# A Comprehensive Survey on Poisoning Attacks and Countermeasures in Machine Learning

ZHIYI TIAN, University of Technology Sydney, Australia
LEI CUI, Shandong Computer Science Center (National Supercomputer Center in Jinan), China
JIE LIANG and SHUI YU, University of Technology Sydney, Australia

The prosperity of machine learning has been accompanied by increasing attacks on the training process. Among them, poisoning attacks have become an emerging threat during model training. Poisoning attacks have profound impacts on the target models, e.g., making them unable to converge or manipulating their prediction results. Moreover, the rapid development of recent distributed learning frameworks, especially federated learning, has further stimulated the development of poisoning attacks. Defending against poisoning attacks is challenging and urgent. However, the systematic review from a unified perspective remains blank. This survey provides an in-depth and up-to-date overview of poisoning attacks and corresponding countermeasures in both centralized and federated learning. We firstly categorize attack methods based on their goals. Secondly, we offer detailed analysis of the differences and connections among the attack techniques. Furthermore, we present countermeasures in different learning framework and highlight their advantages and disadvantages. Finally, we discuss the reasons for the feasibility of poisoning attacks and address the potential research directions from attacks and defenses perspectives, separately.

CCS Concepts: • **Security and privacy**; • **Computing methodologies** → **Machine learning**; **Distributed computing methodologies**;

Additional Key Words and Phrases: Deep learning, federated learning, poisoning attack, backdoor attack

**ACM Reference format:**
Zhiyi Tian, Lei Cui, Jie Liang, and Shui Yu. 2022. A Comprehensive Survey on Poisoning Attacks and Countermeasures in Machine Learning. *ACM Comput. Surv.* 55, 8, Article 166 (December 2022), 35 pages.
https://doi.org/10.1145/3551636

## 1 INTRODUCTION

Machine (Deep) learning models have been widely adopted in critical domains, such as autonomous vehicles, identity recognition, and disease diagnosis. The prevalence of machine learning, especially deep learning, has drawn attention to their security issues [54, 94, 154]. Among them, poisoning attack is the most immediate threat against the training process of machine learning models. It is originally introduced in conventional machine learning, such as support vector



This work was partially supported by Australia ARC DP200101374 and LP190100676.
Authors' addresses: Z. Tian, J. Liang, and S. Yu, University of Technology Sydney, 15 Broadway, Ultimo, NSW, Australia, CRICOS Provider No: 00099F, Sydney, Australia; emails: zhiyi.tian@student.uts.edu.au, {Jie.Liang, shui.yu}@uts.edu.au; L. Cui, Shandong Computer Science Center (National Supercomputer Center in Jinan), 19 Keyuan Road, Jinan City, Shandong Province, China (250014), Jinan, China; email: alencui@outlook.com.

0360-0300/2022/12-ART166 $15.00
https://doi.org/10.1145/3551636

machines [10, 11]. The core idea of poisoning attacks is to introduce malicious data into training datasets of target models to hinder the model training [4, 5, 13]. The data-driven attacking intuition endows poisoning attack the transferability of attacking multiple machine learning models. Moreover, in recent years, federated learning has emerged as a new leading training architecture to respond to the concern of privacy leakage during model training. Although federated learning has made some contributions to protecting data privacy, the distributed architecture reduces the transparency of training data. Training dataset of a local client is inaccessible to other participants in the learning system, which makes the federated learning framework more vulnerable to poisoning attacks [8, 152].

The security challenges brought by poisoning attacks have prompted many researchers to devote themselves to the development of countermeasures. Existing countermeasures [13, 71] are largely attack-specific: they can only defend against several known attack methods, and once the adversary knows the existence of these countermeasures, it is easy to bypass them [8, 136]. Many reasons have led to the current disadvantage of defenders. For example, the development of countermeasures is often only based on some observations, rather than a global understanding of attack methods and learning algorithms. Therefore, to better counter poisoning attacks, a comprehensive and in-depth survey is needed.

