Backdoor Learning: A Survey

Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shu-Tao Xia

***Abstract*—Backdoor attack intends to embed hidden backdoor into deep neural networks (DNNs), such that the attacked model performs well on benign samples, whereas its prediction will be maliciously changed if the hidden backdoor is activated by the attacker-defined trigger. This threat could happen when the training process is not fully controlled, such as training on third-party datasets or adopting third-party models, which poses a new and realistic threat. Although backdoor learning is an emerging and rapidly growing research area, its systematic review, however, remains blank. In this paper, we present the first comprehensive survey of this realm. We summarize and categorize existing backdoor attacks and defenses based on their characteristics, and provide a unified framework for analyzing poisoning-based backdoor attacks. Besides, we also analyze the relation between backdoor attacks and relevant fields ($i.e.$, adversarial attacks and data poisoning), and summarize widely adopted benchmark datasets. Finally, we briefly outline certain future research directions relying upon reviewed works. A curated list of backdoor-related resources is also available at https://github.com/THUYimingLi/backdoor-learning-resources.**

***Index Terms*—Backdoor Attack, Backdoor Defense, AI Security, Deep Learning.**

I. Introduction

Over the past decade, deep neural networks (DNNs) have been successfully applied in many mission-critical tasks, such as face recognition, autonomous driving, etc. Accordingly, its security is of great significance and has attracted extensive concerns. One well-studied example is adversarial examples [1], [2], [3], [4], [5], [6], which explores the adversarial vulnerability of DNNs at the inference stage. Compared to the inference stage, the training stage of DNNs involves more steps, including data collection, data pre-processing, model selection and construction, training, model saving, model deployment, etc. More steps mean more chances for the attacker, $i.e.$, more security threats to DNNs. Meanwhile, it is well known that the powerful capability of DNNs significantly depends on the huge amount of training data and computing resource. To reduce the training cost, users may choose to adopt third-party datasets, rather than to collect the training data by themselves, since there are many freely available datasets in the Internet; users may also train DNNs based on third-party platforms ($e.g.$, cloud computing platforms), rather than to train DNNs locally; users may even directly utilize third-party models. The cost of convenience is the loss of the control or the right to know to the training stage, which may further enlarge the security risk of training DNNs. One typical threat to the training stage is the *backdoor attacks*[1], which is the main focus of this survey. Different from the adversarial attack whose vulnerability results from the differences in behaviors of models and humans, backdoor attackers utilize the 'excessive' learning ability towards non-robust features (such as textures) of DNNs. More comparisons between backdoor attack and related fields are in Section V.

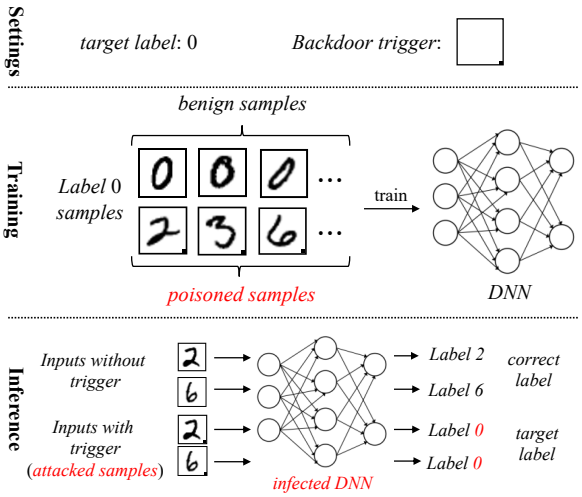

Fig. 1. An illustration of poisoning-based backdoor attacks. In this example, the trigger is a black square on the bottom right corner and the target label is '0'. Part of the benign training image is modified to have the trigger stamped during the training process, and their label is re-assigned as the attacker-specified target label. Accordingly, the trained DNN is infected, which will recognize attacked images ($i.e.$, test images containing backdoor trigger) as the target label while still correctly predict the label for the benign test images.

Gu et al. [7] firstly revealed the threat of backdoor attacks. In general, backdoor attacks aim at embedding the hidden backdoor into DNNs so that the infected model performs well on benign testing samples when the backdoor is not activated, similarly to the model trained under benign settings; however,

Manuscript received xxx, xxx; revised xxx, xxx.

Corresponding author(s): Baoyuan Wu (wubaoyuan@cuhk.edu.cn) and Shu-Tao Xia (xiast@sz.tsinghua.edu.cn).

Yiming Li is with Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China (email: li-ym18@mails.tsinghua.edu.cn).

Baoyuan Wu is with School of Data Science, The Chinese University of Hong Kong, Shenzhen, China and also with Secure Computing Lab of Big Data, Shenzhen Research Institute of Big Data, Shenzhen, China (email: wubaoyuan@cuhk.edu.cn).

Yong Jiang is with Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China and also with PCL Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen, China (email: jiangy@sz.tsinghua.edu.cn).

Zhifeng Li is with Tencent AI Lab, Shenzhen, China (email: michaelzfli@tencent.com).

Shu-Tao Xia is with Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China and also with PCL Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen, China (email: xiast@sz.tsinghua.edu.cn).

[1]In this survey, *backdoor attack* refers to the targeted attack towards the training process of DNNs. *Backdoor* is also commonly called the *neural trojan* or *trojan*. We use 'backdoor' instead of other terms in this survey since it is most frequently used.

TABLE I
Three classical scenarios and correspondingly attackers and defenders capacities. From top to bottom, the attacker's capacities gradually increase, while the defender's ones gradually decrease.

Roles →	Attackers				Defenders			
Scenario ↓, Capacity →	Training Set	Training Schedule	Model	Inference Pipeline	Training Set	Training Schedule	Model	Inference Pipeline
Adopt Third-Party Datasets	●	○	○	○	●	●	●	●
Adopt Third-Party Platforms	●	●	○	○	○	○	●	●
Adopt Third-Party Models	●	●	●	○	○	○	◐	●

[1] ●: controllable; ○: uncontrollable; ◐: partly controllable (It is uncontrollable for defenders when using the third-party model's API, while it is somehow controllable when adopting pre-trained models).

if the backdoor is activated by the attacker, then its prediction will be changed to the attacker-specified target label. Since the infected DNNs perform normally under benign settings and the backdoor is activated (only) by the attacker-specified trigger, it is difficult for users to realize its existence. Accordingly, the insidious backdoor attack is a serious threat to DNNs. Currently, training data poisoning [7], [8], [9] is the most straightforward and common way to encode backdoor functionality into the model's weights during the training process. As demonstrated in Fig. 1, some training samples are modified by adding an attacker-specified trigger ($e.g.$, a local patch). These modified samples with attacker-specified target label and benign training samples are fed into DNNs for training. Besides, backdoor triggers could be invisible [10], [11], [12] and the ground-truth label of poisoned samples could also consistent with the target label [13], [14], [15], which increases the stealthiness of backdoor attacks. Except by directly poisoning the training samples, the hidden backdoor could also be embedded through transfer learning [7], [16], [17], directly modifying model's weights [18], [19], and introducing extra malicious module [20]. In other words, backdoor attacks could happen at all steps in the training process.

To alleviate the backdoor threat, different defense methods were proposed. In general, those methods can be divided into two main categories, including *empirical backdoor defenses* and *certified backdoor defenses*. Empirical backdoor defenses [21], [22], [23] are proposed based on some observations or understandings of existing attacks and have decent performance in practice; however, their effectiveness have no theoretical guarantee and may probably be bypassed by some adaptive attacks. In contrast, the validity of certified backdoor defenses [24], [25] is theoretically guaranteed under certain assumptions, whereas it is generally weaker than that of empirical defenses in practice. How to better defend backdoor attacks is still an important open question.

As we mentioned, backdoor attacks are a realistic threat towards the applications of DNNs and their defenses are also of great significance. However, there is still no comprehensive review of both aspects and no framework about how to analyze different works in a systematic way. In this paper, we provide a timely overview of the current status, a general framework for analyzing poisoning-based backdoor attacks, and some insights about future research directions of backdoor learning. We believe this survey will facilitate continuing research in this emerging research area.

The rest of this paper is organized as follows. Section II briefly describes common technical terms and threat scenarios. Section III-IV provides an overview of existing backdoor attacks. Section V analyzes the relation between backdoor attacks and related realms, while Section VI demonstrates and categorizes existing defenses. Section VII illustrates existing benchmark datasets, and Section VIII discusses remaining challenges and suggests future directions. The conclusion is provided in Section IX at the end.

II. Preliminaries

A. Definition of Technical Terms

In this section, we briefly describe and explain common technical terms used in the backdoor learning relevant literature. We will follow the same definition of terms in the remaining paper.

- *Benign model* refers to the model trained under benign settings.
- *Infected model* refers to the model with hidden backdoor(s).
- *Poisoned sample* is the modified training sample used in poisoning-based backdoor attacks for embedding backdoor(s) in the model during the training process.
- *Trigger* is the pattern used for generating poisoned samples and activating the hidden backdoor(s).
- *Attacked sample* indicates the malicious testing sample (with trigger) used for querying the infected model.
- *Attack scenario* refers to the scenario that the backdoor attack might happen. Usually, it happens when the training process is inaccessible or out of control by the user, such as training with third-party datasets, training through third-party platforms, or adopting third-party models.
- *Source label* indicates the ground-truth label of a poisoned or an attacked sample.
- *Target label* is the attacker-specified label. The attacker intends to make all attacked samples to be predicted as the target label by the infected model.
- *Attack success rate (ASR)* denotes the proportion of attacked samples which are predicted as the target label by the infected model.
- *Benign accuracy (BA)* indicates the accuracy of benign test samples predicted by the infected model.
- *Attacker's goal* describe what the backdoor attacker intends to do. In general, the attacker wishes to design an infected model that performs well on the benign testing sample while achieving high ASR.
- *Capacity* defines what the attacker/defender can and cannot do to achieve their goal.
- *Attack/Defense approach* illustrates the process of the designed backdoor attack/defense.

B. Classical Scenarios and Corresponding Capacities

In this section, we introduce three classical real-world scenarios that backdoor threat could occur, and their corresponding attackers and defenders capacities. More details are summarize in Table I and illustrated as follows:

Scenario 1: Adopt Third-Party Datasets. In this scenario, attackers provide the poisoned dataset to users directly or through the Internet. Users will adopt the (poisoned) dataset to train and deploy their models. Accordingly, the attacker can only manipulate the dataset, whereas cannot modify the model, the training schedule, and the inference pipeline. In contrast, defenders can manipulate everything in this scenario. For example, they can clean up the (poisoned) dataset to alleviate the backdoor threat.

Scenario 2: Adopt Third-Party Platforms. In this scenario, users provide their (benign) dataset, model structure, and training schedule to an untrusted third-party platform ($e.g.$, Google Cloud) to train their model. Although the benign dataset and training schedule is provided, the attacker ($i.e.$, the malicious platform) can modify them during the actual training process. However, the attacker cannot change the model structure otherwise users will notice the attack. On contrary, defenders can not control the training set and schedule while can modify the trained model to alleviate the attack. For example, they can fine-tune it on a small local benign dataset.

