only focused on preventing the performance degradation caused by unlabeled data, which is easy to detect by evaluating on a hold-out test set. In contrast, the proposed backdoor will not harm the SSL performance but will cause more serious consequences for SSL. To the best of our knowledge, this is the first work to achieve the backdoor and its defense in semi-supervised learning.

B. Neural Backdoor in Supervised Settings

The neural backdoor is first discussed in [7], which made us understand that the black-box of deep neural networks could be intentionally manipulated. Since then, many research had been conducted on improving its stealthiness to strength its threats. Generally, there are two ways to launch backdoor attacks: data-poisoning [5]–[12] and model replacing [30]. As a branch of data-poisoning backdoor, clean-label backdoor [9]–[12] tries to generate poisoned samples with semantically correct labels to evade scrutinization by human eyes. Turner *et al.* [9] introduced adversarial perturbations to weaken the features of original images and incline the model decision to trigger patterns. Zhao *et al.* [11] extended [9] to video recognition models. Shafahi *et al.* [10] and Zhu *et al.* [12] proposed instance-specific backdoor method based on clean-label poisoned images. Besides clean-label backdoor, there were many studies that attempted to hide trigger patterns in the poisoned dataset to make the attack more imperceptible [8]. Those methods achieved higher ASR and better concealment than the primitive clean-label backdoor. However, under SSL settings, those backdoor methods are either infeasible due to their dependence on labels, or ineffective due to the ambiguous learning procedure of SSL.

C. Backdoor Defense

A large number of backdoor defense methods have been proposed in supervised settings. We summarize the defense methods most related to our work into two branches: 1) detecting poisoned data, and 2) fixing backdoored models.

Poison data detection aimed at detecting and removing poisoned data during the training process by clustering [14] or singular value decomposition [15] in the feature space. These methods assumed that the feature distribution of data in a backdoor-targeted class will have two different sub-populations: one is the feature of clean data and the other is that of poisoned data [15]. Based on this assumption, the poisoned data can be separated by identifying the above different features. However, different from supervised learning where the training data are class-structured, the unlabeled data contains images from all classes so that its sub-populations are difficult to model. Thus, the existing methods to detect the poisoned data from labeled data during the training process were not suitable for the unlabeled data. Even if the poisoned data is identified, current methods simply discard all the poisoned data, wasting the useful information of poisoned data. To improve the practicality of active defense in SSL, the DePuD introduces a two-category discriminator to automatically discover the poisoned data and it supports pixel-level positioning.

Backdoored model fixing aimed at erasing the backdoor reactions in a well-trained model. Finetuning, pruning, and distillation are common techniques employed in the backdoor-erasing process [30]–[32], [34]–[36]. Without any priori information of backdoor, finetuning [30], pruning [35], and neural attention distillation [34] had been directly applied to backdoored models using a small number of clean labeled samples. Other techniques such as mode connectivity repair [32] had also been investigated. Their motivation was to suppress the abnormal response within the network, which was demonstrated to be harmful to the model's accuracy. Wang *et al.* [31] proposed to reconstruct a possible trigger pattern first, then utilize finetuning or pruning to eliminate the reconstructed backdoor. However, its performance largely depends on the resemblance between the reconstructed and the actual trigger patterns. Qiao *et al.* [36] extended the single reconstructed trigger in [31] to a staircase generator that can approximate the distribution of possible backdoor triggers. Despite their unstable results in reconstructing the backdoor trigger, they still impaired the model's accuracy after finetuning or distillation. The proposed DePuD demonstrated stable results in detecting and localizing trigger patterns of arbitrary size, and overwhelming performance in backdoor erasing without detriment to the model's accuracy.

VIII. Conclusion

Recently, semi-supervised learning had achieved spectacular success in various industries. To enhance the robustness and security of semi-supervised learning, in this paper, we identify a potential threat of using untrusted unlabeled data to train neural networks. To demonstrate the danger and destructiveness of such threat, we proposed a deep neural backdoor (DeNeB) scheme for semi-supervised learning, which is established based on unlabeled data. The experimental results on the natural image datasets have shown that the DeNeB is effective and insidious. To thwart the aforementioned backdoor attacks, we also proposed a detection-and-purification defense (DePuD) framework to mine the characteristics of backdoor triggers from the untrusted dataset and ultimately estimate their negative impacts.

References

[1] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Label propagation for deep semi-supervised learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5070–5079.