Several surveys addressing security issues in centralized learning or federated learning have been published in recent years. Papernot et al. [93] discussed security and privacy issues in machine learning. They defined poisoning attacks as integrity attacks. Their survey introduced several distinguished poisoning attacks against conventional machine learning, while it ignored poisoning attacks against other learning methods, e.g., deep learning and federated learning. Pitropakis et al. [102] focused on poisoning attacks from the perspective of target models. They classified the existing attack methods based on the target models. However, the connection between the attack methods is missing in their survey. Backdoor attacks and corresponding countermeasures are carefully discussed in surveys [36, 66]. The main method of implementing a backdoor attack is to poison the dataset. Therefore, these surveys are overlapped with poisoning attacks. Several other surveys [7, 148] analyzed the security issues of decentralized learning from a system level.

Although there are many researchers that discussed poisoning attacks in their surveys, few of them took a unified perspective to examine security vulnerabilities of centralized learning and federated learning caused by poisoning attacks. Most current surveys are technique driven. They focus on security and privacy issues in a specific learning scenario, e.g., machine learning [93] and decentralized learning [7, 148]. Poisoning attacks are only one of the threats in these surveys, which makes them contribute less to improving the understanding of security vulnerabilities caused by poisoning attacks. As a data-driven threat, poisoning attacks are not sensitive to victim learning systems. Every time a new learning method is proposed, adversaries can quickly compromise it. We blame the unfavorable situation to the lack of a comprehensive understanding of security community. A comprehensive understanding of poisoning attacks will be helpful to guide the academia and industry to develop more robust machine learning methods.

The motivation of this survey is to fill this research gap by a consistent literature review on poisoning attacks and corresponding countermeasures. It intends to cover most victim learning systems, and provide a unified perspective to analyze the internal intuition of these poisoning attacks. Although the manifestations of poisoning attacks in centralized learning and federated learning seem to be different, their internal intuition is unified, which is to hinder model training by introducing poisoning samples into the training dataset. In this survey, poisoning attacks are classified by their adversarial goals, as shown in Figure 1, we focus on three kinds of poisoning attacks: **untargeted poisoning attacks**, **target poisoning attacks**, and **backdoor poisoning**