Scenario 3: Adopt Third-Party Models. In this scenario, attackers provide trained infected DNNs through the application programming interface (API) or the Internet. Attackers can change everything except for the inference pipeline. For example, the user can introduce a pre-processing module on the test image before the prediction, which is out of control by the attackers. For the defenders, they can control the inference pipeline and also the model when its source files are provided; however, if they can only get access to the model API, they can not modify the model.

III. Poisoning-based Backdoor Attacks

In the past three years, many backdoor attacks were proposed. In this section, we first propose a unified framework to analyze existing poisoning-based attacks towards image classification, based on the understanding of attack properties. After that, we summarize and categorize existing poisoning-based attacks in detail based on the proposed framework. Attacks for other tasks or paradigms and the well-intentioned applications of backdoor attacks are also discussed at the end.

A. A Unified Framework of Poisoning-based Attacks

We first define three necessary risks in this area, then describe the optimization process of poisoning-based backdoor attacks. Based on the characteristic of the process, poisoning-based attacks can be categorized based on different criteria, as shown in Fig. 2. More details about categorization criteria are summarized in Table II.

We denote the classifier as $f_{\boldsymbol{w}} : \mathcal{X} \to [0,1]^{|\mathcal{Y}|}$, where $\boldsymbol{w}$ is the model parameter, $\mathcal{X} \subset \mathbb{R}^d$ being the instance space, and $\mathcal{Y} = \{1, 2, \cdots, K\}$ being the label space. $f(\boldsymbol{x})$ indicates the posterior vector with respect to K classes, and $C(\boldsymbol{x}) = \arg\max f_{\boldsymbol{w}}(\boldsymbol{x})$ denotes the predicted label. Let y_t denotes the target label, $\mathcal{D}_L = \{(\boldsymbol{x}_i, y_i) | i = 1, \ldots, N_l\}$ indicates the labeled dataset, and $\mathcal{D}'_L = \{\boldsymbol{x} | (\boldsymbol{x}, y) \in \mathcal{D}_L\}$ indicates the instance set of $\mathcal{D}_L$. Three risks involved in existing attacks are defined as follows:

Definition 1 (Standard, Backdoor, and Perceivable Risk)**.**

- *The standard risk R_s measures whether the prediction of $\boldsymbol{x}$ ($i.e.$, $C(\boldsymbol{x})$), is same with its ground-truth label y. Its definition with respect to a labeled dataset $\mathcal{D}_L$ is formulated as*

$$R_s(\mathcal{D}_L) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}_{\mathcal{D}_L}} \left[\mathbb{I}\{C(\boldsymbol{x}) \neq y\}\right], \tag{1}$$

  where $\mathcal{P}_{\mathcal{D}_L}$ indicates the distribution behind $\mathcal{D}_L$. $\mathbb{I}(a)$ denotes the indicator function: $\mathbb{I}(a) = 1$ if a is true, otherwise $\mathbb{I}(a) = 0$.
- *The backdoor risk R_b indicates whether the backdoor trigger $\boldsymbol{t}$ can successfully activates the hidden backdoor within the classifier. Its definition with respect to $\mathcal{D}_L$ is formulated as*

$$R_b(\mathcal{D}_L) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}_{\mathcal{D}_L}} \left[\mathbb{I}\{C(\boldsymbol{x}') \neq y_t\}\right], \tag{2}$$

  where $\boldsymbol{x}' = G(\boldsymbol{x}, \boldsymbol{t})$ is the poisoned version of benign sample $\boldsymbol{x}$ under generation function $G(\cdot)$ with trigger $\boldsymbol{t}$. For example, $G(\boldsymbol{x}, \boldsymbol{t}) = (\boldsymbol{1} - \boldsymbol{\alpha}) \cdot \boldsymbol{x} + \boldsymbol{\alpha} \cdot \boldsymbol{t}$ is the most commonly adopted generation function, where $\boldsymbol{\alpha} \in [0,1]^d$ and y_t indicate the blended parameter and target label, respectively.
- *The perceivable risk R_p denotes whether the poisoned sample ($i.e.$, $\boldsymbol{x}'$) can be detected as the malicious sample (by human or machine). Its definition with respect to $\mathcal{D}_L$ is formulated as*

$$R_p(\mathcal{D}_L) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}_{\mathcal{D}_L}} \left[D(\boldsymbol{x}')\right], \tag{3}$$

  where $D(\cdot)$ is an indicator function: $D(\boldsymbol{x}') = 1$ if $\boldsymbol{x}'$ is detected as the malicious sample, otherwise $D(\boldsymbol{x}') = 0$.

Based on aforementioned definition, existing attacks can be summarized in a unified framework, as follows:

$$\min_{\boldsymbol{t},\boldsymbol{w}} R_s(\mathcal{D}_L - \mathcal{D}_{sL}) + \lambda_1 \cdot R_b(\mathcal{D}_{sL}) + \lambda_2 \cdot R_p(\mathcal{D}_{sL}), \tag{4}$$

where $\boldsymbol{t} \in \mathcal{T}$, λ_1 and λ_2 are two non-negative trade-off hyper-parameters, $\mathcal{D}_{sL}$ is a subset of $\mathcal{D}_L$, and $\frac{|\mathcal{D}_{sL}|}{|\mathcal{D}_L|}$ is called *poisoning rate* defined in existing works [7], [26], [9].

Remark. Since the indicator function $\mathbb{I}(\cdot)$ used in R_s and R_b is non-differentiable, it is usually replaced by its surrogate loss ($e.g.$, cross-entropy, KL-divergence) in practice. Besides, as we mentioned, optimization (4) can reduce to existing attacks through different specifications. For example, when $\lambda_1 = \frac{|\mathcal{D}_{sL}|}{|\mathcal{D}_L - \mathcal{D}_{sL}|}$, $\lambda_2 = 0$, and $\boldsymbol{t}$ is non-optimized ($i.e.$, $|\mathcal{T}| = 1$), it reduces to the BadNets [7] and the Blended Attack [26]; when $\lambda_2 = +\infty$ and $D(\boldsymbol{x}'; \boldsymbol{x}) = ||\boldsymbol{x}' - \boldsymbol{x}||_p$, it reduces to ℓ^p-ball bounded invisible backdoor attacks [11]. Moreover, parameters $\boldsymbol{t}$ and $\boldsymbol{w}$ could be optimized simultaneously or separately through a multi-stage method.

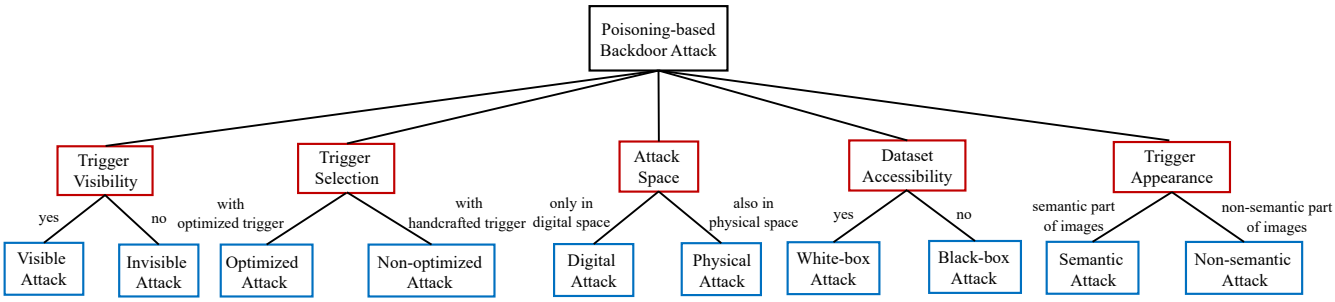

Fig. 2. Taxonomy of poisoning-based backdoor attacks with different categorization criteria. In this figure, the red boxes represent categorization criteria, while the blue boxes indicates attack subtypes.

TABLE II
Summary of existing poisoning-based backdoor attacks.

$\min_{\boldsymbol{t},\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}_{\mathcal{D}_L-\mathcal{D}_{sL}}}[\mathbb{I}\{C(\boldsymbol{x})\neq y\}] + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}_{\mathcal{D}_{sL}}}[\lambda_1\cdot\mathbb{I}\{C(\boldsymbol{x}')\neq y_t\} + \lambda_2\cdot D(\boldsymbol{x}')]$, where $\boldsymbol{t}\in\mathcal{T}$, $\boldsymbol{x}'=G(\boldsymbol{x},\boldsymbol{t})$.				
Visible Attack	$D(\boldsymbol{x}')=1$.	Invisible Attack	Clean-label Poison-label	$D(\boldsymbol{x}')=0$, and $y_t=y$. $D(\boldsymbol{x}')=0$, and $y_t\neq y$.
Optimized Attack	$\lvert\mathcal{T}\rvert>1$.	Non-optimized Attack		$\lvert\mathcal{T}\rvert=1$.
Digital Attack	$\boldsymbol{x}'$ is generated in digital space.	Physical Attack		Physical space is involved in generating $\boldsymbol{x}'$.
White-box Attack	$\mathcal{D}_L$ is known.	Black-box Attack		$\mathcal{D}_L$ is unknown.
Semantic Attack	$\boldsymbol{t}$ is a semantic part of samples.	Non-semantic Attack		$\boldsymbol{t}$ is not a semantic part of samples.

Note that this framework can be easily generalized towards other tasks, such as speech recognition, as well.

B. Evaluation Metrics

To evaluate the performance of backdoor attacks in the image classification, two classical metrics are usually adopted, including **(1)** benign accuracy (BA) and **(2)** attack success rate (ASR), as defined in Section II-A. The higher the ASR and the closer the BA between the infected model and the benign model, the better the attack performance. Besides, the smaller the poisoning rate ($i.e.$, rate of poisoned samples over all training samples) and the perturbation between the benign image and the poisoned image, the more stealthy the attack and therefore the better the attack.

C. Attacks for Image and Video Recognition

1) BadNets: Embedding hidden backdoors in a model typically involves the encoding of malicious functionalities within model parameters. Gu et al. [7] first introduced the backdoor attack in deep learning and proposed a method, dubbed BadNets, to inject backdoor by poisoning some training samples. Specifically, as demonstrated in Fig. 1, its training process consists of two main parts, including **(1)** generating the poisoned image $\boldsymbol{x}'$ by stamping the backdoor trigger onto the benign image $\boldsymbol{x}$ to achieve poisoned sample $(\boldsymbol{x}', y_t)$ associated with the attacker-specified target label y_t, and **(2)** training the model with poisoned samples as well as benign samples. Accordingly, the trained DNN will be infected, which performs well on benign testing samples, similarly to the model trained using only benign samples; however, if the same trigger is contained in an attacked image, then its prediction will be changed to the target label. BadNets could happen in every scenario described in Section II-B, which is a serious security threat. BadNets is the representative of *visible attacks*, which opened the era of this field. Almost all follow-up poisoning-based attacks were carried out based on this method.