[2] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1195–1204.

[3] T. Miyato, S.-I. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1979–1993, Aug. 2019.

[4] K. Sohn *et al.*, "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 596–608.

[5] B. Nassi, Y. Mirsky, D. Nassi, R. Ben-Netanel, O. Drokin, and Y. Elovici, "Phantom of the ADAS: Securing advanced driver-assistance systems from split-second phantom attacks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 293–308.

[6] Y. Ji, X. Zhang, and T. Wang, "Backdoor attacks against learning systems," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2017, pp. 1–9.

[7] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "BadNets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47230–47244, 2019.

[8] A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden trigger backdoor attacks," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11957–11965.

[9] A. Turner, D. Tsipras, and A. Madry. (2019). *Clean-Label Backdoor Attacks*. [Online]. Available: https://people.csail.mit.edu/madry/lab

[10] A. Shafahi *et al.*, "Poison frogs targeted clean-label poisoning attacks on neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1–18.

[11] S. Zhao, X. Ma, X. Zheng, J. Bailey, J. Chen, and Y.-G. Jiang, "Clean-label backdoor attacks on video recognition models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14443–14452.

[12] C. Zhu, W. R. Huang, H. Li, G. Taylor, C. Studer, and T. Goldstein, "Transferable clean-label poisoning attacks on deep neural nets," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 7614–7623.

[13] M. Du, R. Jia, and D. Song, "Robust anomaly detection and backdoor attack detection via differential privacy," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–11.

[14] B. Chen *et al.*, "Detecting backdoor attacks on deep neural networks by activation clustering," in *Proc. Artif. Intell. Safety Workshop*, 2019, pp. 1–10.

[15] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8011–8021.

[16] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–18.

[17] Y. Chen, X. Zhu, W. Li, and S. Gong, "Semi-supervised learning under class distribution mismatch," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 3569–3576.

[18] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "DBA: Distributed backdoor attacks against federated learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–19.

[19] Z. Yan *et al.*, "DeHiB: Deep hidden backdoor attack on semi-supervised learning via adversarial perturbation," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 10585–10593.

[20] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[21] A. Albaseer, B. S. Ciftler, M. Abdallah, and A. Al-Fuqaha, "Exploiting unlabeled data in smart cities using federated edge learning," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2020, pp. 1666–1671.

[22] W. Jeong, J. Yoon, E. Yang, and S. J. Hwang, "Federated semi-supervised learning with inter-client consistency," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–15.

[23] D. Berthelot *et al.*, "ReMixMatch: Semi-supervised learning with distribution alignment and augmentation anchoring," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–13.

[24] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "MixMatch: A holistic approach to semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 5049–5059.

[25] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.

[26] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. Adv. Neural Inf. Process. Syst. Workshop*, 2011, pp. 1–9.

[27] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–28.

[28] G. Li, K. Ota, M. Dong, J. Wu, and J. Li, "DeSVig: Decentralized swift vigilance against adversarial attacks in industrial artificial intelligence systems," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3267–3277, May 2020.

[29] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "*Mixup*: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–13.

[30] Y. Liu *et al.*, "Trojaning attack on neural networks," in *Proc. 25nd Annu. Netw. Distrib. Syst. Secur. Symp.*, 2018, pp. 1–17.

[31] B. Wang *et al.*, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 707–723.

[32] P. Zhao, P.-Y. Chen, P. Das, K. N. Ramamurthy, and X. Lin, "Bridging mode connectivity in loss landscapes and adversarial robustness," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–28.

[33] X. Liu, F. Li, B. Wen, and Q. Li, "Removing backdoor-based watermarks in neural networks with limited data," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 10149–10156.

[34] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Neural attention distillation: Erasing backdoor triggers from deep neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–19.

[35] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *Proc. Int. Symp. Res. Att. Intru. Def. (RAID)*, 2018, pp. 273–294.

[36] X. Qiao, Y. Yang, and H. Li, "Defending neural backdoors via generative distribution modeling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 14004–14013.

[37] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–13.

[38] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 87.1–87.12.

[39] J. Gao, J. Wang, S. Dai, L.-J. Li, and R. Nevatia, "NOTE-RCNN: Noise tolerant ensemble RCNN for semi-supervised object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9508–9517.

[40] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–14.

