Backdoor Learning: A Survey

Yiming Li
Tsinghua University

Backdoor Learning: A Survey

Yiming Li, Yong Jiang, Zhifeng Li, Shu-Tao Xia

***Abstract*—Backdoor attack intends to embed hidden backdoors into deep neural networks (DNNs), so that the attacked models perform well on benign samples, whereas their predictions will be maliciously changed if the hidden backdoor is activated by attacker-specified triggers. 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, there is still no comprehensive and timely review of it. 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, Backdoor Learning, 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 explored the adversarial vulnerability of DNNs at the inference stage. Compared to the inference stage, the training of DNNs contains more steps, including data collection, data cleaning and pre-processing, feature engineering, model selection and construction, training, model evaluation and fine-tuning, model saving, and model deployment. More steps mean more chances for the attackers. Meanwhile, it is well known that the powerful capability of DNNs significantly depends on a large amount of training data and computing resources. To reduce the training costs, users may choose to adopt third-party datasets, rather than to collect the training data by themselves, since there are many publicly available datasets; 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 use third-party backbones or pre-trained models. The cost of convenience is the loss of control 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.

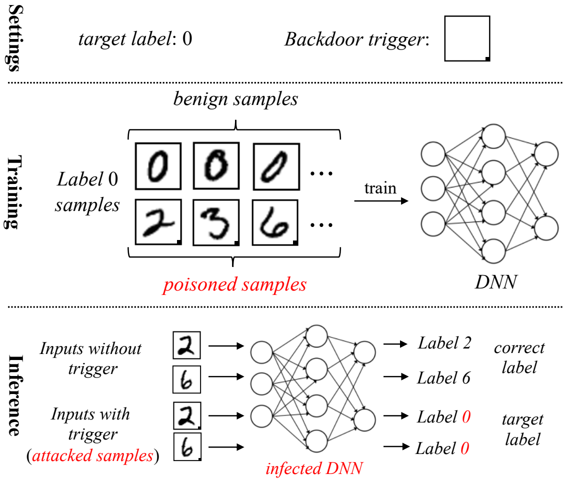

Fig. 1. An illustration of poisoning-based backdoor attacks. In this example, the trigger is a black square on the bottom right corner and the target label is '0'. Part of the benign training images are modified to have the trigger stamped, and their label is re-assigned as the attacker-specified target 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 predicting the label for the benign test images.

In general, backdoor attackers intend to embed hidden backdoors in DNNs during the training process, so that the attacked DNNs behave normally on benign samples whereas their predictions will be maliciously and consistently changed if hidden backdoors are activated by attacker-specified trigger patterns. These attacks may lead serious consequences in mission-critical applications. For example, the adversaries could cause a backdoored automated driving system to incorrectly identify traffic signs attached with the backdoor trigger, causing traffic accidents. Currently, poisoning training samples [7], [8], [9] is the most straightforward and widely adopted method to encode backdoor functionality during the training process. For example, 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 labels and remaining 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 be 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

Manuscript received xxx, xxx; revised xxx, xxx.

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

Yong Jiang and Shu-Tao Xia are with Tsinghua Shenzhen International Graduate School, Tsinghua University, and also with Research Center of Artificial Intelligence, Peng Cheng Laboratory, Shenzhen, China (e-mail: jiangy@sz.tsinghua.edu.cn, xiast@sz.tsinghua.edu.cn).

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

[1] *Backdoor* is also commonly called *neural trojan* or *trojan*. In this survey, we use 'backdoor' instead of other terms since it is most frequently used.

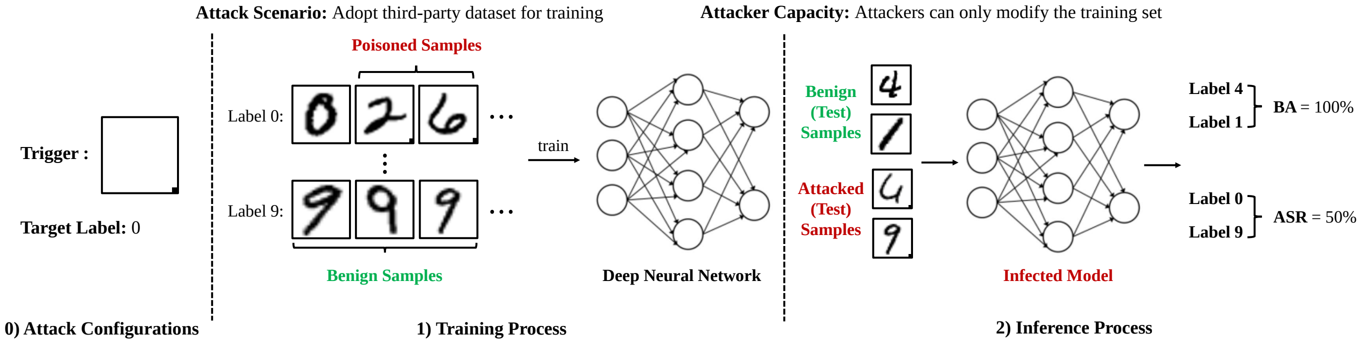

Fig. 2. The illustration of technical terms.

[16], [17], [18], directly modifying model parameters [19], [20], [21], and adding extra malicious modules [22], [23], [24]. In other words, backdoor attacks may happen at all steps involved in the training process.

To alleviate the backdoor threat, different defenses were proposed. In general, those methods can be divided into two main categories, including *empirical backdoor defenses* and *certified backdoor defenses*. Empirical backdoor defenses [25], [26], [27] 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 [28], [29], [30] is theoretically guaranteed under certain assumptions, whereas its performance is generally weaker than that of empirical defenses in practice since those assumptions are usually unsatisfied. How to better defend against backdoor attacks is still an important open question.

Given the fast development of backdoor attacks and defenses, in this survey, we intend to provide a timely overview and discussion of existing methods. Different from concurrent papers which summarized only limited research [31], [32], [33] or classified existing methods simply by the adversary capabilities [34], [35], [36], we provide a brief yet comprehensive review as well as the taxonomy for existing methods based on their characteristics and properties. To the best of our knowledge, this is the first systematic taxonomy for backdoor attacks and defenses. With this taxonomy, researchers and practitioners can better identify the properties and limitations of each method to facilitate the design of more advanced approaches. We hope that our survey can inspire more understandings of backdoor attacks and defenses, to facilitate the design of more robust and secure DNNs.

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 areas, while Section VI categorizes existing backdoor defenses. Section VII illustrates existing benchmark datasets and toolboxes, while Section VIII discusses remaining challenges and suggests future research 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. 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 containing backdoor trigger(s).
- *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 successfully 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 intends to design an infected model that performs well on the benign testing sample while achieving high attack success rate.
- *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.

The illustration of main technical terms is shown in Fig. 2.

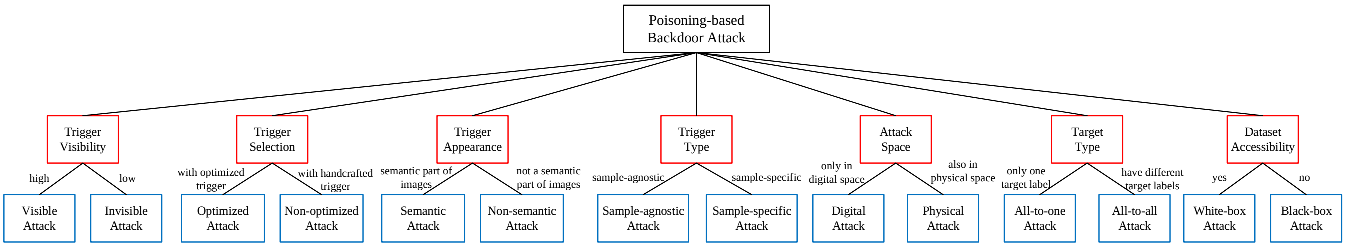

Fig. 3. Taxonomy of poisoning-based backdoor attacks with different categorization criteria. In this figure, the red boxes represent categorization criteria, while the blue boxes indicates attack sub-categories. Please refer to Table II for more technical details.

TABLE I

Three classical scenarios and correspondingly attacker's and defender's 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 partly uncontrollable for defenders when using the third-party model's API, while it is controllable when adopting pre-trained models).

B. Classical Scenarios and Corresponding Capacities

In this section, we introduce three classical real-world scenarios that backdoor threats could occur, and their corresponding attacker's and defender's capacities. More details are summarized 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 their models, which will then be deployed. 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 cannot control the training set and schedule, while they 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.

In particular, attackers' capacities increase while defenders' capacities decrease from Scenario 1 to Scenario 3. In other words, attacks designed for a previous scenario could also occur in the following ones; similarly, defenses designed for a later scenario could also be used in previous ones.

III. Poisoning-based Backdoor Attacks

In the past four 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 positive applications of backdoor attacks are also discussed at the end.