2) Invisible Backdoor Attacks: Chen et al. [26] first discussed the *invisibility requirement* of poisoning-based backdoor attacks. They suggested that the poisoned image should be indistinguishable compared with its benign version to evade human inspection. To fulfill this requirement, they proposed a *blended strategy*, which generated poisoned images by blending the backdoor trigger with benign images instead of by stamping (as proposed in BadNets [7]). Besides, they demonstrated that even adopting a random noise with a small magnitude as the backdoor trigger can still create the backdoor successfully, which further reduces the risk of being detected.

After that, there was a series of works dedicated to the research of *invisible backdoor attacks*. In [10], Turner et al. proposed to perturb the pixel values of benign images by a backdoor trigger amplitude instead of by replacing the corresponding pixels with the chosen pattern. Zhong et al. [12] adopted the universal adversarial attack [27] to generate the backdoor trigger, which minimizes the ℓ^2 norm of the perturbation. Li et al. [11] proposed to regularize the ℓ^p norm of the perturbation when optimizing the backdoor trigger. In [28], Bagdasaryan et al. viewed the backdoor attack as a special multi-task optimization, where they fulfilled the invisibility through poisoning the loss computation. Moreover, Liu et al. [8] proposed to adopt a common phenomenon, the reflection, as the trigger for the stealthiness. Cheng et al. [29] proposed to conduct the invisible attack in the feature space in a style transfer way. Most recently, Li et al. [30] adopted DNN-based image steganography to generate invisible triggers for the backdoor attack. Compared with previous methods, this attack is not only invisible but can also bypass most of defenses, since generated triggers are sample-specific.

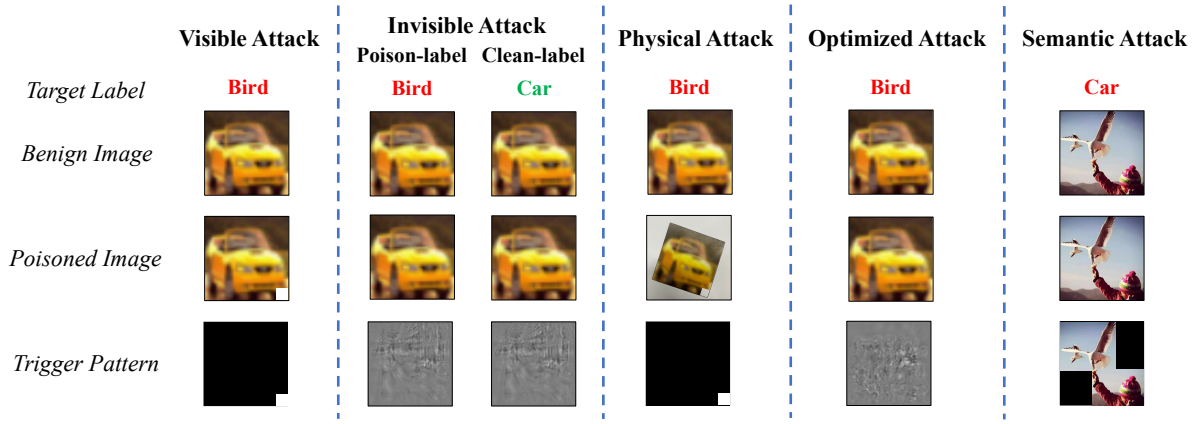

Fig. 3. An example of poisoned samples generated by different types of backdoor attacks. **(1)** In the visible attack, the backdoor trigger is a white-square stamped on the bottom right corner of the poisoned image, which is visible. **(2)** In the visible attack, the trigger is a noise with small perturbations, which is invisible. Moreover, the target label of the poisoned image is different from the ground-truth label of its benign version in the poison-label attack, whereas these labels are the same in the clean-label attack. **(3)** In the physical attack, the (digital) poisoned image is captured by the camera from the physical space. **(4)** In the optimized attack, the trigger is a universal perturbation towards the target class instead of a random selection one. **(5)** The poisoned image is exactly the same as its benign version in the semantic attack. In this case, the trigger is the combination of two semantic objects ($i.e.$, 'bird' and 'human'). Images containing these objects simultaneously will be classified by the infected models as the 'car'.

Although a poisoned image is similar to its benign version in invisible attacks, however, its source label is usually different from the target label. In other words, all those methods are *poison-label invisible attacks*, where the poisoned samples seem to be mislabeled. Accordingly, an invisible attack still could be detected by humans by examining the image-label relationship of training samples. To address this problem, a special sub-class of invisible poisoning-based attacks, dubbed *clean-label invisible attacks*, was proposed, which has more serious threats and research value. Turner et al. [10] first explored the clean-label attack, where they leveraged adversarial perturbations or generative models to first modify some benign images from the target class and then conducted the standard invisible attack. The modification approach is to alleviate the effects of 'robust features' contained in the poisoned samples to ensure that the trigger can be successfully learned by the DNNs. Recently, Zhao et al. [14] extended this idea in attacking video classification, where they adopted universal perturbation instead of a given one as the trigger pattern. Another interesting clean-label attack method is to inject the information of a poisoned sample generated by a previous visible attack into the texture of an image from target class by minimizing their distance in the feature space, as suggested in [13]. Recently, Quiring et al. [15] proposed to conceal the trigger as well as hide the overlays of clean-label poisoning through *image-scaling attacks* [31].

Note that clean-label attacks usually suffered from significantly lower attack success rate (ASR) and even fail to succeed, although they are more stealthy compared with poison-label ones. How to balance the stealthiness and effectiveness of attacks is still an open question and worth further exploration.

3) Optimized Attacks: Triggers are the core of poisoning-based attacks, therefore analyzing how to design a better trigger instead of simply using a given non-optimized trigger pattern is of great significance and has attracted some concerns. Optimized attacks generated poisoned samples with optimized triggers to achieve better attack performance. To the best of our knowledge, Liu et al. [32] first explored this problem, where they proposed to optimize the trigger so that the important neurons can achieve the maximum values. In [11], Li et al. formulated the trigger generation as a bilevel optimization, where the trigger was optimized to amplify a set of neuron activations with ℓ^p regularization for invisibility. Bagdasaryan et al. [28] treated backdoor attacks as a multi-objective optimization and proposed to optimize trigger and train DNNs simultaneously. Recently, with the hypothesis that if a perturbation can induce most samples toward the decision boundary of the target class then it will serve as an effective trigger, [12], [14], [33] proposed to generate trigger through universal adversarial perturbation. Although these design methods have achieved certain success, most of them are still heuristic. How to design triggers in a more optimized way is still an important open question.

4) Physical Backdoor Attacks: Different from previous *digital attacks* where the attack was conducted completely in the digital space, the physical space is involved when generating poisoned samples in the *physical attacks*. Chen et al. [26] first explored the landscape of this attack. In [26], they adopted a pair of glasses as the physical trigger to mislead the infected face recognition system developed in a camera. Further exploration of attacking face recognition in the physical world was also discussed by Wenger et al. [34]. A similar idea was also discussed in [7], where a post-it note was adopted as the trigger in attacking traffic sign recognition deployed in the camera. Recently, Li et al. [9] demonstrated that existing digital attacks fail in the physical world since the involved transformations ($e.g.$, rotation, and shrinkage) change the location and appearance of trigger in attacked samples compared with the one used for training. This inconsistency will greatly reduce the performance of the attack. Based on this understanding, they proposed a transformation-based attack enhancement so that the enhanced attacks remain effective in the physical world. This attempt is an important step towards successful backdoor attacks in real-world applications.

5) Black-box Backdoor Attacks: Different from previous *white-box attacks*, which require the knowledge of training samples, *black-box attacks* adopt the settings that the training set is inaccessible. In practice, the training dataset is usually not shared due to privacy or copyright concerns, therefore black-box attacks are more realistic than white-box ones. Specifically, black-box backdoor attacks need to generate some training samples at first. For example, in [32], they generated some representative images of each class by optimizing images initialized from another dataset such that the prediction confidence of the selected class reaches maximum. With the reversed training set, white-box attacks can be adopted for backdoor injection. Black-box backdoor attacks are significantly difficult than white-box ones. Currently, there are only a few works focusing on this area.

6) Semantic Backdoor Attacks: The majority of backdoor attacks, $i.e.$, the *non-semantic attacks*, assume that the trigger is independent of benign images. In other words, attackers need to modify the image in the inference stage to activate the hidden backdoor. Is it possible that a semantic part of samples can also serve as the trigger, such that the attacker is not required to modify the input at inference time to deceive the infected model? Bagdasaryan et al. first explored this problem and proposed a novel type of backdoor attacks [35], [28], $i.e.$, the *semantic backdoor attacks*. Specifically, they demonstrated that assigning an attacker-chosen label to all images with certain features, $e.g.$, green cars or cars with racing stripes, for training can create a semantic hidden backdoor in infected DNNs. Accordingly, the infected model will automatically misclassify testing images, which contain pre-defined semantic information, without any image modification. A similar idea was also explored in [36], where the hidden backdoor can be activated by the combination of certain objects in the image. Since this type of attacks does not require modifying images in the inference process in the digital space, we believe it is very malicious and worth further exploration.

D. Attacks against Other Fields or Paradigms

Currently, most existing backdoor attacks against other tasks or paradigms were still poisoning-based. Accordingly, except for task-specific requirements, most methods focused on **(1)** how to design the trigger, **(2)** how to define the attack stealthiness, and **(3)** how to bypass potential defenses. The huge differences between different tasks and paradigms make the answers to the above questions completely different. For example, the stealthiness in image-related tasks can be defined as the pixel-wise distance ($e.g.$, ℓ^p norm) between the poisoned sample and its benign version; however, in natural language processing (NLP), changing even a word or character may still make the modification visible to human since it may cause grammar or spelling errors. As such, simply the difference between the poisoned sample and its benign version may not serve as a good stealthiness metric in NLP-related tasks.

Natural language processing is the most extensive research field in backdoor attacks besides image or video classification. In [37], Dai et al. discussed how to attack against LSTM-based sentiment analysis. Specifically, they proposed a BadNets-like approach, where an emotionally neutral sentence was used as the trigger and was randomly inserted into some benign training samples. In [38], Chen et al. further explored this problem, where three different types of triggers ($i.e.$, char-level, word-level, and sentence-level triggers) were proposed and reached decent performance. Besides, Kurita et al. [16] demonstrated that sentiment classification, toxicity detection, and spam detection can also be attacked even after fine-tuning. Most recently, Chan et al. [39] proposed to attack NLP models in the latent space based on the conditional adversarially regularized auto-encoder. Except for NLP-related tasks, some researches also revealed the backdoor threat towards graph neural networks (GNN) [40], [41]. In general, an attacker-specified sub-graph was defined as the trigger so that the infected GNN will predict the target label for an attacked graph once the trigger is contained in attacked samples. Besides, the backdoor threat towards other tasks, such as reinforcement learning [42], [43], [44], speaker verification [45], and wireless signal classification [46], were also studied.