[41] L. Guo, Z. Zhang, Y. Jiang, Y. Li, and Z. Zhou, "Safe deep semi-supervised learning for unseen-class unlabeled data," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 3897–3906.

Zhicong Yan (Student Member, IEEE) received the B.S. degree in electronic information engineering from Wuhan University, Hubei, China, in 2017. He is currently pursuing the Ph.D. degree with the Department of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. He has published several high-quality papers on top-level international conferences (e.g., AAAI 2021). His research interests include weakly-supervised learning and adversarial learning.

Jun Wu (Member, IEEE) received the Ph.D. degree in information and telecommunication studies from Waseda University, Japan, in 2011. He was a Post-Doctoral Researcher with the Research Institute for Secure Systems, National Institute of Advanced Industrial Science and Technology (AIST), Japan, from 2011 to 2012. He was a Researcher with the Global Information and Telecommunication Institute, Waseda University, Japan, from 2011 to 2013. He is currently a Professor with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China. He is also the Vice Dean of the Institute of Cyber Science and Technology and the Vice Director of the National Engineering Laboratory for Information Content Analysis Technology, Shanghai Jiao Tong University. His research interests include edge intelligence, machine learning, the intelligent Internet of Things, and their security. He has hosted and participated in a lot of research projects, including the National Natural Science Foundation of China (NFSC), the National 863 Plan and 973 Plan of China, and Japan Society of the Promotion of Science Projects (JSPS). He has been the Track Chair of VTC 2019 and VTC 2020, and a TPC member of more than ten international conferences. He is the Chair of IEEE P21451-1-5 Standard Working Group. He has been a Guest Editor of IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE SENSORS JOURNAL, and *Sensors*. He is an Associate Editor of the IEEE ACCESS and IEEE NETWORKING LETTERS.

Gaolei Li (Member, IEEE) received the B.S. degree from Sichuan University, Chengdu, China, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China. From 2018 to 2019, he visited Muroran Institute of Technology, Muroran, Japan, granted by the China Scholarship Council Program. He is currently an Assistant Professor with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. His research interests include adversarial machine learning and privacy protection. He has received best paper awards from the IEEE ComSoc CSIM Committee, Chinese Association for Cryptologic Research (CACR), and IEEE GLOBECOM Student Travel Grant Award. He is a Reviewer of IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, and a TPC Member of AAAI, IEEE GLOBECOM, and IEEE ICC.

Shenghong Li (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Jilin University of Technology, Changchun, China, in 1993 and 1996, respectively, and the Ph.D. degree in radio engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1999. Since 1999, he has been with Shanghai Jiao Tong University, Shanghai, China, as a Research Fellow, an Associate Professor, and a Professor, successively. In 2010, he was a Visiting Scholar with Nanyang Technological University, Singapore. He has published over 80 articles, coauthored four books, and holds ten granted patents. His current research interests include information security, signal and information processing, and artificial intelligence. He was a recipient of the First Prize of Shanghai Science and Technology Progress in China in 2013. He was elected for New Century Talent of Chinese Education Ministry and Shanghai Dawn Scholar in 2006 and 2007.

Mohsen Guizani (Fellow, IEEE) received the B.S. (Hons.) and M.S. degrees in electrical engineering and the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He is currently a Professor with the Department of Computer Science and Engineering, Qatar University, Qatar. Previously, he served in different academic and administrative positions at the University of Idaho, Western Michigan University, the University of West Florida, the University of Missouri-Kansas City, the University of Colorado Boulder, and Syracuse University. He has authored nine books and more than 500 publications. He has guest edited a number of special issues in IEEE journals and magazines. His research interests include wireless communications and mobile computing, computer networks, and cyber security. He is currently the Editor-in Chief of the *IEEE Network magazine,* serves on the editorial boards of several international technical journals and the Founder and Editor-in-Chief of *Wireless Communications and Mobile Computing* journal (Wiley). He also served as a member, chair, and general chair for many international conferences. Throughout his career, he received three teaching awards and four research awards. He also received the 2017 IEEE Communications Society WTC Recognition Award as well as the 2018 AdHoc Technical Committee Recognition Award for his contribution to outstanding research in wireless communications and Ad-Hoc Sensor networks. He was the Chair of the IEEE ComSoc WTC and the Chair of the TAOS TC. He served as the IEEE Computer Society Distinguished Speaker. He is currently a Senior Member of ACM and Distinguished Lecturer of IEEE ComSoc.