A. A Unified Framework of Poisoning-based Attacks

Poisoning-based backdoor attacks can be categorized based on different property-related criteria, as shown in Fig. 3 and summarized in Table II. More details are as follows:

We denote the classifier as $f_{\boldsymbol{w}}: \mathcal{X} \rightarrow [0,1]^K$, where $\boldsymbol{w}$ is the model parameters, $\mathcal{X} \subset \mathbb{R}^d$ 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 $G_{\boldsymbol{t}}: \mathcal{X} \rightarrow \mathcal{X}$ indicates the attacker-specified poisoned image generator with trigger pattern $\boldsymbol{t}$, and $S: \mathcal{Y} \rightarrow \mathcal{Y}$ is the attacker-specified label shifting function. Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ indicates a benign dataset, three classical risks (with respect to $\mathcal{D}$) involved in existing backdoor attacks can be defined as follows:

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

- *The standard risk R_s measures whether the infected model C can correctly predict benign samples, i.e.,*

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

where $\mathcal{P}_{\mathcal{D}}$ indicates the distribution behind $\mathcal{D}$ and $\mathbb{I}(\cdot)$ is the indicator function. $\mathbb{I}\{A\} = 1$ if and only if the event 'A' is true.

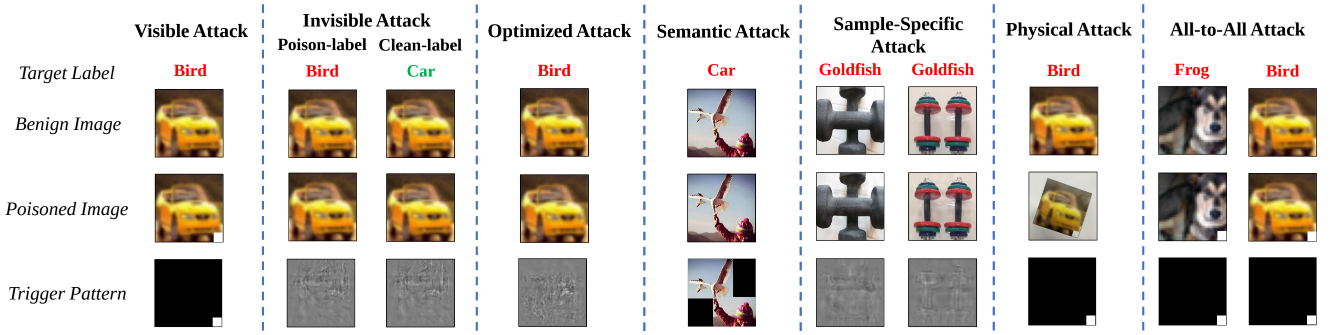

Fig. 4. An example of poisoned samples generated by different types of backdoor attacks. **(1)** In the visible attack, the backdoor trigger is a white-square stamped on the bottom right corner of the poisoned image, which is visible. **(2)** In the invisible attack, the trigger is a noise with a small magnitude, 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 optimized attack, the trigger is optimized through the targeted universal adversarial attack associated with the target class instead of a simple handcraft pattern. **(4)** 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'. **(5)** In the sample-specific attack, the trigger patterns are sample-specific instead of sample-agnostic. **(6)** In the physical attack, the (digital) poisoned image is captured by the camera from the physical space. **(7)** Different from all-to-one attacks where all poisoned samples have the same target label, different poisoned samples may have different target labels in the all-to-all attack.

TABLE II
Summary of existing poisoning-based backdoor attacks.

$\min_{\boldsymbol{t},\boldsymbol{w}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}_{\mathcal{D}_t-\mathcal{D}_s}} \{\mathbb{I}\{C(\boldsymbol{x})\neq y\}\} + \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}_{\mathcal{D}_s}} \{\lambda_1 \cdot \mathbb{I}\{C(\boldsymbol{x}')\neq S(y)\} + \lambda_2 \cdot D(\boldsymbol{x}')\}$, where $\boldsymbol{t}\in\mathcal{T}$ and $\boldsymbol{x}'=G_{\boldsymbol{t}}(\boldsymbol{x})$.				
Visible Attack	$D(\boldsymbol{x}')=1$.	Invisible Attack	Clean-label	$D(\boldsymbol{x}')=0$, and $y_t=y$.
			Poison-label	$D(\boldsymbol{x}')=0$, and $y_t\neq y$.
Optimized Attack	$\lvert\mathcal{T}\rvert>1$.	Non-optimized Attack		$\lvert\mathcal{T}\rvert=1$.
Semantic Attack	$\boldsymbol{t}$ is a semantic part of samples.	Non-semantic Attack		$\boldsymbol{t}$ is not a semantic part of samples.
Sample-agnostic Attack	All $\boldsymbol{x}'$ contain the same $\boldsymbol{t}$.	Sample-specific Attack		Trigger patterns are sample-specific.
Digital Attack	$\boldsymbol{x}'$ is generated in digital space.	Physical Attack		Physical space is involved in generating $\boldsymbol{x}'$.
All-to-one Attack	All $\boldsymbol{x}'$ have the same label.	All-to-all Attack		Different $\boldsymbol{x}'$ have different labels.
White-box Attack	$\mathcal{D}_t$ is known.	Black-box Attack		$\mathcal{D}_t$ is unknown.

- *The backdoor risk R_b indicates whether backdoor attackers can successfully achieve their malicious purposes in predicting attacked samples, $i.e.$,*

$$R_b(\mathcal{D}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{P}_{\mathcal{D}}} \mathbb{I}\{C(\boldsymbol{x}')\neq S(y)\}, \tag{2}$$

where $\boldsymbol{x}'=G_{\boldsymbol{t}}(\boldsymbol{x})$ is the attacked image.

- *The perceivable risk R_p denotes whether the poisoned sample is detectable (by human or machine), $i.e.$,*

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

where $D(\cdot)$ is an indicator function. $D(\boldsymbol{x}')=1$ if and only if $\boldsymbol{x}'$ can be detected as the malicious sample.

Given a benign training set $\mathcal{D}_t$, existing poisoning-based backdoor attacks can be summarized in a unified framework based on aforementioned definitions, as follows:

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

where $\boldsymbol{t}\in\mathcal{T}$, λ_1 and λ_2 are two non-negative trade-off hyper-parameters, and $\mathcal{D}_s$ is a subset of $\mathcal{D}_t$. In particular, $\frac{|\mathcal{D}_s|}{|\mathcal{D}_t|}$ is called *poisoning rate* in existing works.

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 $\lambda_1=\frac{|\mathcal{D}_s|}{|\mathcal{D}_t-\mathcal{D}_s|}$, $\lambda_2=0$, and $\boldsymbol{t}$ is non-optimized ($i.e.$, $|\mathcal{T}|=1$), it reduces to the BadNets [7] and the blended attack [10]; when $\lambda_2=+\infty$ and $D(\boldsymbol{x}')=\mathbb{I}\{||\boldsymbol{x}'-\boldsymbol{x}||_p\leq\epsilon\}$, it reduces to ϵ-bounded invisible backdoor attacks [11]. Besides, parameters $\boldsymbol{t}$ and $\boldsymbol{w}$ could be optimized simultaneously or separately.

In particular, this framework can be easily generalized towards other tasks, such as speech recognition, as well. Since there were many different types of tasks and their papers were limited, this generalization is out of the scope of this survey.

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 BA and ASR, the better the attack. Besides, the smaller the poisoning rate and the perturbation between the benign image and the poisoned image, the more *stealthy* the attack.

C. Attacks for Image and Video Classification

1) BadNets: Gu et al. [7] introduced the first backdoor attack in deep learning by poisoning some training samples. This method was called BadNets. Specifically, as demonstrated in Fig. 1, its training process consists of two main parts, including **(1)** generate some poisoned images via stamping the backdoor trigger onto selected benign images to achieve poisoned sample $(\boldsymbol{x}', y_t)$, associated with the attacker-specified target label y_t, and **(2)** release the poisoned training set containing both poisoned and benign samples to victims for training their own models. 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. This attack could happen in all scenarios described in Section II-B and therefore 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. [10] 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 adopted in BadNets [7]). Besides, they showed 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 [13], 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. [37] adopted the universal adversarial attack [38] to generate backdoor triggers, which minimizes the ℓ^2 norm of the perturbation to ensure invisibility. After that, [11], [39], [40] proposed to regularize the ℓ^p norm of the perturbation when optimizing the backdoor trigger. Liu et al. [8] proposed to adopt a common phenomenon ($i.e.$, the reflection) as the trigger for stealthiness. Nguyen et al. [41] adopted warping-based triggers, which are more invisible for human inspection. Recently, [42] viewed the backdoor attack as a special multi-task learning, where they fulfilled the invisibility through poisoning the loss computation. Cheng et al. [43] proposed to conduct the invisible attack in the feature space via style transfer. Different from previous works whose poisoned samples were generated in the pixel domain, [44], [45] generated invisible trigger patterns in the frequency domain. Most recently, Li et al. [12] adopted DNN-based image steganography to generate invisible backdoor triggers. Compared with previous methods, this attack is not only invisible but can also bypass most existing backdoor defenses, since its trigger patterns are sample-specific.