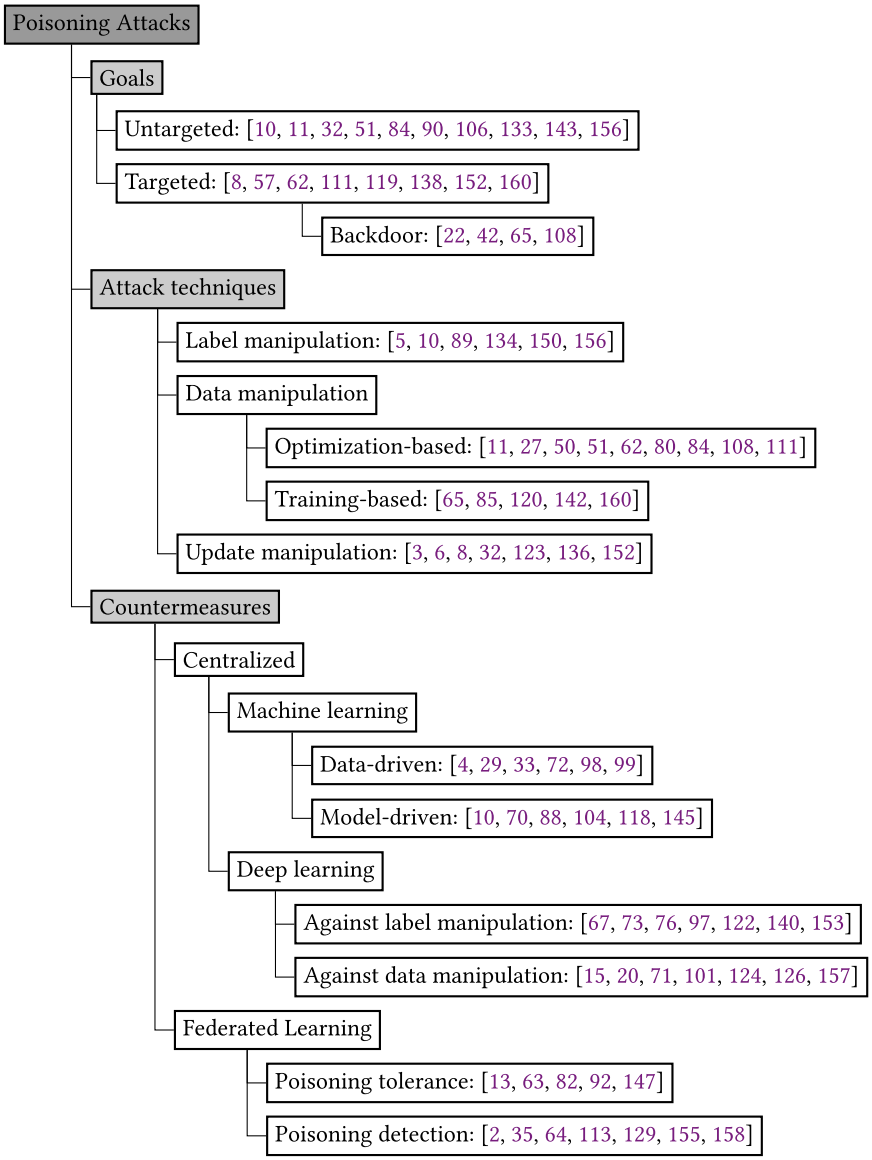


Fig. 1. Our taxonomy for poisoning attacks and countermeasures in centralized and federated learning.

**attacks** in both centralized learning and federated learning scenarios. Among them, the backdoor poisoning attack is a special case of the target poisoning attack. They share a similar adversarial goal with different stealthy requirements. Poisoning attacks against centralized learning are the basis of poisoning attacks against federated learning. They share the adversarial goals and some attack techniques, and only differ in the form of poisoning entity, according to different learning architectures. Therefore, we first introduce poisoning attacks against conventional machine learning models and deep learning models. On this basis, we introduce the research of poisoning attacks in the federated learning scenario in detail. Furthermore, we present a detailed review of countermeasures against poisoning attacks. Unlike attack methods, countermeasures are more integrated

with training methods. Therefore, we classify and analyze countermeasures based on the protected models. Although the protected models are different, there are some commonalities between these countermeasures.

The main contributions of this paper are as follows.

(1) We present a unified perspective of poisoning attacks in both centralized learning and federated learning, illustrate the connection among techniques used in different learning scenarios, and address the trade-off in efficiency and stealth of poisoning attacks in federated learning.

(2) We introduce the off-the-shelf countermeasures and classify them based on protected models. We further subdivide each class of countermeasures according to their strategies. Based on the taxonomy, the general idea of existing countermeasures and corresponding limitations are sorted out and revealed.

(3) We discuss the challenges that adversaries face in poisoning attacks. Adversaries can work towards improving the efficiency, stealth, and robustness of poisoning attacks. To better respond to the threat of poisoning attacks, we also present potential research directions for defenders. Defenders can provide a more comprehensive countermeasure by adopting multiple detection features.

The rest of this survey is organized as follows: The preliminary of poisoning attack is summarized in Section 2. In Section 3, we present the taxonomy of poisoning attacks, where poisoning attacks are reviewed in three categories classified by the adversarial goals. In Section 4, techniques of poisoning samples construction in centralized learning are in-depth introduced. Section 5 details the poisoning attack against federated learning. We discuss countermeasures in different learning frameworks in Section 6. In Section 7 we first discuss the potential reasons that account for the feasibility of poisoning attacks. Then, we list the open gaps that new studies can target. Finally, we conclude the survey in Section 8.

## 2 PRELIMINARY

### 2.1 Privacy and Security Attacks in Machine Learning

The initial idea of machine learning was proposed a few decades ago [77, 110]. However, it is only in recent years that learning algorithms are applied in wide domains, because of the progress of learning algorithms [16, 55, 61, 107] and computing power. Machine learning is not designed under privacy and security considerations, which leads machine learning methods vulnerable to adversarial attacks.

Privacy and security are two major kinds of threats to machine learning methods. The privacy attack could be defined as malicious attempts revealing training data, such as membership inference [109, 114, 125] and model inversion [34, 45, 146, 154, 161]. Adversaries infer the privacy of training data from the well-trained model or the federated learning training process by various means. This kind of attack has no impact on a model's availability.

The goal of security attacks is to hinder the legitimate function of target models. As shown in Figure 2, machine learning can be divided into two stages: training stage and inference stage. In the training stage, a model learns the knowledge from a training dataset collected from various sources. Then, in the inference stage, the well trained model is used to infer previous unseen inputs from the same domain [86]. The security attacks can be classified into two classes based on the target stage: the training stage attack and the inference stage attack.