Except for the classical training paradigm, how to backdoor collaborative learning, especially federated learning, have attracted the most attention. In [35], Bagdasaryan et al. introduced the first backdoor attack against federated learning by amplifying the poisoned gradient of node servers. After that, Bhagoji et al. [47] discussed the stealthy model-poisoning backdoor attack, and Xie et al. [48] introduced a distributed backdoor attacks against the federated learning. Most recently, [49] theoretically verified that backdoor attacks are unavoidable if a model is vulnerable to adversarial examples under mild conditions in federated learning. Besides, the backdoor attacks towards meta federated learning [50] and feature-partitioned collaborative learning [51] were also discussed. In contrast, some works [52], [53], [54], [55], [56], [57] also questioned whether federal learning is really easy to be attacked. Except for collaborative learning, the backdoor threat of another important learning paradigm, the transfer learning, was also discussed in [7], [58], [16], [17], [59].

E. Backdoor Attack for Good

Except for malicious purposes, how to use the backdoor attack in the right way has also obtained some preliminary explorations. Adi et al. [60] exploited backdoor attacks in verifying model ownership. They proposed to watermark the DNNs through backdoor embedding. Accordingly, the hidden backdoor in the model can be used to examine the ownership, while the watermarking process still preserves original model functionality. Besides, Sommer et al. [61] revealed how to verify whether the server truly erases their data when users require data deletion through poisoning-based backdoor attacks. Specifically, in their verification framework, each user poisons part of its data with user-specific trigger and target label. Accordingly, each user can leave a unique trace in the server for deletion verification after the server being trained on user data while having a negligible impact on the benign model functionality. Shan et al. [62] introduced a trapdoor-enabled adversarial defense, where the hidden backdoor is injected by the defender to prevent attackers from discovering the natural

TABLE III
Comparison among the backdoor attack, adversarial attack, and data poisoning.

Attack Category	Attack Target	Attack Mechanism	Training Process	Inference Process
Backdoor Attack	Misclassify attacked samples; Behave normal on benign samples.	Excessive learning ability of models.	Under control.	Out of control.
Adversarial Attack	Misclassify attacked samples; Behave normal on benign samples.	Behavior differences between models and humans.	Out of control.	Attackers need to generate adversarial perturbation through an iterative optimization process.
Data Poisoning	Reduce model generalization.	Overfitting to bad local optima.	Can only modify the training set.	Out of control.

weakness in a model. The motivation is that the adversarial perturbation generated by gradient-descent-based attacks towards an infected model will converge near the trapdoor pattern, which is easily detected by the defender. Moreover, Li et al. [63] discussed how to protect open-sourced datasets based on backdoor attacks. Specifically, they formulated this problem as determining whether the dataset has been adopted to train a third-party model. Specifically, they proposed a hypothesis test based method for the verification, based on the posterior probability generated by the suspicious third-party model of the benign samples and their correspondingly attacked samples. Most recently, backdoor attacks were also adopted for DNNs interpretability [64] and the evaluation of explainable AI methods [65].

IV. Non-poisoning-based Backdoor Attacks

Except for poisoning-based attacks, some non-poisoning-based attacks were also proposed. These methods inject backdoor not directly through optimizing model parameters during the training process with poisoned samples. Their existence demonstrates that except for happening at the data collection, the backdoor attack could also happen at other stages ($e.g.$, deployment stage) of the training process, which further reveals the severity of the backdoor attack.

A. Targeted Weight Perturbation

In [18], Dumford et al. first explored the non-poisoning-based attack, where they proposed to modify the model's parameters directly instead of through training with poisoned samples. The primary task in this work is face recognition, where they assumed that the training samples can not be modified by attackers. The attacker's goal is to make their own face to be granted access despite not being a valid user while ensuring that the network still behaves normally for all other inputs. To fulfill this target, they adopted a greedy search across models with different perturbations applied to a pre-trained model's weights.

B. Targeted Bit Trojan

Instead of modifying the model's parameters simply through a search-based approach, Rakin et al. [19] demonstrated a new method, dubbed targeted bit trojan (TBT), discussing how to inject a hidden backdoor without the training process. TBT contains two main processes, including gradient-based vulnerable bits determination (similar to the process proposed in [32]), and targeted bits flipping in main memory by adopting *row-hammer attack* [66]. The proposed method achieved remarkable performance, the authors were able to mislead ResNet-18 [67] on the CIFAR-10 dataset [68] with 84 bit-flips out of 88 million weight bits.

C. TrojanNet

Different from previous approaches where the backdoor is embedded in the parameters directly, Guo et al. [69] proposed TrojanNet to encode the backdoor in the infected DNNs activated through a secret weight permutation. They assumed that the infected network is used with a hidden backdoor software which could permute the parameters when the backdoor trigger is presented. Training a TrojanNet is similar to the *multi-task learning*, although the benign task and malicious task share no common features. Besides, the authors also proved that the decision problem to determine whether the model contains a permutation that triggers the hidden backdoor is NP-complete, and therefore the backdoor detection is almost impossible.

D. Attack with Trojan Module

Most recently, Tang et al. [20] proposed a novel non-poisoning-based backdoor attack, which inserted a trained malicious backdoor module ($i.e.$, a sub-DNN) into the target model instead of changing parameters in the original model to embed hidden backdoors. Since the trigger is only associated with the malicious module, which can be combined with any DNN, the proposed method is model-agnostic. Moreover, since the attacker only needs to train the (small) trojan module once, this method significantly reduced the computational cost compared to previous attack methods. It could serve as a strong baseline in the scenario that users adopt third-party models.

V. Connection with Related Realms

In this section, we discuss the similarities and differences between backdoor attacks and related realms. Those connections are summarized in Table III.

A. Backdoor Attacks and Adversarial Attacks

Adversarial attacks and (poisoning-based) backdoor attacks share many similarities. Firstly, both types of attacks intend to modify the benign testing sample to make the model misbehave. Although the perturbation is usually image-specified for adversarial attacks, when the adversarial attacks are with universal perturbation ($e.g.$, [27], [70], [71]), those types of attacks have a similar pattern. Accordingly, researchers who are not familiar with backdoor attacks may question the significance of the research in this area.

Although adversarial attacks and backdoor attacks enjoy certain similarities, they still have essential differences. **(1)** From the aspect of the attacker's capacity, adversarial attackers can control the inference process (to a certain extent) but not the training process of models. In contrast, for backdoor attackers, parameters of the model can be modified whereas the inference process is out of control. **(2)** From the perspective of attacked samples, the perturbation is known ($i.e.$, non-optimized) by backdoor attackers whereas adversarial attackers need to obtain it through the optimization process based on the output of the model. Such optimization in adversarial attacks requires multiple queries [106], [107], [108] and therefore may probably be detected. **(3)** Their mechanism also has essential differences. Adversarial vulnerability results from the differences in behaviors of models and humans. In contrast, backdoor attackers utilize the excessive learning ability of DNNs towards non-robust features ($e.g.$, textures).

Most recently, there were also a few early works researching the latent connection between adversarial learning and backdoor learning. For example, Pang et al. [109] revealed that there exist intriguing 'mutual reinforcement' effects between the data poisoning and adversarial attacks, which can be used to enhance backdoor attacks; Weng et al. [110] empirically demonstrated that the adversarial robustness may at odd with the backdoor robustness.

B. Backdoor Attacks and Data Poisoning

Data poisoning and (poisoning-based) backdoor attacks share many similarities in the training phase. In general, they all aim at misleading models in the inference process by introducing poisoned samples during the training process. However, they also have significant differences. From the perspective of the attacker's goal, data poisoning aims at degrading the performance in predicting benign testing samples. In contrast, backdoor attacks preserve the performance on benign samples, similarly with the benign model, while changing the prediction of attacked samples ($i.e.$, benign testing samples with trigger) to the target label. From this angle, data poisoning can be regarded as the 'non-targeted poisoning-based backdoor attack' with transparent triggers to some extent. From the aspect of stealthiness, backdoor attacks are more malicious than data poisoning. Users can detect data poisoning by the evaluation under the local verification set, while this approach has limited benefits in detecting backdoor attacks.

Note that existing data poisoning works have also inspired the research on backdoor learning due to their similarities. For example, Hong et al. [96] demonstrated that the defense towards data poisoning may also have benefits in defending backdoor attacks, as illustrated in Section VI-A5.

VI. Backdoor Defenses

To alleviate the backdoor threats, several backdoor defenses were proposed. Existing methods mostly aim at defending poisoning-based attacks and can be divided into two main categories, including *empirical backdoor defenses* and *certified backdoor defenses*. Empirical backdoor defenses are proposed based on some understandings of existing attacks and have decent performance in practice, whereas their effectiveness has no theoretical guarantee. In contrast, the validity of certified backdoor defenses is theoretically guaranteed under certain assumptions, whereas it is generally weaker than that of empirical defenses in practice. At present, certified defenses are all based on the *random smoothing* [111], while empirical ones have multiple types of approaches.

A. Empirical Backdoor Defenses

Intuitively, poisoning-based backdoor attacks are similar to unlock a door with the corresponding key. In other words, there are three indispensable requirements to ensure the success of backdoor attacks, including **(1)** having a hidden backdoor in the (infected) model, **(2)** containing triggers in (attacked) samples, and **(3)** the trigger and the backdoor are matched, as shown in Fig. 4. Accordingly, three main defense paradigms, including **(1)** trigger-backdoor mismatch, **(2)** backdoor elimination, and **(3)** trigger elimination, can be adopted to defend existing attacks. Different types of approaches were proposed towards the aforementioned paradigms, which are summarized in Table IV and will be further demonstrated as follows:

1) Preprocessing-based Defenses: Preprocessing-based defenses introduce a preprocessing module before the original inference process, which changes the pattern of the triggers in the attacked samples. Accordingly, the modified trigger no longer matches the hidden backdoor and therefore preventing backdoor activation.

Liu et al. [72] were the first to exploit the preprocessing technique as a defense approach towards image classification, where they adopted a pre-trained auto-encoder as the preprocessor. Inspired by the idea that the trigger region contributes most to the prediction, a two-stage image preprocessing approach, dubbed Februus, was proposed by Doan et al in [73]. At the first stage, Februus utilizes GradCAM [112] to identify influential regions, which will then be removed and replaced by a neutralized-color box. After that, Februus adopts a GAN-based inpainting method to reconstruct the masked region for alleviating the adverse effect towards benign samples. Udeshi el al. [74] proposed to utilize the dominant color in the image to make a square-like trigger blocker in the preprocessing stage, which was adopted to locate and remove the backdoor trigger. This approach was motivated by the fact that placing a trigger blocker at the position of the backdoor trigger in the attacked image will result in a change in the prediction of the model. Vasquez et al. [75] proposed to preprocess the image through style transfer. Recently, Li et al. [9] discussed the property of existing poisoning-based attacks with static trigger pattern. They demonstrated that if the *appearance* or *location* of the trigger is slightly changed, then the attack performance may degrade sharply. Based on this observation, they proposed to adopt spatial transformations ($e.g.$, shrinking, flipping) for the defense. Compared with previous methods, this method is more efficient since it requires almost no additional computational costs. A similar idea was also explored in [76], where they evaluated and introduced different transformations in both the training and inference process.