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 examining the image-label relationship of training samples. To address this problem, a special subclass of invisible poisoning-based attacks, dubbed *clean-label invisible attacks*, was proposed. It is more serious and therefore worth more attention. Turner et al. [13] 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 process 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. [15] extended this idea in attacking video classification, where they adopted universal perturbation instead of a given one as the trigger pattern. Quiring et al. [46] proposed to conceal the trigger via image-scaling attacks [47]. 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 the target class by minimizing their distance in the feature space, as suggested in [14]. Following the settings in [14], Souri et al. [48] formulated the backdoor attacks as a bi-level optimization [49], based on which they proposed a new clean-label backdoor attack. Most recently, Shumailov et al. [50] proposed to inject hidden backdoors via manipulating the order of training samples without changing samples.

In particular, clean-label backdoor attacks usually suffered from low attack effectiveness compared with poison-label attacks, although they are more stealthy. How to balance the stealthiness and effectiveness of attacks is still an open question and worth further explorations.

3) Optimized Backdoor Attacks: Triggers are the core of poisoning-based attacks. As such, analyzing how to design a better trigger instead of simply using a given non-optimized patch is of great significance and has attracted some attention. In general, backdoor attacks can be formulated as a bi-level optimization [49], $i.e.$, $\min_{\boldsymbol{w}} R_s(\mathcal{D}_t - \mathcal{D}_s; \boldsymbol{w}) + \lambda_1 \cdot R_b(\mathcal{D}_s; \boldsymbol{t}^*, \boldsymbol{w}) + \lambda_2 \cdot R_p(\mathcal{D}_s; \boldsymbol{t}^*, \boldsymbol{w})$, $s.t.$, $\boldsymbol{t}^* = \min_t R_b(\mathcal{D}_s; \boldsymbol{t}, \boldsymbol{w})$. Optimized attacks generated poisoned samples with optimized triggers to achieve better performance. To the best of our knowledge, Liu et al. [51] first explored this problem, where they proposed to optimize the trigger so that the important neurons can achieve the maximum values. After that, 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, [37], [15], [52] proposed to generate trigger through universal adversarial perturbation. These methods can be regarded as the heuristic solutions of the aforementioned bi-level optimization. Recently, [11], [39], [40], [48] solved the bi-level optimization problem directly. For example, [11], [39], [40] alternately optimized the upper-level and lower-level sub-problems while [48] adopted the gradient matching [53]. However, optimized backdoor attacks usually suffer from poor generalization, $i.e.$, overfits to a particular model structure or model status. Although existing works introduced model-ensemble or carefully designed the alternately optimization process to alleviate the overfitting, how to better balance the effectiveness and generalization of the optimized triggers is still an important open question.

4) Semantic Backdoor Attacks: The majority of backdoor attacks, $i.e.$, the *non-semantic attacks*, assume that backdoor triggers are independent of benign images. As such, attackers need to modify the image in the digital space to activate hidden backdoors in the inference process. That raises the question if a semantic part of samples can also serve as the trigger pattern, 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 [54], [42], $i.e.$, 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 semantic backdoors in the infected DNNs. Accordingly, the infected model will automatically misclassify testing images containing pre-defined semantic information without any image modification. A similar idea was also explored in [55], where the hidden backdoor can be activated by the combination of certain objects in the image. Since these attacks do not require modifying images in the digital space, they are more malicious and worth further explorations.

5) Sample-specific Backdoor Attacks: Currently, almost all backdoor attacks were sample-agnostic, $i.e.$, all poisoned samples contained the same trigger pattern. This property was widely used in the design of backdoor defenses, such as trigger synthesis based defenses [25], [56], [57], [58], [59] and saliency-based defenses [60], [61]. Nguyen et al. [62] proposed the first sample-specific backdoor attack, where different poisoned samples contain different trigger patterns. This attack bypassed many existing backdoor defenses for it broke their fundamental assumptions. However, it needs to control the training loss except for solely modifying training samples and their triggers are still visible, which significantly reduces its threat in real-world applications. After that, Li et al. [12] proposed the first poison-only sample-specific backdoor attack with invisible trigger patterns, inspired by the advanced DNN-based image steganography. A similar idea is also explored in [63], where they embedded trigger patterns in the edge structure of poisoned images. Since these attacks can bypass most existing backdoor defenses, they pose a serious security threat and therefore worth further explorations.

6) Physical Backdoor Attacks: Different from previous *digital attacks* where attacks were conducted completely in the digital space, the physical space was also involved when generating poisoned samples in the *physical attacks*. Chen et al. [10] first explored the landscape of this attack, where 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. [64]. 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 triggers in attacked samples. This inconsistency will greatly reduce the performance of backdoor attacks. 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.

7) All-to-all Backdoor Attacks: Based on the type of target labels, existing backdoor attacks can be divided into two main categories, including the *all-to-one attacks* and the *all-to-all attacks*. Specifically, all-to-one attacks assumed that all poisoned samples have the same target label no matter what their ground-truth label is, $i.e.$, $S(y) = y_t, \forall y \in \{1, \cdots, K\}$. In contrast, different poisoned samples may have different labels in all-to-all attacks. For example, the label shifting function was assigned as $S(y) = (y+1) \mod K$ in [7], [62], [39]. The all-to-all attacks can bypass many target-oriented defenses ($e.g.$, [25], [61], [59]) for its complicated target shifting and therefore is more serious compared with the all-to-one attacks. However, there were only a few studies in all-to-all attacks. How to better design the all-to-all attack and the analysis of its properties remain blank.

8) Black-box Backdoor Attacks: Different from previous *white-box attacks* which required to access the training samples, *black-box attacks* adopted 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. In general, black-box backdoor attackers generated some substitute training samples at first. For example, in [51], attackers 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 substitute training samples, white-box attacks can be adopted for backdoor injection. Black-box backdoor attacks are significantly more difficult than white-box ones and there were only a few works in this area.

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.

Natural language processing is currently the most extensive research field in backdoor attacks besides image classification. In [65], 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 [66], 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, other backdoor attacks were also introduced, targeting different trigger types [67], [68], [69], [70] and model components [71], [72] in different NLP tasks. Except for NLP-related tasks, researchers also revealed the backdoor threats in graph neural networks (GNN) [73], [74], [75], 3D point cloud [76], [77], [78], semi-/self-supervised learning [79], [80], [81], reinforcement learning [82], [83], [84], model quantization [85], [86], [87], acoustics signal processing [88], [89], malware detection [90], [91], and others [92], [93], [94].

Except for the classical training paradigm, how to backdoor collaborative learning, especially federated learning, have attracted the most attention. In [54], Bagdasaryan et al. introduced the first backdoor attack against federated learning by amplifying the poisoned gradient of node servers. After that, Bhagoji et al. [95] discussed the stealthy model-poisoning backdoor attack, and Xie et al. [96] introduced a distributed backdoor attacks against the federated learning. Most recently, [97] 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 [98] and feature-partitioned collaborative learning [99], [100] were also discussed. In contrast, some works [101], [30], [102], [103] also questioned whether federal learning is really easy to be attacked. Except for collaborative learning, the backdoor threat towards another important learning paradigm, $i.e.$, the transfer learning, was also discussed in [104], [17], [105].

E. Backdoor Attacks for Positive Purposes

Except for malicious applications, how to use backdoor attacks for positive purposes has also obtained some preliminary explorations. Adi et al. [106] adopted backdoor attacks in defending against model stealing via ownership verification. Specifically, they proposed to watermark the DNNs through backdoor embedding, which can be used to examine the model ownership. However, a recent study [107] revealed that this approach could fail, especially when it is complicated, since the stealing process may change or even remove hidden backdoors contained in the victim models. Besides, Sommer et al. [108] discussed how to verify whether the server truly erases their data when users require data deletion through poisoning-based backdoor attacks. Specifically, under their settings, each user can poison part of its data with a specific trigger and target label. Accordingly, each user can leave a unique trace in the server for deletion verification after the server is trained on user data. Besides, Shan et al. [109] introduced a trapdoor-enabled adversarial defense, where the hidden backdoor was injected by the defender to prevent attackers from discovering the real weakness in a model. The motivation was that the generated adversarial perturbation towards an infected model will converge near the trapdoor pattern, which was easily detected by the defender. Moreover, Li et al. [110] 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 suspicious model. They proposed a hypothesis test based method for the verification, based on the posterior probability of the benign samples and their attacked version generated by the suspicious model. Most recently, backdoor attacks were also adopted for interpreting DNNs [111] and the evaluation of explainable AI methods [112].