In inference stage attack, the target model is a well-trained model. For example, in a Machine-Learning-as-a-Service systems [103], an adversary can generate adversarial examples to maliciously change corresponding predictions [41, 59]. In these attacks, adversaries cannot modify the target model. Therefore, each attack is relatively independent and results in no permanent impact on the victim model.

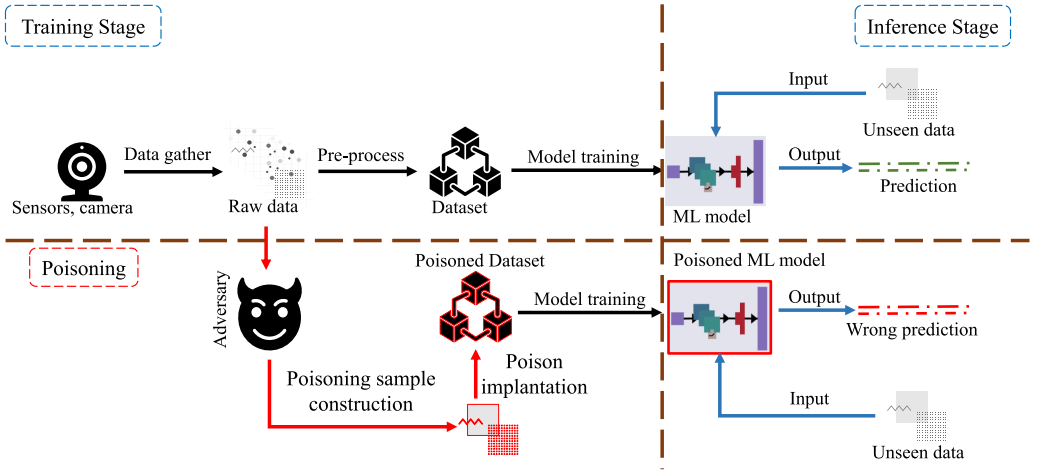


Fig. 2. Poisoning attack architecture. Above the dotted line is the normal machine learning life cycle, while the bottom half is the machine learning process under poisoning attack.

Different from attacks against well-trained models, attacks in the training stage can affect the target model's legitimate function or manipulate its behavior in the long term. Poisoning attack is a kind of training stage attack. In a poisoning attack, the adversary forces the target model to fail to converge or perform abnormally on specific inputs, through implanting carefully constructed poisoning samples into the training dataset. An adversary can control the behaviors of a well-trained model as long as the model is trained on a poisoned dataset [22, 108].

In addition, poisoning attacks are data-driven techniques. In common threat models, adversaries do not disrupt the legitimate function of the target training system. In contrast, the purpose of adversaries is to cause the victim training system to produce erroneous models after using poisoning data. Therefore, poisoning adversaries can be defined as **legitimate-but-malicious**. It means that adversaries follow the training protocol legitimately. However, adversaries try to implant poisoning samples in the training process to corrupt the trained model. Poisoning samples could be implanted through sensors in the form of raw data [22, 138] and feature files after pre-processing [10, 11]. Therefore, poisoning attack is essentially a data-driven technique, although it may involve software and hardware.

### 2.2 Overview of Centralized and Federated Learning: The Target

Machine learning can be divided into supervised learning and unsupervised learning based on whether the training dataset is labeled or not. The training samples of supervised learning are labeled with corresponding classes. The goal of supervised learning algorithm is to learn knowledge instructed by the sample-label pairs [24]. The training dataset of unsupervised learning contains only samples. Therefore, the learning algorithm needs to mine the class information by themselves [44]. In recent years, the security vulnerabilities of supervised learning attracts more attentions than unsupervised learning, due to their wide applications. Therefore, the main research object of this survey is poisoning attack in supervised learning scenario.

Due to privacy concerns, users are unwilling to share their private data. To protect users' privacy, federated learning is introduced [78]. There are two roles in a federated learning system: server and client. The task of clients is to train local models with their own private data and upload the training results to the server. Meanwhile, the task of server is to train a global model based on the received updates. Through this design, the private data of clients is kept stored locally and does not need