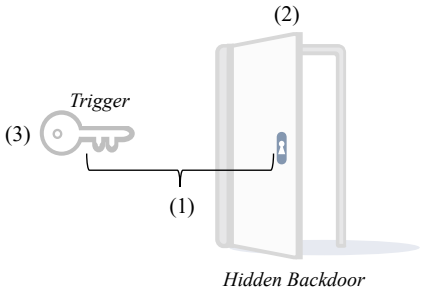

Fig. 4. An illustration of backdoor attacks and three corresponding defense paradigms. Intuitively, the poisoning-based backdoor attack is similar to unlock a door with the corresponding key. Accordingly, three main paradigms, including **(1)** trigger-backdoor mismatch, **(2)** backdoor elimination, and **(3)** trigger elimination, can be adopted to defend the attack. Different types of approaches were proposed towards the aforementioned paradigms, as illustrated in Table IV.

TABLE IV
Summary of existing empirical backdoor defenses in image recognition. (Some literature proposed different types of defenses simultaneously, therefore they will appear multiple times in this table.)

Defense Paradigm	Defense Sub-category	Literarure
Trigger-backdoor Mismatch	Preprocessing-based Defense	[72], [73], [74], [75] [9], [76]
Backdoor Elimination	Model Reconstruction based Defense	[72], [21], [77], [78] [79]
	Trigger Synthesis based Defense	[80], [81], [82], [83] [84], [85], [86], [87] [88], [89], [90]
	Model Diagnosis based Defense	[91], [92], [23], [93] [94]
	Poison Suppression based Defense	[95], [96]
	Training Sample Filteing based Defense	[97], [98], [99], [100] [101], [102]
Trigger Elimination	Testing Sample Filtering based Defense	[22], [103], [95], [104] [105]

2) Model Reconstruction based Defenses: Different from preprocessing based defenses, model reconstruction based defenses aim at removing the hidden backdoor in the infected model. Accordingly, even if the trigger is still contained in the attacked samples, the prediction remains unmalicious since the backdoor was already removed.

Liu et al. [72] proposed to retrain the given model with local benign samples starting from the weight of the given model. The effectiveness of this method may probably due to the *catastrophic forgetting* in DNNs [113], $i.e.$, the hidden backdoor is gradually removed as the training goes since the retraining set contains no poisoned samples. Motivated by the observation that the backdoor related neurons are usually dormant for benign samples, Liu et al. [21] proposed to prune those neurons to remove the hidden backdoor. Besides, they proposed a fine-pruning method, which first prunes the DNNs and then fine-tunes the pruned network to combine the benefits of the pruning and fine-tuning defenses. In [77], Zhao et al. showed that the hidden backdoor of the infected DNNs can be repaired based on the *mode connectivity* technique [114] with a certain amount of benign samples. Most recently, Yoshida et al. [78] and Li et al. [79] proposed to utilize *knowledge distillation* technique [115] to reconstruct (infected) DNNs, based on the understanding that the distillation process can perturb backdoor-related model weights and therefore can remove network's attentions on the trigger pattern.

3) Trigger Synthesis based Defenses: Except for eliminating hidden backdoors directly, trigger synthesis based defenses propose to synthesizes the backdoor trigger at first, following by the second stage that the hidden backdoor is eliminated by suppressing the effect of the (synthesized) trigger.

This type of defense enjoys certain similarities with model reconstruction based ones in the second stage. For example, pruning and retraining are the common techniques used in removing the hidden backdoor in both types of defenses. However, compared with the model reconstruction based defenses, the trigger information obtained in synthesis based defenses makes the removal process more effective and efficient.

Wang et al. [80] first proposed to remove the hidden backdoor based on the synthetic trigger in a 'black-box' scenario, where the training set is inaccessible. Specifically, the proposed method, $i.e.$ Neural Cleanse, first obtained potential trigger patterns towards every class, and then determined the final synthetic trigger and its target label based on an anomaly detector at the first stage. In the second stage, they evaluated two possible strategies, $i.e.$, an early detector for identifying the existence of trigger and a model patching algorithm based on pruning or retraining. Similar idea was also discussed in [81], [88], [89]. Qiao et al. [82] noticed that the reversed trigger synthesized by [80] is usually significantly different from that was used in the training process, inspired by which they first discussed the generalization of the backdoor trigger. They demonstrated that an infected model generalizes its original trigger during the training process. Accordingly, they proposed to recover the trigger distribution rather than a specific trigger based on a max-entropy staircase approximator for building a more backdoor-robust model. A similar idea was also discussed by Zhu et al. [84], where they proposed a GAN-based trigger synthesis method for the backdoor defense. In [83], they showed that the detection process used for determining the synthetic trigger in [80] suffers from several failure modes, based on which they proposed a new defense method. Besides, Cheng et al. [85] revealed that the ℓ^∞ norm of the activation values can be used to distinguish backdoor related neurons based on the synthetic trigger. Accordingly, they proposed to perform ℓ^∞-based neuron pruning, which removes neurons with high activation values in response to the trigger from the final convolutional layer, to defend against attacks. Similarly, Aiken et al. [86] also proposed to remove the hidden backdoor by pruning DNNs based on the synthetic trigger from another perspective. An online Neural-Cleanse-like defense method was also discussed in [87]. Most recently, Shen et al. [90] proposed an efficient trigger synthesis based defense. Different from previous approaches, which needed to generate all potential triggers towards each class, the proposed method selects only one class for trigger optimization in each round, inspired by the K-Arm bandit [116].

4) Model Diagnosis based Defenses: Model diagnosis based defenses justify whether the provided model is infected through a trained meta-classifier and refuse to deploy infected models. Since only the benign model is used for deployment, it naturally eliminates the hidden backdoor.

To the best of our knowledge, Kolouri el al. [23] first discussed how to diagnose a given model. Specifically, they jointly optimized some universal litmus patterns (ULPs) and a classifier, which is further used to determine whether a given model is infected based on the prediction of obtained universal litmus patterns. Similarly, Xu et al. [92] proposed two strategies to train the meta-classifier without knowing the attack strategies. Different from the previous approach where both infected model samples and benign model samples are required in the training set, an effective meta-classifier can be trained only with benign model samples based on the strategies proposed in [92]. Besides, motivated by the observation that the heatmaps from benign and infected models have different characteristics, Huang et al. [91] proposed to adopt an outlier detector as the meta-classifier based on three extracted features of generated saliency maps. In [93], they designed an one-pixel signature representation, based on which to distinguish benign and infected models. Most recently, Wang et al. [94] discussed how to detect whether a given mode is benign or infected in the data-limited and data-free cases.

5) Poison Suppression based Defenses: Poison suppression based defenses depress the effectiveness of poisoned samples during the training process to prevent the creation of hidden backdoor. Du et al. [95] first explored such type of defenses, where they adopted noisy SGD to learn differentially private DNNs for the defense. With the randomness in the training process, the contribution of poisoned samples will be reduced by random noise, resulting in the creation failure of the backdoor. Motivated by the observation that the ℓ^2 norm of the gradient of poisoned samples have significantly higher magnitudes than those of benign samples and their gradient orientations are also different, Hong et al. [96] adopted differentially private stochastic gradient descent (DPSGD) to clip and perturb individual gradients during the training process. Accordingly, the trained model has no hidden backdoor as well as its adversarial robustness towards targeted adversarial attacks is also increased.

6) Training Sample Filtering based Defenses: Training sample filtering based defenses aim at distinguishing between benign samples and poisoned samples. Only benign samples or purified poisoned samples will be used in the training process, which eliminates the backdoor from the source.

Tran et al. [97] first explored such type of defenses, where they demonstrated that poisoned samples tend to leave behind a detectable trace in the spectrum of the covariance of feature representations. Accordingly, the singular value decomposition of the covariance matrix of feature representations can be used to filter poisoned samples from the training set. Also inspired by the idea that poisoned samples and benign samples should have different characteristics in the feature space, Chen et al. [98] proposed to identify poisoned samples through a two-stage method, including **(1)** clustering the activations of training samples of each class into two clusters and **(2)** determining which, if any, of the clusters corresponds to poisoned samples. Tang et al. [99] demonstrated that simple target contamination can cause the representation of a poisoned sample to be less distinguishable from that of benign one, therefore existing filtering-based defenses can be bypassed. To address this problem, they proposed a more robust sample filter based on representation decomposition and its statistical analysis. Similarly, Soremekun et al. [100] proposed to counter poisoned samples based on the difference between benign and poisoned samples in the feature space. Different from previous methods, Chan et al. [101] separated poisoned samples based on the poison signal in the input gradients. A similar idea was explored in [102], where they adopted the saliency map to identify trigger region and filter samples.

7) Testing Sample Filtering based Defenses: Similar to training samples filtering based ones, testing samples filtering based defenses also aim at distinguishing between malicious samples and benign samples. However, compared with the previous type of methods, testing samples filtering based ones are adopted in the inference instead of the training stage. Only benign or purified attacked samples will be predicted, which prevents backdoor activation by removing the trigger.

Motivated by the observation that most of existing backdoor triggers are input-agnostic, Gao et al. [22] proposed to filter attacked samples through superimposing various image patterns and observe the randomness of the prediction of perturbed inputs. The smaller the randomness, the higher the probability to be the attacked sample. In [103], Subedar et al. adopted model uncertainty to distinguish between benign and attacked samples. Du et al. [95] treated it as the outlier detection and proposed a differential privacy based method. Besides, Jin et al. [104] proposed to detect attacked samples motivated by existing methods adopted in detection-based adversarial defenses [117], [118], [119]. Most recently, Javaheripi et al. [105] proposed a lightweight method which can filter attacked samples without the need for labeled data, model retraining, or prior assumptions on the trigger or the attack.

B. Certified Backdoor Defenses

Although multiple empirical backdoor defenses have been proposed and reached decent performance against previous attacks, almost all of them were bypassed by following stronger adaptive attacks [134], [135]. To terminate such a cat-and-mouse game, Wang et al. [24] took the first step towards the certified defense against backdoor attacks based on the *random smoothing* technique [111]. Randomized smoothing was originally developed to certify robustness against adversarial examples, where the smoothed function is built from the base function via adding random noise to the data vector to certify the robustness of a classifier under certain conditions. Similar to [136], Wang et al. treated the entire training procedure of the classifier as the base function to generalize classical randomized smoothing to defend against backdoor attacks. In [25], Weber et al. demonstrated that directly applying randomized smoothing, as in [24], will not provide high certified robustness bounds. Instead, they proposed a unified framework

TABLE V
Summary of benchmark datasets used in image recognition.