IV. Non-poisoning-based Backdoor Attacks

Except for poisoning-based backdoor attacks, recent literature also proposed some non-poisoning-based attacks. These methods embedded hidden backdoors not directly based on data poisoning during the training process. For example, attackers may directly change model weights or even the model structure without the training process. Their existence demonstrates that backdoor attacks could also happen at other stages ($e.g.$, deployment stage) instead of simply the data collection or training stages, which further reveals the severity of backdoor threats.

A. Weights-oriented Backdoor Attacks

In the weights-oriented backdoor attacks, attackers modified model parameters directly instead of through training with poisoned samples. To the best of our knowledge, Dumford et al. [19] proposed the first weights-oriented attack where they adopted a greedy search across models with different perturbations applied to a pre-trained model's weights. It is also the first non-poisoning-based backdoor attack. After that, Rakin et al. [20] introduced a bit-level weights-oriented backdoor attack, $i.e.$, the targeted bit trojan (TBT), which flipped critical bits of weights stored in the memory. The proposed method achieved remarkable performance, where attackers were able to mislead ResNet-18 [113] on the CIFAR-10 dataset [114] with 84 bit-flips out of 88 million weight bits. A similar idea was also introduced in [21], where attackers can significantly reduce the required flipping bits to embed hidden backdoors. Besides, Garg et al. [52] proposed to add adversarial perturbations on the model parameters of the benign model for injecting backdoors, showing a novel security threat of using publicly available trained models. Most recently, Zhang et al. [115] formulated the behavior of maintaining accuracy on benign samples as the consistency of infected models and provided a theoretical explanation of the adversarial weight perturbation (AWP) in backdoor attacks. Based on the analysis, they also introduced a new AWP-based backdoor attack with better global and instance-wise consistency.

Different from previous approaches where the backdoor is embedded in the parameters directly, Guo et al. [116] proposed TrojanNet to encode the backdoor in the infected DNNs activated through a secret weight permutation. Specifically, 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.

B. Structure-modified Backdoor Attacks

Structure-modified backdoor attacks injected hidden backdoors into benign models by changing their model structures.

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

Attack Category	Attacker's Goals	Attack Mechanism	Training Capacities	Inference Capacities
Backdoor Attack	Misclassify (modified) attacked samples; Behave normally on benign samples.	Excessive learning ability of models.	Under control.	Out of control.
Adversarial Attack	Misclassify (modified) attacked samples; Behave normally on benign samples.	Behavior differences between models and humans.	Out of control.	Query the model multiple times to generate adversarial perturbations by optimization.
Classical Data Poisoning	Reduce model generalization.	The sensitiveness of training process.	Can only modify the training set.	Out of control.
Advanced Data Poisoning	Misclassify (unmodified) targeted samples; Behave normally on benign samples.	Excessive learning ability of models.	Can only modify the training set.	Out of control.

These attacks could happen when using third-party models or in the deployment stage. To the best of our knowledge, Tang et al. [22] proposed the first structure-modified attack, where they inserted a trained malicious backdoor module (*i.e.*, a sub-DNN) into the target model for embedding hidden backdoors. This attack was simple yet effective and the malicious module can be combined with all DNNs. A similar idea was also explored in [23], where attackers embedded malicious conditional logics into the target DNNs by adding malicious payload containing the conditional module and the trigger detector. Most recently, Qi et al. [24] proposed to directly replace instead of adding a narrow subnet of the benign model to conduct the backdoor attack. This method is effective in both digital and physical scenarios.

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

Both adversarial attacks and backdoor attacks modify the benign testing samples to make models misbehave during the inference process. Especially when the adversarial perturbations are sample-agnostic in universal adversarial attacks [38], [117], [118], these attacks seem to be the same. As such, researchers who are not familiar with the backdoor attack may question its research significance for it requires additional controls of the training process to some extent.

However, these attacks still have essential differences although they enjoy certain similarities. **(1)** From the aspect of the attacker's capacity, adversarial attackers need to control the inference process (to a certain extent) but not the training process of models. Specifically, they need to query the model results or even gradients multiple times to generate adversarial perturbations by optimization given a fixed targeted model. In contrast, backdoor attackers require to modify some training stages (*e.g.*, data collection, model training) without any additional requirements in the inference process. **(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 [160], [161], [162]. As such, adversarial attacks are unable to be real-time in many cases for the optimization process takes time. **(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 to build a latent connection between the trigger patterns and the target labels.

Most recently, there were also a few works studying the latent connection between adversarial attacks and backdoor attacks. For example, Weng et al. [163] empirically demonstrated that defending against adversarial attacks via adversarial training may increase the risks of backdoor attacks.

B. Backdoor Attacks and Data Poisoning

In general, there are two types of data poisoning, including the classical and the advanced one. The former one intends to reduce model generalization, *i.e.*, letting the infected models behave well on training samples whereas having bad performance on testing samples. In contrast, advanced data poisoning makes infected models behave well on testing samples whereas having bad performance on some attacker-specified targeted samples which are not contained in the training set.

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 many intrinsic differences.

Firstly, compared with classical data poisoning, backdoor attacks preserve the performance of predicting benign samples. In other words, backdoor attacks have different attacker's goals compared with classical data poisoning. Besides, these attacks have different mechanisms. Specifically, the effectiveness of classical data poisoning is mostly due to the sensitiveness of the training process so that even a small domain shift of training samples may lead to significantly different decision surfaces of infected models. Moreover, backdoor attacks are also more stealthy than classical data poisoning. Users can easily detect classical data poisoning by evaluating the performance of trained models on a local verification set, while this method has limited benefits in detecting backdoor attacks. Secondly, backdoor attacks are also different from advanced data poisoning. Specifically, there is no trigger in advanced data poisoning, which does not require modifying targeted samples during the inference process. Correspondingly, advanced data poisoning can only misclassify (a few) specific samples, which limits its threats in many scenarios.

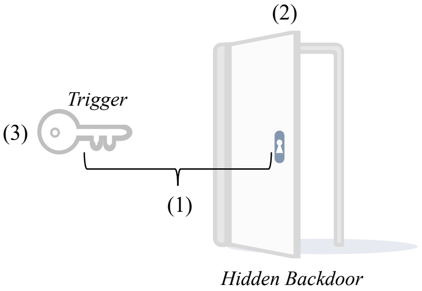

Fig. 5. An illustration of backdoor attacks and three corresponding defense paradigms. Intuitively, the poisoning-based backdoor attack is similar to unlock a door with the corresponding key. Accordingly, three main paradigms, including **(1)** trigger-backdoor mismatch, **(2)** backdoor elimination, 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 particular, since some literature proposed different types of defenses simultaneously, they will appear multiple times in this table.

Defense Paradigm	**Defense Sub-category**	**Literarure**
Trigger-backdoor Mismatch	Preprocessing-based Defenses	[119], [120], [121], [122] [9], [123]
Backdoor Elimination	Model Reconstruction based Defenses	[119], [124], [125], [126] [127], [128], [129]
	Trigger Synthesis based Defenses	[25], [56], [130], [57] [131], [132], [133], [134] [135], [58], [59], [136] [137]
	Model Diagnosis based Defenses	[60], [138], [26], [139] [140], [141], [142]
	Poison Suppression based Defenses	[143], [144], [145], [146] [27], [147]
	Training Sample Filtering based Defenses	[148], [149], [150], [151] [152], [61], [153], [154] [155]
Trigger Elimination	Testing Sample Filtering based Defenses	[156], [157], [143], [158] [159]

In particular, the studies of existing data poisoning have also inspired the research on backdoor learning due to their similarities. For example, Hong et al. [144] 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 against poisoning-based attacks and can be divided into two main categories, including *empirical backdoor defenses* and *certified backdoor defenses*. Specifically, empirical defenses were proposed based on some understandings of existing attacks and had decent performances 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* [164], 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. 5. 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: These methods introduce a preprocessing module before feeding samples into DNNs to change trigger patterns contained in attacked samples. Accordingly, the modified triggers no longer match the hidden backdoor and therefore preventing backdoor activation.

Liu et al. [119] proposed the first preprocessing-based backdoor defense, where they adopted a pre-trained auto-encoder as the preprocessor. Inspired by the idea that the trigger regions contributed most to the prediction, Doan et al introduced a two-stage image preprocessing approach ($i.e.$, Februus) in [120]. At the first stage, Februus used GradCAM [165] to identify influential regions, which will then be removed and replaced by a neutralized-color box. After that, Februus adopted a GAN-based inpainting method to reconstruct the masked regions to alleviate its adverse effects ($e.g.$, benign accuracy drop). After that, Udeshi el al. [121] used 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 understanding that placing a trigger blocker at the position of trigger patterns in attacked images will significantly change model predictions. Vasquez et al. [122] proposed to preprocess the image through style transfer. Recently, Li et al. [9] discussed the property of existing poisoning-based attacks with static trigger patterns. They demonstrated that if the *appearance* or *location* of the trigger is slightly changed, 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 explored in [123], where they introduced and evaluated more transformations in both fine-tuning and inference processes.