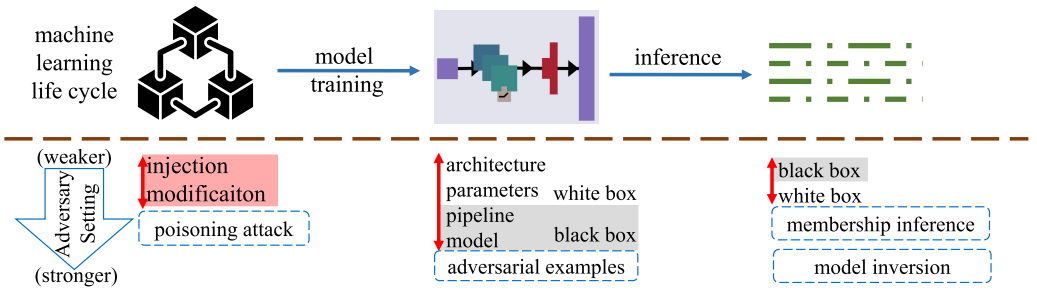


Fig. 3. Capabilities of adversaries required in different attacks.

to be shared with others. This architecture improves the protection of clients' privacy. However, it reduces the transparency of training data and exacerbates the threat of poisoning attacks, leading to poisoning attacks becoming a major security threat in federated learning [6, 8, 32, 123].

As a data-driven threat, poisoning attacks are insensitive to target models. They can be applied to both supervised learning and unsupervised learning [5, 9, 12, 106]. Moreover, carefully constructed poisoning samples show transferability against various learning algorithms [143, 160]. Therefore, in this survey, we also highlight the connection between different poisoning attacks.

### 2.3 Basics of Poisoning Attack

The basic goal of a poisoning attack is to degrade the overall performance of a target model [11, 51]. On this basis, there are advanced goals, e.g., hiding the tracks of attacking or specifying the range of abnormality [42, 108]. Specifically, the following metrics can be used to measure the attacking performance. The classification results can be divided into four types: **True Positive (TP)**, **True Negative (TN)**, **False Positive (FP)**, and **False Negative (FN)**. TP and TN are the correct predictions, and FP and FN are the wrong predictions. More advanced metrics can be calculated based on these predictions, e.g., Precision, Recall, Accuracy, and F-measure. The goal of an attack is trying to decrease or increase some of these metrics while keeping others unaffected.

The prerequisite for the implementation of a poisoning attack is that the adversary has the capabilities to influence the training dataset. As shown in Figure 3, in a poisoning attack, the necessary capabilities of adversaries are to inject and modify a certain amount of training data [5, 108]. In contrast to other attacks, poisoning attacks require capabilities only related to the dataset. It is easy to obtain these capabilities, since training datasets are usually collected from public resources. The situation is slightly different in federated learning, where the poisoning attacks are launched through poisoning updates rather than poisoning data. Furthermore, the adversaries' knowledge of the target model also affects poisoning attacks. In general, the more the adversary knows, the more successful the attack will be. They could be classified as perfect-knowledge adversaries, limited-knowledge adversaries, and no-knowledge adversaries [4, 5].

## 3 TAXONOMY OF POISONING ATTACKS

**Poisoning attack** is firstly proposed by Barreno et al. [5]. Since then, poisoning attacks have received widespread attention and became the primary security threat in the training stage of machine learning. From the perspective of adversarial goals, poisoning attacks can be classified into two classes: **untargeted poisoning attack** and **targeted poisoning attack**. The goal of untargeted poisoning attacks is to hinder the convergence of the target model and eventually lead to denial-of-service. For example, adversaries can flip labels of some samples in the training dataset

Table 1. Comparison of Untargeted Poisoning Attacks

| Literature | Adversarial Setting | | Target Model | Poisoning Technique | Application | Evaluation | Year |
|---|---|---|---|---|---|---|---|
| | Knowledge | Capability | | | | | |
| [106] | N-K to P-K | IN | PCA | data inserting | intrusion detection | ROC | 2009 |
| [10] | N-K to L-K | MO | SVM | label manipulation | binary classification | accuracy | 2011 |
| [11] | P-K | IN | SVM | data manipulation | image classification | classification error | 2012 |
| [90] | L-K | IN | Naives Bayes, SVM | label, data manipulation | sentiment analysis | accuracy | 2014 |
| [133] | P-K | MO | SVM | label manipulation | DNA,acoustic,seismic | classification error | 2015 |
| [143] | L-K to P-K | IN | [26, 69] | data inserting | recommender systems | attack success rate | 2017 |
| [156] | N-K | MO | linear regression,SVM | label manipulation | binary classification | accuracy | 2017 |
| [84] | L-K | MO | Deep Learning | data manipulation | malware, image classification | classification error | 2017 |
| [51] | N-K to L-K | IN | linear regression | data manipulation | regression | MSE | 2018 |
| [32] | L-K | MO | Federated Learning | updates manipulation | multi-classification | classification error | 2020 |

N-K: no-knowledge, P-K: perfect-knowledge, L-K:limited-knowledge.
IN: injection, MO:modification.

to perturb the knowledge contained in the benign sample-label pairs [11, 17, 32, 84]. In targeted poisoning attacks, the adversarial goal is to force the target model to produce abnormal predictions on particular inputs [74, 111, 119]. For example, an adversary may force the target model to predict all digit "3" as "5".