Category	Datasets	# Image Size	# Training Samples	# Testing Samples	Cited Literature
Natural Image Recognition	MNIST [120]	28×28	60,000	10,000	[72], [18], [91], [92], [74] [80], [98], [7], [81], [22] [103], [12], [121], [51], [61] [86], [23], [95], [24], [25] [100], [104], [62], [84], [122] [123], [93], [105], [78]
	Fashion MNIST [124]	28×28	60,000	10,000	[93], [96], [100]
	CIFAR [68]	$32 \times 32 \times 3$	50,000	10,000	[60], [97], [10], [11], [73] [82], [92], [22], [103], [101] [12], [13], [33], [15], [61] [86], [23], [87], [96], [25] [122], [100], [69], [19], [9] [84], [88], [94], [62], [77] [123], [63], [36], [89], [76] [29], [79]
	SVHN [125]	$32 \times 32 \times 3$	73,257	26,032	[77], [69], [19]
	ImageNet [126]	$224 \times 224 \times 3$	1,281,167	50,000	[60], [83], [99], [28], [8] [13], [102], [25], [122], [20] [19], [84], [93], [94], [29] [30], [90]
Traffic Sign Recognition	GTSRB [127]	—	34,799	12,630	[11], [73], [81], [85], [91] [80], [83], [22], [99], [12] [8], [75], [23], [87], [122] [69], [20], [88], [104], [62] [63], [84], [123], [93], [94] [78], [76], [105], [29], [79]
	U.S. Traffic Sign [128]	—	6,889	1,724	[21], [7], [74]
Face Recognition	YouTube Face [129]	—	3,425 videos of 1,595 people		[26], [21], [80], [87], [20] [62], [36]
	PubFig [130]	—	58,797 images of 200 people		[80], [8], [20], [104], [76]
	VGGFace [131]	—	2.6 million images of 2,622 people		[32], [74], [81], [80], [75] [17], [102], [84], [105], [29]
	VGGFace2 [132]	—	3.3 million images of 9,131 people		[18], [73]
	LFW [133]	—	13,233 images of 5,749 people		[83], [17], [102]

Note: **(1)** The sign sizes vary from 6×6 to 167×168 pixels in the U.S. Traffic Sign dataset; **(2)** There is no given division between the training set and the testing set in most face recognition datasets. Users need to divide the dataset by themselves according to their needs.

with the examination of different smoothing noise distributions and provided the tightness analysis for the robustness bound.

C. Evaluation Metrics

Evaluation Metrics for Detection-like Empirical Defenses. Model diagnosis based defenses and testing sample filtering based defenses are all detection-like methods, whose main target is to identify whether an (untrusted) object ($e.g.$, a trained DNN or test image) is malicious. This is essentially a binary classification problem. To evaluate their performance, three metrics, including **(1)** precision, **(2)** recall, and **(3)** F1-score, are usually adopted [137]. The higher the precision, recall, and F1-score, the better the attack performance.

Evaluation Metrics for Non-detection-like Empirical Defenses. Except for detection-like empirical defenses, other types of empirical defenses, including pre-processing based defenses, model reconstruction based defenses, trigger synthesis based defenses, poison suppression based defenses, and training sample filtering based defenses, are all non-detection-like ones. Their main target is to achieve correct predictions for both benign and attacked samples. Accordingly, both benign accuracy and attack success rate (as defined in Section II-A) are also adopted for the evaluation. In particular, although a detection process is also involved in training sample filtering based defenses, three metrics ($i.e.$, precision, recall, and F1-score) described above are not suitable for their evaluation. These defenses may try to discard as many poisoned samples as possible to reduce the possibility of creating hidden backdoors trained on the filtered dataset, even with the sacrifice of certain benign samples.

Evaluation Metrics for Certified Backdoor Defenses. As mentioned in Section VI-B, existing certified backdoor defenses all adopted the random smoothing technique. These methods can provide a certified radius, where all perturbation within the ℓ^p ball with the certified radius can not change the prediction of the model under certain assumptions. To evaluate their performance, people usually use the **(1)** benign accuracy, **(2)** certified rate, and **(3)** certified accuracy as the evaluation metric [24], [25]. Specifically, the benign accuracy indicates how well the (smoothed) classifier performs in classifying benign samples; the certified rate is the fraction of samples that can be certified at radius greater than the certified radius; and the certified accuracy is the fraction of the test set which is classified correctly and is certified as robust with a radius greater than the certified radius. The greater the benign accuracy, certified rate, and certified accuracy, the better the attack performance.

VII. Benchmark Datasets

Similar to that of adversarial learning, most of the existing related literature focused on the image recognition task. In this section, we summarize all benchmark datasets which were used at least twice in related literature in Table V.

Those benchmark datasets can be divided into three main categories, including *natural image recognition*, *traffic sign recognition*, and *face recognition*. The first type of dataset is the classic one in the image classification field, while the second and third ones are tasks that require strict security guarantees. We recommend that future work should be evaluated on these datasets to facilitate comparison and ensure fairness.

VIII. Outlook of Future Directions

As presented above, many works have been proposed in the literature of backdoor learning, covering several branches and different scenarios. However, we believe that the development of this field is still in its infancy, as many critical problems of backdoor learning have not been well studied. In this section, we present five potential research directions to inspire the future development of backdoor learning.

A. Trigger Design

The effectiveness and efficiency of poisoning-based backdoor attacks are closely related to their trigger patterns. However, the trigger of existing methods was designed in a heuristic ($e.g.$, design with universal perturbation), or even a non-optimized way. How to better optimize the trigger pattern is still an important open question. Besides, only the effectiveness and invisibility were considered in the trigger design, other criteria, such as with minimized necessary poisoned proportion, are also worth further exploration.

B. Semantic and Physical Backdoor Attacks

As presented in Section III-C, semantic and physical attacks are more serious threats to AI systems in practical scenarios, while their studies are still left far behind, compared to other types of backdoor attacks. More thorough studies to obtain better understandings of this two attacks would be important steps towards alleviating the backdoor threat in practice.

C. Attacks Towards Other Tasks

The success of backdoor attacks is significantly due to the specific design of triggers according to the characteristics of the target task. For example, the visual invisibility of the trigger is one of the critical criteria in visual tasks, which ensures the attack stealthiness. However, the design of backdoor triggers in different tasks could be quite different ($e.g.$, hiding a trigger into a sentence when attacking a task in natural language processing is quite different with hiding a trigger into an image). Accordingly, it is of great significance to study task-specified backdoor attacks. Existing backdoor attacks mainly focused on the tasks of computer vision, especially image classification. However, the research towards other tasks ($e.g.$, recommendation system, speech recognition, and natural language processing) have not been well studied.

D. Effective and Efficient Defenses

Although many types of empirical backdoor defenses have been proposed (see Section VI), almost all of them can be bypassed by subsequent adaptive attacks. Besides, except for the pre-processing based defenses, the high computational cost is also one common drawback of most existing defenses. More efforts on designing effective and efficient defense methods ($e.g.$, analyzing the weaknesses of existing attacks and how to reduce the computational cost of defenses) should be made to keep up the fast pace of backdoor attacks. Moreover, certified backdoor defenses are important yet currently have been rarely studied, which deserve more explorations.

E. Mechanism Exploration

The principle of backdoor generation and the activation mechanism of backdoor triggers are the holy grail problems in the field of backdoor learning. For example, why does the backdoor exist, and what happens inside the model when the backdoor trigger appears, have not been carefully studied in existing works. The intrinsic mechanism of backdoor learning is supposed to serve as the key role to guide the design of backdoor attacks and defenses.

IX. Conclusion

Backdoor learning, including backdoor attacks and backdoor defenses, is a critical and booming research area. In this survey, we summarize and categorize existing backdoor attacks and propose a unified framework for analyzing poisoning-based backdoor attacks. We also analyze the defense techniques and discuss the relation between backdoor attacks and related realms. The potential research directions are illustrated at the end. Almost all researches in this field were completed in the last three years, and the cat-and-mouse game between attacks and defenses is likely to continue in the future. We hope that this paper could remind researchers of the backdoor threat and provide a timely view. It would be an important step towards trust-worthy deep learning.

Acknowledgements

This work was done when Yiming Li was an intern at Tencent AI Lab, supported by the Tencent Rhino-Bird Elite Training Program (2020). This work is also supported in part by the National Key Research and Development Program of China under Grant 2018YFB1800204, the National Natural Science Foundation of China under Grant 61771273 and 62076213, the Natural Science Foundation of Zhejiang Province (LSY19A010002), University Development Fund of CUHK(SZ) (UDF01001810), the project "PCL Future Greater-Bay Area Network Facilities for Large-scale Experiments and Applications" (LZC0019), and the R&D Program of Shenzhen (JCYJ20180508152204044). We also sincerely thank Yalei Lv, Jia Xu, Haoxiang Zhong, and Ziqi Zhang from Tsinghua Shenzhen International Graduate School, Tsinghua University, for their helpful comments on an early draft of this paper.

REFERENCES

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[2] Y. Fan, B. Wu, T. Li, Y. Zhang, M. Li, Z. Li, and Y. Yang, "Sparse adversarial attack via perturbation factorization," in *ECCV*, 2020.

[3] J. Bai, B. Chen, Y. Li, D. Wu, W. Guo, S.-t. Xia, and E.-h. Yang, "Targeted attack for deep hashing based retrieval," in *ECCV*, 2020.

[4] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *ICLR*, 2018.

[5] J. Xu, Y. Li, Y. Jiang, and S.-T. Xia, "Adversarial defense via local flatness regularization," in *ICIP*, 2020.

[6] D. Wu, S. Xia, and Y. Wang, "Adversarial weight perturbation helps robust generalization," in *NeurIPS*, 2020.

[7] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.

[8] Y. Liu, X. Ma, J. Bailey, and F. Lu, "Reflection backdoor: A natural backdoor attack on deep neural networks," in *ECCV*, 2020.

[9] Y. Li, T. Zhai, B. Wu, Y. Jiang, Z. Li, and S. Xia, "Rethinking the trigger of backdoor attack," *arXiv preprint arXiv:2004.04692*, 2020.

[10] A. Turner, D. Tsipras, and A. Madry, "Label-consistent backdoor attacks," *arXiv preprint arXiv:1912.02771*, 2019.

[11] S. Li, M. Xue, B. Zhao, H. Zhu, and X. Zhang, "Invisible backdoor attacks on deep neural networks via steganography and regularization," *IEEE Transactions on Dependable and Secure Computing*, 2020.