2) Model Reconstruction based Defenses: Different from preprocessing-based defenses, model reconstruction based methods aim at removing hidden backdoors in the infected model by modifying suspicious models directly. As such, even if the trigger is contained in attacked samples, the reconstructed model will still correctly predict them since the hidden backdoors were already removed.

Liu et al. [119] proposed to retrain the trained suspicious model with some local benign samples to reduce backdoor threats. Its effectiveness is mostly due to the *catastrophic* *forgetting* of DNNs [166], $i.e.$, the hidden backdoor is gradually removed as the training goes since the retraining set contains no poisoned samples. This idea was further explored by Zeng et al. [129], where they formulated the retraining as a mini-max problem and adopted the implicit hyper-gradients to account for the interdependence between inner and outer optimization. Motivated by the observation that the backdoor-related neurons are usually dormant when predicting benign samples, Liu et al. [124] proposed to prune those neurons to remove the hidden backdoor. Specifically, 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. A similar idea was further explored in [128], where they used adversarial weight perturbation to amplify the differences between benign and malicious neurons. In [125], Zhao et al. showed that the hidden backdoor of infected DNNs can be repaired based on the *mode connectivity* technique [167] with a certain amount of benign samples. Most recently, Yoshida et al. [126] and Li et al. [127] adopted *knowledge distillation* technique [168] to reconstruct (infected) DNNs, based on the understanding that the distillation process perturbs backdoor-related neurons and therefore can remove hidden backdoors.

3) Trigger Synthesis based Defenses: Except for eliminating hidden backdoors directly, trigger synthesis based defenses first synthesize the backdoor trigger, followed by the second stage that the hidden backdoor is eliminated by suppressing trigger's effects. These defenses enjoy certain similarities with 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 reconstruction-based defenses, the trigger information obtained in synthesis-based defenses makes the removal process more effective and efficient.

To the best of our knowledge, Wang et al. [25] proposed the first trigger synthesis based defense ($i.e.$ Neural Cleanse), where defenders first obtained potential trigger patterns towards every class and then determined the final synthetic trigger and its target label based on anomaly detection. A similar idea was also discussed in [56], [134], [135], [142] where they designed different trigger reversion or detection techniques. Qiao et al. [130] noticed that the reversed trigger synthesized by Neural Cleanse 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 infected models will generalize their original triggers during the training process. Accordingly, they proposed to recover the trigger distribution rather than a specific trigger for the defense, based on a max-entropy staircase approximator. A similar idea was also discussed in [131], where they proposed a GAN-based method to synthesize trigger distribution. In [57], they showed that the detection process used for determining the synthetic trigger in [25] suffers from several failure modes, based on which they proposed a new defense method. Besides, Cheng et al. [132] 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 to remove neurons with high activation values in response to the trigger. Similarly, Aiken et al. [133] also proposed to remove the hidden backdoor by pruning DNNs based on the synthetic trigger from another perspective. Moreover, Shen et al. [58] proposed an efficient trigger synthesis based defense. Different from previous defenses, which needed to generate all potential triggers towards each class, this defense selects only one class for trigger optimization in each round, inspired by the K-Arm bandit [169]. Recently, Hu et al. [137] designed a topological prior to improve the quality of trigger synthesis. Note that all previous defenses are white-box, requiring defenders have the access to model source files. Most recently, a few black-box synthesis-based defenses [59], [136] were also proposed, where defenders can reverse trigger patterns even when they can only obtain model predictions ($e.g.$, probability vectors or predicted labels).

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

To the best of our knowledge, Kolouri el al. [26] first discussed how to diagnose a given model. Specifically, they jointly optimized some universal litmus patterns (ULPs) and a meta-classifier, which was further used to diagnose suspicious models based on the predictions of obtained ULPs. Different from the previous defense where both infected models and benign models are required to train the meta-classifier, an effective meta-classifier can be trained only on benign models based on the strategies proposed in [138]. Besides, motivated by the observation that the heatmaps from benign and infected models have different characteristics, Huang et al. [60] adopted an outlier detector as the meta-classifier based on three extracted features of generated saliency maps. In [139], they designed an one-pixel signature representation, based on which to distinguish benign and infected models. Besides, Wang et al. [140] discussed how to detect whether a given mode is benign or infected in the data-limited and data-free cases. Most recently, [141] revealed that benign models and infected DNNs have significant topologically structural differences, which can be used to diagnose suspicious models.

5) Poison Suppression based Defenses: These defenses depress the effectiveness of poisoned samples during the training process to prevent the creation of hidden backdoors. Du et al. [143] first explored poison suppression based defenses, where they adopted noisy SGD to learn differentially private DNNs for the defense. With the randomness in the training process, the malicious effects of poisoned samples were reduced by random noise, preventing backdoor creation. Motivated by the observation that the ℓ^2 norm of the gradients of poisoned samples have significantly higher magnitudes than those of benign samples and their gradient orientations are also different, Hong et al. [144] adopted differentially private stochastic gradient descent (DPSGD) to clip and perturb individual gradients during the training process. Accordingly, the trained model had no hidden backdoor as well as its robustness towards targeted adversarial attacks was also increased. Besides, Borgain et al. [145] revealed that introducing strong data augmentation methods (*e.g.*, CutMix [170]) can effectively prevent the creation of hidden backdoors for they significantly perturbed the trigger patterns during the training process. Li et al. [27] proposed a gradient ascent based anti-backdoor method, based on the observations that backdoor attacks have faster learning on poisoned data and target-class dependency. Most recently, Huang et al. [147] revealed that the hidden backdoors are learned mostly due to the end-to-end supervised training paradigm, based on which they proposed a simple yet effective decoupling-based training method for backdoor suppression.

6) Training Sample Filtering based Defenses: These defenses aim at filtering poisoned samples from the training dataset. After the filtering process, only benign samples or purified poisoned samples will be used in the training process, which eliminates backdoor creation from the source.

To the best of our knowledge, Tran et al. [148] first explored how to filter malicious samples from the training set. Specifically, they demonstrated that poisoned samples tend to leave behind a detectable trace in the spectrum of the covariance of feature representations, which can be used to filter poisoned samples from the training set. Recently, Hayase et al. [153] introduced robust covariance estimation to amplify the spectral signature of poisoned samples, based on which they designed a more effective filtering method (*i.e.*, SPECTRE). Also inspired by the idea that poisoned samples and benign samples should have different characteristics in the hidden feature space, Chen et al. [149] proposed a two-stage filtering method, including **(1)** clustering the activations of training samples in each class into two clusters and **(2)** determining which, if any, of the clusters corresponds to poisoned samples. A similar idea was also explored in [151]. However, Tang et al. [150] demonstrated that simple target contamination can cause the representation of poisoned samples to be less distinguishable from that of benign ones, therefore most of existing filtering-based defenses can be easily bypassed. To address this problem, they proposed a more robust sample filter, based on representation decomposition and its statistical analysis. Different from previous methods, Chan et al. [152] separated poisoned samples based on signals contained in input gradients. A similar idea was explored in [61], where they adopted the saliency map to identify trigger regions and filter poisoned samples. Besides, Wang et al. [154] formulated the filtering as optimal data selection, based on which they proposed a unified framework to filter different types of malicious training samples. Most recently, Zeng et al. [155] revealed that poisoned samples of existing attacks had some high-frequency artifacts even if their trigger patterns are invisible in the input space. Based on this observation, they designed a simple yet effective filtering method based on those artifacts.

7) Testing Sample Filtering based Defenses: These defenses also filter malicious samples, whereas the filtering happened in the inference instead of the training process. Only benign testing or purified attacked samples will be predicted by the deployed model. These defenses prevent backdoor activation for they can remove trigger patterns.

Motivated by the observation that most of the existing backdoor triggers are input-agnostic, Gao et al. [156] proposed to filter attacked samples via superimposing various image patterns on the suspicious samples. The smaller the randomness among the predictions of perturbed inputs, the higher the probability that the suspicious sample is attacked. In [157], Subedar et al. adopted model uncertainty to distinguish between benign and attacked samples. After that, Du et al. [143] treated the filtering as outlier detection, based on which they proposed a differential privacy based filtering method. Besides, Jin et al. [158] proposed to detect attacked samples based on existing detection-based adversarial defenses [171], [172], [173]. Most recently, a lightweight method was proposed in [159], which can filter attacked samples without labeled samples or prior assumptions on trigger patterns.

B. Certified Backdoor Defenses

Although multiple empirical defenses have been proposed and reached decent performance against some backdoor attacks, almost all of them were bypassed by following adaptive attacks [188], [189]. To terminate this ‘cat-and-mouse chasing game’, Wang et al. [28] took the first step towards the certified defense against backdoor attacks based on the *random smoothing* technique [164]. Randomized smoothing was originally developed to certify robustness against adversarial examples, where the smoothed function was 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 [190], 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 [29], Weber et al. demonstrated that directly applying randomized smoothing, as in [28], will not provide high certified robustness bounds. Instead, they proposed a unified framework with the examination of different smoothing noise distributions and provided a tightness analysis for the robustness bound. Most recently, a few studies [191], [192], [193] also adopted ensemble techniques (*e.g.*, Bagging [194]) in designing certified defenses to further improve effectiveness.

C. Evaluation Metrics

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 a suspicious object (*e.g.*, a trained DNN or sample) is malicious. This is essentially a binary classification problem. To evaluate their performance, three metrics [195], including **(1)** precision, **(2)** recall, and **(3)** F1-score, are usually adopted. The higher the precision, recall, and F1-score, the better the defense performance.

Metrics for Non-detection-like Empirical Defenses. Except for model diagnosis based and testing sample filtering based defenses, all other methods are non-detection-like. Their main target is to achieve correct predictions for both benign and attacked samples. Accordingly, both benign accuracy and

TABLE V
Summary of benchmark datasets used in image recognition.

Category	Datasets	# Image Size	# Training Samples	# Testing Samples	Cited Literature
Natural Image Recognition	MNIST [174]	28 × 28	60,000	10,000	[119], [19], [60], [138], [121] [25], [149], [7], [56], [156] [157], [37], [175], [99], [108] [133], [26], [143], [28], [29] [151], [158], [109], [131], [176] [41], [62], [139], [159], [126] [39], [40], [45], [146], [142]
	Fashion MNIST [177]	28 × 28	60,000	10,000	[139], [144], [151], [142]
	CIFAR [114]	32 × 32 × 3	50,000	10,000	[106], [148], [13], [11], [120] [130], [138], [156], [157], [152] [37], [14], [52], [46], [108] [133], [26], [178], [144], [29] [176], [151], [116], [20], [9] [131], [134], [140], [109], [125] [41], [62], [110], [55], [135] [123], [43], [127], [39], [40] [44], [45], [48], [50], [63] [21], [59], [107], [111], [115] [24], [128], [136], [145], [146] [27], [153], [154], [155], [142]
	SVHN [179]	32 × 32 × 3	73,257	26,032	[125], [116], [20], [21]
	ImageNet [180]	224 × 224 × 3	1,281,167	50,000	[106], [57], [150], [42], [8] [14], [61], [29], [176], [22] [20], [131], [139], [140], [43] [12], [58], [39], [40], [45] [48], [63], [21], [59], [64] [107], [112], [111], [115], [23] [136], [27], [142]
Traffic Sign Recognition	GTSRB [181]	—	34,799	12,630	[11], [120], [56], [132], [60] [25], [57], [156], [150], [37] [8], [122], [26], [178], [176] [116], [22], [134], [158], [109] [110], [131], [62], [139], [140] [126], [123], [159], [41], [43] [127], [39], [40], [44], [45] [59], [63], [27], [155]
	U.S. Traffic Sign [182]	—	6,889	1,724	[124], [7], [121]
Face Recognition	YouTube Face [183]	—	3,425 videos of 1,595 people		[10], [124], [25], [178], [22] [109], [55]
	PubFig [184]	—	58,797 images of 200 people		[25], [8], [22], [158], [123] [45], [154], [155]
	VGGFace [185]	—	2.6 million images of 2,622 people		[51], [121], [56], [25], [122] [17], [61], [131], [159], [43] [63], [64], [24]
	VGGFace2 [186]	—	3.3 million images of 9,131 people		[19], [120], [64]
	LFW [187]	—	13,233 images of 5,749 people		[57], [17], [61]

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.

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.

Metrics for Certified Defenses. As mentioned in Section VI-B, existing certified backdoor defenses all adopted random smoothing. As such, 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 [28], [29]. 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 defense performance.

VII. Benchmark Datasets and Open-sourced Backdoor Toolboxes

Similar to that of adversarial learning and data poisoning, most of the existing backdoor-related literature focused on image classification tasks. In this section, we summarize all classical image classification benchmark datasets in Table V. Specifically, these benchmark datasets can be divided into three main categories, including *natural image recognition*, *traffic sign recognition*, and *face recognition*. The former ones are classical in the image classification, while the second and third ones are tasks requiring strict security guarantees. We recommend that future studies should be evaluated on these datasets to facilitate comparison and ensure fairness.

Currently, there are also a few open-sourced toolboxes aiming to provide the implementation of representative and advanced backdoor attacks and defenses. Specifically, both `TorjanZoo`[2] and `BackdoorBox`[3] provided methods under the centralized learning paradigm. In general, `TorjanZoo` is more comprehensive for it included more methods, while `BackdoorBox` is more flexible and user-friendly since all its methods were developed in a unified manner and its attack and defense modules can be used jointly or separately. Different from the previous two toolboxes, `Backdoors101`[4] mainly focused on methods under the federated learning. However, there were only a few attacks and defenses contained in the `Backdoors101`. In particular, `BackdoorBox` is still keep updating to track the latest backdoor attacks and defenses.

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 most existing attacks was designed in a heuristic ($e.g.$, design with universal perturbation) or even a non-optimized way. How to better optimize the trigger pattern ($e.g.$, based on bi-level optimization) is still an important open question. Besides, only the effectiveness and trigger invisibility were considered in the trigger design. Other criteria, such as minimal poisoning rate and trigger generalization, 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 left far behind compared to other types of backdoor attacks. More thorough studies to obtain better understandings of these attacks would be important steps towards alleviating the backdoor threats in practice. For example, one may explore whether other physical phenomena ($e.g.$, specific illumination), can also serve as effective physical trigger patterns and why semantic triggers are also effective.

C. Attacks Towards Other Tasks

The success of backdoor attacks heavily relied on the trigger design 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 stealthiness. However, the design of backdoor triggers in different tasks could be quite different ($e.g.$, hiding a trigger into a sentence when attacking NLP-related tasks is quite different from hiding a trigger into an image). As such, it is necessary to study task-specified backdoor attacks. Currently, existing backdoor attacks mainly focused on computer vision tasks, especially image classification. The research towards other tasks ($e.g.$, recommendation system, speech recognition, and natural language processing) have not been well studied. Besides, regression as another important paradigm deserves more attention and backdoor-related exploration.

D. Effective and Efficient Defenses

Although many types of empirical backdoor defenses have been proposed (as demonstrated in Section VI), almost all of them can be bypassed by subsequent adaptive attacks. Besides, except for the preprocessing-based defenses, existing defenses usually suffer from high computational costs. More efforts on designing effective and efficient defenses ($e.g.$, analyzing the weaknesses of existing attacks and how to reduce the computational costs of defenses) should be made to keep up the fast pace of backdoor attacks. For example, it would be possible to locate the trigger patterns or malicious hidden features based on Explainable AI (XAI) methods. Besides, how to design black-box defenses is also worth more attention since these methods are more practical in reality. 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 backdoor learning. For example, why hidden backdoors can be created and what happens inside the infected models when the trigger appears have not been carefully studied in existing works. A deeper understanding of the intrinsic mechanism of backdoor attacks can guide the design of more effective attacks and defenses, and the understandings of DNN's behaviors.

IX. Conclusion

Backdoor learning, including backdoor attacks and backdoor defenses, is a critical and booming research area. In this survey, we summarized and categorized existing backdoor attacks and proposed a unified framework for analyzing poisoning-based backdoor attacks. Specifically, we proposed seven different criteria for classifying existing attacks. Among all these attacks, we believed that semantic, sample-specific, and black-box poisoning-based backdoor attacks are worth more attention since they are more threatening in practice. We also discussed the relation between backdoor attacks and related research areas ($i.e.$, data poisoning and adversarial attacks) and analyzed existing defenses. Specifically, we identified three fundamental defense paradigms and divided existing defenses into eight main categories. Classical benchmark datasets and potential research directions were illustrated at the end. Note that almost all studies in this field were completed

[2] https://github.com/ain-soph/trojanzoo

[3] https://github.com/THUYimingLi/BackdoorBox

[4] https://github.com/ebagdasa/backdoors101

in the last four years and the cat-and-mouse game between attacks and defenses is likely to continue in the future. We hope that this survey could remind researchers of backdoor threats and provide a timely view. It would be an important step towards building more robust and secure DNNs.

Acknowledgements

This work was partly done when Yiming Li was a research 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 Natural Science Foundation of China under grant 62171248, the Guangdong Province Key Area R&D Program under Grant 2018B010113001, the R&D Program of Shenzhen (JCYJ20180508152204044), the PCNL KEY project (PCL2021A07), and the Shenzhen Philosophical and Social Science Plan (SZ2020D009). We would also sincerely thank Dr. Baoyuan Wu from the Chinese University of Hong Kong (Shenzhen) and Dr. Bo Li from the University of Illinois Urbana-Champaign for their helpful comments and suggestions 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] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *ICLR*, 2018.