[12] H. Zhong, C. Liao, A. C. Squicciarini, S. Zhu, and D. Miller, "Backdoor embedding in convolutional neural network models via invisible perturbation," in *ACM CODASPY*, 2020.

[13] A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden trigger backdoor attacks," in *AAAI*, 2020.

[14] S. Zhao, X. Ma, X. Zheng, J. Bailey, J. Chen, and Y.-G. Jiang, "Clean-label backdoor attacks on video recognition models," in *CVPR*, 2020.

[15] E. Quiring and K. Rieck, "Backdooring and poisoning neural networks with image-scaling attacks," in *IEEE S&P Workshop*, 2020.

[16] K. Kurita, P. Michel, and G. Neubig, "Weight poisoning attacks on pre-trained models," in *ACL*, 2020.

[17] S. Wang, S. Nepal, C. Rudolph, M. Grobler, S. Chen, and T. Chen, "Backdoor attacks against transfer learning with pre-trained deep learning models," *IEEE Transactions on Services Computing*, 2020.

[18] J. Dumford and W. Scheirer, "Backdooring convolutional neural networks via targeted weight perturbations," *arXiv preprint arXiv:1812.03128*, 2018.

[19] A. S. Rakin, Z. He, and D. Fan, "Tbt: Targeted neural network attack with bit trojan," in *CVPR*, 2020.

[20] R. Tang, M. Du, N. Liu, F. Yang, and X. Hu, "An embarrassingly simple approach for trojan attack in deep neural networks," in *KDD*, 2020.

[21] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *RAID*, 2018.

[22] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "Strip: A defence against trojan attacks on deep neural networks," in *ACSAC*, 2019.

[23] S. Kolouri, A. Saha, H. Pirsiavash, and H. Hoffmann, "Universal litmus patterns: Revealing backdoor attacks in cnns," in *CVPR*, 2020.

[24] B. Wang, X. Cao, N. Z. Gong *et al.*, "On certifying robustness against backdoor attacks via randomized smoothing," in *CVPR Workshop*, 2020.

[25] M. Weber, X. Xu, B. Karlas, C. Zhang, and B. Li, "Rab: Provable robustness against backdoor attacks," *arXiv preprint arXiv:2003.08904*, 2020.

[26] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv preprint arXiv:1712.05526*, 2017.

[27] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *CVPR*, 2017.

[28] E. Bagdasaryan and V. Shmatikov, "Blind backdoors in deep learning models," *arXiv preprint arXiv:2005.03823*, 2020.

[29] S. Cheng, Y. Liu, S. Ma, and X. Zhang, "Deep feature space trojan attack of neural networks by controlled detoxification," in *AAAI*, 2021.

[30] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, "Backdoor attack with sample-specific triggers," *arXiv preprint arXiv:2012.03816*, 2020.

[31] Q. Xiao, Y. Chen, C. Shen, Y. Chen, and K. Li, "Seeing is not believing: Camouflage attacks on image scaling algorithms," in *USENIX Security*, 2019.

[32] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *NDSS*, 2017.

[33] S. Garg, A. Kumar, V. Goel, and Y. Liang, "Can adversarial weight perturbations inject neural backdoors?" in *CIKM*, 2020.

[34] E. Wenger, J. Passananti, Y. Yao, H. Zheng, and B. Y. Zhao, "Backdoor attacks on facial recognition in the physical world," *arXiv preprint arXiv:2006.14580*, 2020.

[35] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *AISTATS*, 2020.

[36] J. Lin, L. Xu, Y. Liu, and X. Zhang, "Composite backdoor attack for deep neural network by mixing existing benign features," in *CCS*, 2020.

[37] J. Dai, C. Chen, and Y. Li, "A backdoor attack against lstm-based text classification systems," *IEEE Access*, vol. 7, pp. 138 872–138 878, 2019.

[38] X. Chen, A. Salem, M. Backes, S. Ma, and Y. Zhang, "Badnl: Backdoor attacks against nlp models," *arXiv preprint arXiv:2006.01043*, 2020.

[39] A. Chan, Y. Tay, Y.-S. Ong, and A. Zhang, "Poison attacks against text datasets with conditional adversarially regularized autoencoder," in *EMNLP-Findings*, 2020.

[40] Z. Zhang, J. Jia, B. Wang, and N. Z. Gong, "Backdoor attacks to graph neural networks," in *NeurIPS Workshop*, 2020.

[41] Z. Xi, R. Pang, S. Ji, and T. Wang, "Graph backdoor," *arXiv preprint arXiv:2006.11890*, 2020.

[42] P. Kiourti, K. Wardega, S. Jha, and W. Li, "Trojdrl: Trojan attacks on deep reinforcement learning agents," *arXiv preprint arXiv:1903.06638*, 2019.

[43] Z. Yang, N. Iyer, J. Reimann, and N. Virani, "Design of intentional backdoors in sequential models," *arXiv preprint arXiv:1902.09972*, 2019.

[44] Y. Wang, E. Sarkar, M. Maniatakos, and S. E. Jabari, "Stop-and-go: Exploring backdoor attacks on deep reinforcement learning-based traffic congestion control systems," *arXiv preprint arXiv:2003.07859*, 2020.

[45] T. Zhai, Y. Li, Z. Zhang, B. Wu, Y. Jiang, and S.-T. Xia, "Backdoor attack against speaker verification," in *ICASSP*, 2021.

[46] K. Davaslioglu and Y. E. Sagduyu, "Trojan attacks on wireless signal classification with adversarial machine learning," in *DySPAN*, 2019.

[47] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *ICML*, 2019.

[48] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "Dba: Distributed backdoor attacks against federated learning," in *ICLR*, 2019.

[49] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, "Attack of the tails: Yes, you really can backdoor federated learning," in *NeurIPS*, 2020.

[50] C.-L. Chen, L. Golubchik, and M. Paolieri, "Backdoor attacks on federated meta-learning," *arXiv preprint arXiv:2006.07026*, 2020.

[51] Y. Liu, Z. Yi, and T. Chen, "Backdoor attacks and defenses in feature-partitioned collaborative learning," in *ICML Workshop*, 2020.

[52] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?" in *NeurIPS Workshop*, 2019.

[53] S. Fu, C. Xie, B. Li, and Q. Chen, "Attack-resistant federated learning with residual-based reweighting," *arXiv preprint arXiv:1912.11464*, 2019.

[54] S. Li, Y. Cheng, W. Wang, Y. Liu, and T. Chen, "Learning to detect malicious clients for robust federated learning," *arXiv preprint arXiv:2002.00211*, 2020.

[55] M. Naseri, J. Hayes, and E. De Cristofaro, "Toward robustness and privacy in federated learning: Experimenting with local and central differential privacy," *arXiv preprint arXiv:2009.03561*, 2020.

[56] H. B. Desai, M. S. Ozdayi, and M. Kantarcioglu, "Blockfla: Accountable federated learning via hybrid blockchain architecture," *arXiv preprint arXiv:2010.07427*, 2020.

[57] M. Safa Ozdayi, M. Kantarcioglu, and Y. R. Gel, "Defending against backdoors in federated learning with robust learning rate," *arXiv e-prints*, pp. arXiv–2007, 2020.

[58] Y. Yao, H. Li, H. Zheng, and B. Y. Zhao, "Latent backdoor attacks on deep neural networks," in *CCS*, 2019.

[59] Z. Zhang, G. Xiao, Y. Li, T. Lv, F. Qi, Y. Wang, X. Jiang, Z. Liu, and M. Sun, "Red alarm for pre-trained models: Universal vulnerabilities by neuron-level backdoor attacks," *arXiv preprint arXiv:2101.06969*, 2021.

[60] Y. Adi, C. Baum, M. Cisse, B. Pinkas, and J. Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," in *USENIX Security*, 2018.

[61] D. M. Sommer, L. Song, S. Wagh, and P. Mittal, "Towards probabilistic verification of machine unlearning," *arXiv preprint arXiv:2003.04247*, 2020.

[62] S. Shan, E. Wenger, B. Wang, B. Li, H. Zheng, and B. Y. Zhao, "Using honeypots to catch adversarial attacks on neural networks," in *CCS*, 2020.

[63] Y. Li, Z. Zhang, J. Bai, B. Wu, Y. Jiang, and S.-T. Xia, "Open-sourced dataset protection via backdoor watermarking," in *NeurIPS Workshop*, 2020.

[64] S. Zhao, X. Ma, Y. Wang, J. Bailey, B. Li, and Y.-G. Jiang, "What do deep nets learn? class-wise patterns revealed in the input space," *arXiv preprint arXiv:2101.06898*, 2021.

[65] Y.-S. Lin, W.-C. Lee, and Z. B. Celik, "What do you see? evaluation of explainable artificial intelligence (xai) interpretability through neural backdoors," *arXiv preprint arXiv:2009.10639*, 2020.

[66] K. Razavi, B. Gras, E. Bosman, B. Preneel, C. Giuffrida, and H. Bos, "Flip feng shui: Hammering a needle in the software stack," in *USENIX Security*, 2016.

[67] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[68] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[69] C. Guo, R. Wu, and K. Q. Weinberger, "Trojannet: Embedding hidden trojan horse models in neural networks," *arXiv preprint arXiv:2002.10078*, 2020.

[70] K. R. Mopuri, A. Ganeshan, and R. V. Babu, "Generalizable data-free objective for crafting universal adversarial perturbations," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 10, pp. 2452–2465, 2018.

[71] S. Thys, W. Van Ranst, and T. Goedemé, "Fooling automated surveillance cameras: adversarial patches to attack person detection," in *CVPR Workshop*, 2019.

[72] Y. Liu, Y. Xie, and A. Srivastava, "Neural trojans," in *ICCD*, 2017.

[73] B. G. Doan, E. Abbasnejad, and D. C. Ranasinghe, "Februus: Input purification defense against trojan attacks on deep neural network systems," *arXiv preprint arXiv:1908.03369*, 2019.

[74] S. Udeshi, S. Peng, G. Woo, L. Loh, L. Rawshan, and S. Chattopadhyay, "Model agnostic defence against backdoor attacks in machine learning," *arXiv preprint arXiv:1908.02203*, 2019.

[75] M. Villarreal-Vasquez and B. Bhargava, "Confoc: Content-focus protection against trojan attacks on neural networks," *arXiv preprint arXiv:2007.00711*, 2020.

[76] Y. Zeng, H. Qiu, S. Guo, T. Zhang, M. Qiu, and B. Thuraisingham, "Deepsweep: An evaluation framework for mitigating dnn backdoor attacks using data augmentation," *arXiv preprint arXiv:2012.07006*, 2020.

[77] P. Zhao, P.-Y. Chen, P. Das, K. N. Ramamurthy, and X. Lin, "Bridging mode connectivity in loss landscapes and adversarial robustness," in *ICLR*, 2020.

[78] K. Yoshida and T. Fujino, "Disabling backdoor and identifying poison data by using knowledge distillation in backdoor attacks on deep neural networks," in *CCS Workshop*, 2020.