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

[5] Y. Bai, Y. Zeng, Y. Jiang, S.-T. Xia, X. Ma, and Y. Wang, "Improving adversarial robustness via channel-wise activation suppressing," in *ICLR*, 2021.

[6] Y. Li, B. Wu, Y. Feng, Y. Fan, Y. Jiang, Z. Li, and S.-T. Xia, "Semi-supervised robust training with generalized perturbed neighborhood," *Pattern Recognition*, vol. 124, p. 108472, 2022.

[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, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor attack in the physical world," in *ICLR Workshop*, 2021.

[10] 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.

[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] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, "Invisible backdoor attack with sample-specific triggers," in *ICCV*, 2021.

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

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

[15] 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.

[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] Y. Ge, Q. Wang, B. Zheng, X. Zhuang, Q. Li, C. Shen, and C. Wang, "Anti-distillation backdoor attacks: Backdoors can really survive in knowledge distillation," in *ACM MM*, 2021.

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

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

[21] H. Chen, C. Fu, J. Zhao, and F. Koushanfar, "Proflip: Targeted trojan attack with progressive bit flips," in *ICCV*, 2021.

[22] 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.

[23] Y. Li, J. Hua, H. Wang, C. Chen, and Y. Liu, "Deeppayload: Black-box backdoor attack on deep learning models through neural payload injection," in *ICSE*, 2021.

[24] X. Qi, J. Zhu, C. Xie, and Y. Yang, "Subnet replacement: Deployment-stage backdoor attack against deep neural networks in gray-box setting," in *ICLR Workshop*, 2021.

[25] 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.

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

[27] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Anti-backdoor learning: Training clean models on poisoned data," in *NeurIPS*, 2021.

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

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

[30] C. Xie, M. Chen, P.-Y. Chen, and B. Li, "Crfl: Certifiably robust federated learning against backdoor attacks," in *ICML*, 2021.

[31] Y. Liu, A. Mondal, A. Chakraborty, M. Zuzak, N. Jacobsen, D. Xing, and A. Srivastava, "A survey on neural trojans," in *ISQED*, 2020.

[32] M. Goldblum, D. Tsipras, C. Xie, X. Chen, A. Schwarzschild, D. Song, A. Madry, B. Li, and T. Goldstein, "Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[33] S. Kaviani and I. Sohn, "Defense against neural trojan attacks: A survey," *Neurocomputing*, vol. 423, pp. 651–667, 2021.

[34] Y. Gao, B. G. Doan, Z. Zhang, S. Ma, J. Zhang, A. Fu, S. Nepal, and H. Kim, "Backdoor attacks and countermeasures on deep learning: A comprehensive review," *arXiv preprint arXiv:2007.10760*, 2020.

[35] S. Li, S. Ma, M. Xue, and B. Z. H. Zhao, "Deep learning backdoors," *arXiv preprint arXiv:2007.08273*, 2020.

[36] W. Guo, B. Tondi, and M. Barni, "An overview of backdoor attacks against deep neural networks and possible defences," *arXiv preprint arXiv:2111.08429*, 2021.

[37] 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.

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

[39] K. Doan, Y. Lao, W. Zhao, and P. Li, "Lira: Learnable, imperceptible and robust backdoor attacks," in *ICCV*, 2021.

[40] K. Doan, Y. Lao, and P. Li, "Backdoor attack with imperceptible input and latent modification," in *NeurIPS*, 2021.

[41] T. A. Nguyen and A. T. Tran, "Wanet-imperceptible warping-based backdoor attack," in *ICLR*, 2021.

[42] E. Bagdasaryan and V. Shmatikov, "Blind backdoors in deep learning models," in *USENIX Security*, 2021.

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

[44] H. A. A. K. Hammoud and B. Ghanem, "Check your other door! establishing backdoor attacks in the frequency domain," *arXiv preprint arXiv:2109.05507*, 2021.

[45] T. Wang, Y. Yao, F. Xu, S. An, and T. Wang, "Backdoor attack through frequency domain," *arXiv preprint arXiv:2111.10991*, 2021.

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

[47] 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.

[48] H. Souri, M. Goldblum, L. Fowl, R. Chellappa, and T. Goldstein, "Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch," *arXiv preprint arXiv:2106.08970*, 2021.

[49] R. Liu, J. Gao, J. Zhang, D. Meng, and Z. Lin, "Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond," *arXiv preprint arXiv:2101.11517*, 2021.

[50] I. Shumailov, Z. Shumaylov, D. Kazhdan, Y. Zhao, N. Papernot, M. A. Erdogdu, and R. Anderson, “Manipulating sgd with data ordering attacks,” in *NeurIPS*, 2021.

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

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

[53] J. Geiping, L. H. Fowl, W. R. Huang, W. Czaja, G. Taylor, M. Moeller, and T. Goldstein, “Witches’ brew: Industrial scale data poisoning via gradient matching,” in *ICLR*, 2020.

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

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

[56] 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.

[57] 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.

[58] 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.

[59] Y. Dong, X. Yang, Z. Deng, T. Pang, Z. Xiao, H. Su, and J. Zhu, “Black-box detection of backdoor attacks with limited information and data,” in *ICCV*, 2021.

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

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

[62] A. Nguyen and A. Tran, “Input-aware dynamic backdoor attack,” in *NeurIPS*, 2020.

[63] J. Zhang, D. Chen, J. Liao, Q. Huang, G. Hua, W. Zhang, and N. Yu, “Poison ink: Robust and invisible backdoor attack,” *arXiv preprint arXiv:2108.02488*, 2021.

[64] E. Wenger, J. Passananti, A. N. Bhagoji, Y. Yao, H. Zheng, and B. Y. Zhao, “Backdoor attacks against deep learning systems in the physical world,” in *CVPR*, 2021.

[65] 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.

[66] X. Chen, A. Salem, D. Chen, M. Backes, S. Ma, Q. Shen, Z. Wu, and Y. Zhang, “Badnl: Backdoor attacks against nlp models with semantic-preserving improvements,” in *ACSAC*, 2021.

[67] F. Qi, Y. Yao, S. Xu, Z. Liu, and M. Sun, “Turn the combination lock: Learnable textual backdoor attacks via word substitution,” in *ACL*, 2021.

[68] F. Qi, M. Li, Y. Chen, Z. Zhang, Z. Liu, Y. Wang, and M. Sun, “Hidden killer: Invisible textual backdoor attacks with syntactic trigger,” in *ACL*, 2021.

[69] W. Yang, Y. Lin, P. Li, J. Zhou, and X. Sun, “Rethinking stealthiness of backdoor attack against nlp models,” in *ACL*, 2021.

[70] F. Qi, Y. Chen, X. Zhang, M. Li, Z. Liu, and M. Sun, “Mind the style of text! adversarial and backdoor attacks based on text style transfer,” in *EMNLP*, 2021.

[71] W. Yang, L. Li, Z. Zhang, X. Ren, X. Sun, and B. He, “Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in nlp models,” in *NAACL*, 2021.

[72] L. Li, D. Song, X. Li, J. Zeng, R. Ma, and X. Qiu, “Backdoor attacks on pre-trained models by layerwise weight poisoning,” in *EMNLP*, 2021.

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

[74] Z. Xi, R. Pang, S. Ji, and T. Wang, “Graph backdoor,” in *USENIX Security*, 2021.

[75] J. Chen, H. Xiong, H. Zheng, J. Zhang, G. Jiang, and Y. Liu, “Dyn-backdoor: Backdoor attack on dynamic link prediction,” *arXiv preprint arXiv:2110.03875*, 2021.

[76] G. Tian, W. Jiang, W. Liu, and Y. Mu, “Poisoning morphnet for clean-label backdoor attack to point clouds,” *arXiv preprint arXiv:2105.04839*, 2021.

[77] Z. Xiang, D. J. Miller, S. Chen, X. Li, and G. Kesidis, “A backdoor attack against 3d point cloud classifiers,” in *ICCV*, 2021.

[78] X. Li, Z. Chen, Y. Zhao, Z. Tong, Y. Zhao, A. Lim, and J. T. Zhou, “Pointba: Towards backdoor attacks in 3d point cloud,” in *ICCV*, 2021.

[79] Z. Yan, J. Wu, G. Li, S. Li, and M. Guizani, “Deep neural backdoor in semi-supervised learning: Threats and countermeasures,” *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4827–4842, 2021.

[80] J. Jia, Y. Liu, and N. Z. Gong, “Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning,” in *IEEE S&P*, 2022.

[81] N. Carlini and A. Terzis, “Poisoning and backdooring contrastive learning,” in *ICLR*, 2022.

[82] P. Kiourti, K. Wardega, S. Jha, and W. Li, “Trojdrl: evaluation of backdoor attacks on deep reinforcement learning,” in *DAC*, 2020.

[83] L. Wang, Z. Javed, X. Wu, W. Guo, X. Xing, and D. Song, “Backdoorl: Backdoor attack against competitive reinforcement learning,” in *IJCAI*, 2021.