[79] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Neural attention distillation: Erasing backdoor triggers from deep neural networks," in *ICLR*, 2021.

[80] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *IEEE S&P*, 2019.

[81] H. Chen, C. Fu, J. Zhao, and F. Koushanfar, "Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks." in *IJCAI*, 2019.

[82] X. Qiao, Y. Yang, and H. Li, "Defending neural backdoors via generative distribution modeling," in *NeurIPS*, 2019.

[83] W. Guo, L. Wang, Y. Xu, X. Xing, M. Du, and D. Song, "Towards inspecting and eliminating trojan backdoors in deep neural networks," in *ICDM*, 2020.

[84] L. Zhu, R. Ning, C. Wang, C. Xin, and H. Wu, "Gangsweep: Sweep out neural backdoors by gan," in *ACM MM*, 2020.

[85] H. Cheng, K. Xu, S. Liu, P.-Y. Chen, P. Zhao, and X. Lin, "Defending against backdoor attack on deep neural networks," in *KDD Workshop*, 2019.

[86] W. Aiken, H. Kim, and S. Woo, "Neural network laundering: Removing black-box backdoor watermarks from deep neural networks," *arXiv preprint arXiv:2004.11368*, 2020.

[87] A. K. Veldanda, K. Liu, B. Tan, P. Krishnamurthy, F. Khorrami, R. Karri, B. Dolan-Gavitt, and S. Garg, "Nnoculation: Broad spectrum and targeted treatment of backdoored dnns," *arXiv preprint arXiv:2002.08313*, 2020.

[88] H. Harikumar, V. Le, S. Rana, S. Bhattacharya, S. Gupta, and S. Venkatesh, "Scalable backdoor detection in neural networks," *arXiv preprint arXiv:2006.05646*, 2020.

[89] Z. Xiang, D. J. Miller, and G. Kesidis, "Detection of backdoors in trained classifiers without access to the training set," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[90] G. Shen, Y. Liu, G. Tao, S. An, Q. Xu, S. Cheng, S. Ma, and X. Zhang, "Backdoor scanning for deep neural networks through k-arm optimization," *arXiv preprint arXiv:2102.05123*, 2021.

[91] X. Huang, M. Alzantot, and M. Srivastava, "Neuroninspect: Detecting backdoors in neural networks via output explanations," *arXiv preprint arXiv:1911.07399*, 2019.

[92] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter, and B. Li, "Detecting ai trojans using meta neural analysis," in *IEEE S&P*, 2021.

[93] S. Huang, W. Peng, Z. Jia, and Z. Tu, "One-pixel signature: Characterizing cnn models for backdoor detection," in *ECCV*, 2020.

[94] R. Wang, G. Zhang, S. Liu, P.-Y. Chen, J. Xiong, and M. Wang, "Practical detection of trojan neural networks: Data-limited and data-free cases," in *ECCV*, 2020.

[95] M. Du, R. Jia, and D. Song, "Robust anomaly detection and backdoor attack detection via differential privacy," in *ICLR*, 2020.

[96] S. Hong, V. Chandrasekaran, Y. Kaya, T. Dumitraş, and N. Papernot, "On the effectiveness of mitigating data poisoning attacks with gradient shaping," *arXiv preprint arXiv:2002.11497*, 2020.

[97] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *NeurIPS*, 2018.

[98] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, and B. Srivastava, "Detecting backdoor attacks on deep neural networks by activation clustering," in *AAAI Workshop*, 2019.

[99] D. Tang, X. Wang, H. Tang, and K. Zhang, "Demon in the variant: Statistical analysis of dnns for robust backdoor contamination detection," in *USENIX Security*, 2021.

[100] E. Soremekun, S. Udeshi, S. Chattopadhyay, and A. Zeller, "Exposing backdoors in robust machine learning models," *arXiv preprint arXiv:2003.00865*, 2020.

[101] A. Chan and Y.-S. Ong, "Poison as a cure: Detecting & neutralizing variable-sized backdoor attacks in deep neural networks," *arXiv preprint arXiv:1911.08040*, 2019.

[102] E. Chou, F. Tramèr, and G. Pellegrino, "Sentinet: Detecting localized universal attacks against deep learning systems," in *IEEE S&P Workshop*, 2020.

[103] M. Subedar, N. Ahuja, R. Krishnan, I. J. Ndiour, and O. Tickoo, "Deep probabilistic models to detect data poisoning attacks," in *NeurIPS Workshop*, 2019.

[104] K. Jin, T. Zhang, C. Shen, Y. Chen, M. Fan, C. Lin, and T. Liu, "A unified framework for analyzing and detecting malicious examples of dnn models," *arXiv preprint arXiv:2006.14871*, 2020.

[105] M. Javaheripi, M. Samragh, G. Fields, T. Javidi, and F. Koushanfar, "Cleann: Accelerated trojan shield for embedded neural networks," in *ICCAD*, 2020.

[106] Y. Dong, H. Su, B. Wu, Z. Li, W. Liu, T. Zhang, and J. Zhu, "Efficient decision-based black-box adversarial attacks on face recognition," in *CVPR*, 2019.

[107] W. Chen, Z. Zhang, X. Hu, and B. Wu, "Boosting decision-based black-box adversarial attacks with random sign flip," in *ECCV*, 2020.

[108] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," in *NeurIPS*, 2020.

[109] R. Pang, H. Shen, X. Zhang, S. Ji, Y. Vorobeychik, X. Luo, A. Liu, and T. Wang, "A tale of evil twins: Adversarial inputs versus poisoned models," in *CCS*, 2020.

[110] C.-H. Weng, Y.-T. Lee, and S.-H. B. Wu, "On the trade-off between adversarial and backdoor robustness," in *NeurIPS*, 2020.

[111] J. M. Cohen, E. Rosenfeld, and J. Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *ICML*, 2019.

[112] 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 *ICCV*, 2017.

[113] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[114] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson, "Loss surfaces, mode connectivity, and fast ensembling of dnns," in *NeurIPS*, 2018.

[115] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NeurIPS Workshop*, 2014.

[116] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2, pp. 235–256, 2002.

[117] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," *arXiv preprint arXiv:1703.00410*, 2017.

[118] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, G. Schoenebeck, D. Song, M. E. Houle, and J. Bailey, "Characterizing adversarial subspaces using local intrinsic dimensionality," in *ICLR*, 2018.

[119] J. Wang, G. Dong, J. Sun, X. Wang, and P. Zhang, "Adversarial sample detection for deep neural network through model mutation testing," in *ICSE*, 2019.

[120] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[121] M. Umer, G. Dawson, and R. Polikar, "Targeted forgetting and false memory formation in continual learners through adversarial backdoor attacks," *arXiv preprint arXiv:2002.07111*, 2020.

[122] Y. Gao, H. Rosenberg, K. Fawaz, S. Jha, and J. Hsu, "Analyzing accuracy loss in randomized smoothing defenses," *arXiv preprint arXiv:2003.01595*, 2020.

[123] A. Nguyen and A. Tran, "Input-aware dynamic backdoor attack," in *NeurIPS*, 2020.

[124] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[125] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NeurIPS Workshop*, 2011.

[126] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[127] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural networks*, vol. 32, pp. 323–332, 2012.

[128] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund, "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1484–1497, 2012.

[129] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *CVPR*, 2011.

[130] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *ICCV*, 2009.

[131] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *BMVC*, 2015.

[132] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," in *IEEE FGR*, 2018.

[133] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[134] T. J. L. Tan and R. Shokri, "Bypassing backdoor detection algorithms in deep learning," in *EuroS&P*, 2020.

[135] A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden trigger backdoor attacks," in *AAAI*, 2020.

[136] E. Rosenfeld, E. Winston, P. Ravikumar, and J. Z. Kolter, "Certified robustness to label-flipping attacks via randomized smoothing," in *ICML*, 2020.

[137] M. Sugiyama, *Introduction to statistical machine learning*. Morgan Kaufmann, 2015.

Yiming Li received the B.S. degree in Mathematics and Applied Mathematics from Ningbo University, China, in 2018. He is currently pursuing the Ph.D. degree in Tsinghua Shenzhen International Graduate School, Tsinghua University. His research interests are in the domain of the AI Security, including Backdoor Learning, Adversarial Learning, Robust Machine Learning, and Data Privacy.

Baoyuan Wu is currently an Associate Professor of School of Data Science, the Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen). He is also the director of the Secure Computing Lab of Big Data, Shenzhen Research Institute of Big Data (SBRID). From November 2016 to August 2020, He was a Senior and Principal Research Scientist at Tencent AI lab. He was Postdoc in the IVUL lab at KAUST, working with Prof. Bernard Ghanem, from August 2014 to November 2016. He received the Ph.D. degree from the National Laboratory of Pattern Recognition, Chinese Academy of Sciences (CASIA) in 2014, supervised by Prof. Baogang Hu. His research interests are machine learning and computer vision, including AI security and privacy, probabilistic graphical models, multi-label learning and integer programming, etc. He has published 40+ top-tier conference and journal papers, including TPAMI, IJCV, CVPR, ICCV, ECCV, AAAI, etc. He serves as a senior program member of IJCAI 2020/2021, AAAI 2021, and Associate Editor of Neurocomputing.

Yong Jiang received his M.S. and Ph.D. degrees in computer science from Tsinghua University, China, in 1998 and 2002, respectively. Since 2002, he has been with the Tsinghua Shenzhen International Graduate School of Tsinghua University, Guangdong, China, where he is currently a full professor. His research interests include Computer Vision, Machine Learning, Internet Architecture and its Protocols, IP Routing Technology, etc.

Zhifeng Li is currently a top-tier principal researcher with Tencent AI Lab. He received the Ph.D. degree from the Chinese University of Hong Kong in 2006. After that, He was a postdoctoral fellow at the Chinese University of Hong Kong and Michigan State University for several years. Before joining Tencent, he was a full professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. His research interests include deep learning, computer vision and pattern recognition, and face detection and recognition. He is currently serving on the Editorial Boards of Neurocomputing and IEEE Transactions on Circuits and Systems for Video Technology. He is a fellow of the British Computer Society (FBCS).

Shu-Tao Xia received the B.S. degree in mathematics and the Ph.D. degree in applied mathematics from Nankai University, Tianjin, China, in 1992 and 1997, respectively. Since January 2004, he has been with the Tsinghua Shenzhen International Graduate School of Tsinghua University, Guangdong, China, where he is currently a full professor. From March 1997 to April 1999, he was with the research group of information theory, Department of Mathematics, Nankai University, Tianjin, China. From September 1997 to March 1998 and from August to September 1998, he visited the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong. His current research interests include coding and information theory, machine learning, and deep learning. His researches have been published in multiple top-tier journals and conferences, including IEEE TIP, IEEE TNNLS, CVPR, ICCV, ECCV, ICLR, etc.