[84] C. Ashcraft and K. Karra, “Poisoning deep reinforcement learning agents with in-distribution triggers,” in *ICLR Workshop*, 2021.

[85] H. Ma, H. Qiu, Y. Gao, Z. Zhang, A. Abuadbba, A. Fu, S. Al-Sarawi, and D. Abbott, “Quantization backdoors to deep learning models,” *arXiv preprint arXiv:2108.09187*, 2021.

[86] S. Hong, M.-A. Panaitescu-Liess, Y. Kaya, and T. Dumitras, “Qu-anti-zation: Exploiting quantization artifacts for achieving adversarial outcomes,” in *NeurIPS*, 2021.

[87] X. Pan, M. Zhang, Y. Yan, and M. Yang, “Understanding the threats of trojaned quantized neural network in model supply chains,” in *ACSAC*, 2021.

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

[89] S. Koffas, J. Xu, M. Conti, and S. Picek, “Can you hear it? backdoor attacks via ultrasonic triggers,” *arXiv preprint arXiv:2107.14569*, 2021.

[90] C. Li, X. Chen, D. Wang, S. Wen, M. E. Ahmed, S. Camtepe, and Y. Xiang, “Backdoor attack on machine learning based android malware detectors,” *IEEE Transactions on Dependable and Secure Computing*, 2021.

[91] G. Severi, J. Meyer, S. Coull, and A. Oprea, “Explanation-guided backdoor poisoning attacks against malware classifiers,” in *USENIX Security*, 2021.

[92] Y. Li, Y. Li, Y. Lv, Y. Jiang, and S.-T. Xia, “Hidden backdoor attack against semantic segmentation models,” in *ICLR Workshop*, 2021.

[93] S. Fang and A. Choromanska, “Backdoor attacks on the dnn interpretation system,” in *AAAI*, 2022.

[94] Y. Li, H. Zhong, X. Ma, Y. Jiang, and S.-T. Xia, “Few-shot backdoor attacks on visual object tracking,” in *ICLR*, 2022.

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

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

[97] 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.

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

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

[100] Y. Liu, Z. Yi, Y. Kang, Y. He, W. Liu, T. Zou, and Q. Yang, “Defending label inference and backdoor attacks in vertical federated learning,” in *AAAI*, 2022.

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

[102] M. S. Ozdayi, M. Kantarcioglu, and Y. R. Gel, “Defending against backdoors in federated learning with robust learning rate,” in *AAAI*, 2021.

[103] X. Liu, H. Li, G. Xu, Z. Chen, X. Huang, and R. Lu, “Privacy-enhanced federated learning against poisoning adversaries,” *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 4574–4588, 2021.

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

[105] K. Chen, Y. Meng, X. Sun, S. Guo, T. Zhang, J. Li, and C. Fan, “Badpre: Task-agnostic backdoor attacks to pre-trained nlp foundation models,” in *ICLR*, 2022.

[106] 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.

[107] Y. Li, L. Zhu, X. Jia, Y. Jiang, S.-T. Xia, and X. Cao, “Defending against model stealing via verifying embedded external features,” in *AAAI*, 2022.

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

[109] 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.

[110] 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.

[111] 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.

[112] Y.-S. Lin, W.-C. Lee, and Z. B. Celik, "What do you see? evaluation of explainable artificial intelligence (xai) interpretability through neural backdoors," in *KDD*, 2021.

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

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

[115] Z. Zhang, L. Lyu, W. Wang, L. Sun, and X. Sun, "How to inject backdoors with better consistency: Logit anchoring on clean data," *arXiv preprint arXiv:2109.01300*, 2021.

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

[117] 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.

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

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

[120] B. G. Doan, E. Abbasnejad, and D. C. Ranasinghe, "Februus: Input purification defense against trojan attacks on deep neural network systems," in *ACSAC*, 2020.

[121] 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.

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

[123] 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," in *AsiaCCS*, 2021.

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

[125] 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.

[126] 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.

[127] 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.

[128] D. Wu and Y. Wang, "Adversarial neuron pruning purifies backdoored deep models," in *NeurIPS*, 2021.

[129] Y. Zeng, S. Chen, W. P. Z. Morley Mao, M. Jin, and R. Jia, "Adversarial unlearning of backdoors via implicit hypergradient," in *ICLR*, 2022.

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

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

[132] 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.

[133] 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.

[134] 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.

[135] 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.

[136] J. Guo, A. Li, and C. Liu, "Aeva: Black-box backdoor detection using adversarial extreme value analysis," in *ICLR*, 2022.

[137] X. Hu, X. Lin, M. Cogswell, Y. Yao, S. Jha, and C. Chen, "Trigger hunting with a topological prior for trojan detection," in *ICLR*, 2022.

[138] 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.

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

[140] 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.

[141] S. Zheng, Y. Zhang, H. Wagner, M. Goswami, and C. Chen, "Topological detection of trojaned neural networks," in *NeurIPS*, 2021.

[142] Z. Xiang, D. J. Miller, and G. Kesidis, "Post-training detection of backdoor attacks for two-class and multi-attack scenarios," in *ICLR*, 2022.

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

[144] 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.

[145] E. Borgnia, V. Cherepanova, L. Fowl, A. Ghiasi, J. Geiping, M. Goldblum, T. Goldstein, and A. Gupta, "Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff," in *ICASSP*, 2021.

[146] X. Liu, F. Li, B. Wen, and Q. Li, "Removing backdoor-based watermarks in neural networks with limited data," in *ICPR*, 2021.

[147] K. Huang, Y. Li, B. Wu, Z. Qin, and K. Ren, "Backdoor defense via decoupling the training process," in *ICLR*, 2022.

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

[149] 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.

[150] 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.

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

[152] 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.

[153] J. Hayase and W. Kong, "Spectre: Defending against backdoor attacks using robust covariance estimation," in *ICML*, 2021.

[154] T. Wang, Y. Zeng, M. Jin, and R. Jia, "A unified framework for task-driven data quality management," *arXiv preprint arXiv:2106.05484*, 2021.

[155] Y. Zeng, W. Park, Z. M. Mao, and R. Jia, "Rethinking the backdoor attacks' triggers: A frequency perspective," in *ICCV*, 2021.

[156] 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.

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

[158] 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.

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

[160] 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.

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

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

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

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

[165] 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.

[166] 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.

[167] 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.

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

[169] 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.

[170] T. DeVries and G. W. Taylor, “Improved regularization of convolutional neural networks with cutout,” *arXiv preprint arXiv:1708.04552*, 2017.

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

[172] 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.

[173] 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.

[174] 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.

[175] 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.

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

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

[178] 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.

[179] 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.

[180] 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.

[181] 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.

[182] 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.

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

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

[185] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” in *BMVC*, 2015.

[186] 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.

[187] 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.

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

[189] A. Saha, A. Subramanya, and H. Pirsiavash, “Hidden trigger backdoor attacks,” in *AAAI*, 2020.

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

[191] J. Jia, X. Cao, and N. Z. Gong, “Intrinsic certified robustness of bagging against data poisoning attacks,” in *AAAI*, 2021.

[192] A. Levine and S. Feizi, “Deep partition aggregation: Provable defenses against general poisoning attacks,” in *ICLR*, 2021.

[193] J. Jia, X. Cao, and N. Z. Gong, “Certified robustness of nearest neighbors against data poisoning attacks,” in *AAAI*, 2022.

[194] L. Breiman, “Bagging predictors,” *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

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

Yiming Li is currently a Ph.D. candidate in Computer Science and Technology from Tsinghua Shenzhen International Graduate School, Tsinghua University, China. Before that, he received his B.S. degree in Mathematics and Applied Mathematics from Ningbo University, China, in 2018. His research interests are in the domain of AI security, especially backdoor learning, adversarial learning, and data privacy. His research has been published in multiple top-tier conferences and journals, such as ICCV, ECCV, ICLR, AAAI, PR Journal, and IEEE IoT Journal. He served as the senior program committee member of AAAI 2022, the program committee member of ICML, NeurIPS, ICLR, ECCV, etc., and the reviewer of IEEE TDSC, IEEE TCSVT, IEEE TII, etc.

Dr. 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. He has received several best paper awards (e.g., IWQoS 2018) and his researches have been published in multiple top-tier journals and conferences, including IEEE ToC, IEEE TMM, IEEE TSP, CVPR, ICLR, ECCV, etc.

Dr. Zhifeng Li is currently a top-tier principal researcher at Tencent Data Platform. 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 Pattern Recognition, IEEE Transactions on Circuits and Systems for Video Technology, and Neurocomputing. He is a fellow of the British Computer Society (FBCS).

Dr. 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, 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, China. His current research interests include coding and information theory, machine learning, and deep learning. His research has been published in multiple top-tier journals and conferences, including IEEE TPAMI, IEEE TIP, IEEE TNNLS, CVPR, ICCV, ICLR, etc.