Moreover, there is an advanced kind of targeted poisoning attack, **backdoor attack**. Backdoor attacks take one more step based on targeted poisoning attacks. The goal of backdoor attacks is to make the targeted attack unnoticed. A backdoor can be implanted into the target model through some poisoning samples with a fixed pattern. The backdoored target model will only perform abnormal on inputs contained the same pattern [3, 22, 42]. For example, as shown in Figure 5, an adversary can patch poisoning samples with same pixel pattern on the same position. Then, model trained by these poisoning samples will perform abnormally on similar backdoor samples. The attacking trace is well hidden, since there is no performance anomaly of the poisoned model on non-backdoor samples.

Targeted poisoning attacks are more sophisticated than untargeted poisoning attacks. An adversary only needs to perturb the knowledge contained in the benign sample-label pairs to make the model denial-of-service. However, to launch a targeted poisoning attack, an adversary needs to implant malicious knowledge into the training dataset while keeping other knowledge unaffected. The technical details are discussed in later sections.

### 3.1 Untargeted Poisoning Attack

Untargeted poisoning attack is the most intuitive kind of attack. As the name suggests, untargeted poisoning attacks do not have a specific target class. The goal of the adversary is declining the overall performance of the target model, such as classification accuracy and **receiver operating characteristic curve (ROC)**.

We list some representative untargeted poisoning attack methods in Table 1. In the initial stage of the research, the main targets are the conventional machine learning models represented by SVM [11, 106]. Biggio et al. [10, 11] have discussed untargeted poisoning in SVM in detail. They carried out poisoning attacks from the perspective of labels and samples. The challenge is to use the smallest possible poisoning data to affect the legitimate function of the target model to the greatest extent. For example, Mei et al. [80] proposed a general algorithmic framework for poisoning attack instance optimization on conventional machine learning. In their research, an optimal change to the training dataset is optimized by a convex optimization loss. In this way, they could poison a target model with small data perturbation.

Conventional machine learning methods need to carry out pre-processing operations (e.g., feature extraction) on the raw data before they are input into the model, and the data involved in the model training is usually low dimension feature vectors [5, 90, 132]. In this scenario,

adversaries need to know the feature set of the target model in advance to launch the attack efficiently. Many works against conventional machine learning have assumed multiple adversaries with different knowledge levels, from perfect-knowledge to no-knowledge. The general trend is that the more knowledge an adversary has, the more effective the attack will be [51, 90, 106, 143]. Feature engineering could also be compromised by poisoning attacks. For example, Xiao et al. [132] investigated the impact of feature selection on poisoning attacks. They managed to reduce several popular feature selection methods to almost random choices, which highlighted the need for countermeasures on feature engineering.

Subsequently, with the development of deep learning and federated learning, more and more studies focus on exploring vulnerabilities of these two types of learning methods [32, 84, 137]. The main difference between deep learning and conventional machine learning on inputs is that there is no need for feature engineering. The collected samples can be directly input into a deep learning model for training, especially in the field of computer vision. Therefore, poisoning attacks in deep learning require relatively less knowledge about the target model. In the literature [151], Zhang et al. conducted a detailed experimental analysis on the generalization ability of deep learning models. They trained several standard architectures on a copy of the data where the true labels were replaced by random labels. They found that deep learning models easily fit random labels. Moreover, they also evaluated the model's training results on samples that had random permutations of the pixels. Their experimental results demonstrated that the model can still converge, although the performance of the trained model is similar to random guessing on normal sample. Their research reveals the vulnerability of deep learning models in untargeted poisoning attacks, laying the foundation for subsequent research.

Untargeted poisoning attacks exhibit good transferability. As illustrated in Table 1, many poisoning attack methods can attack more than one target model. This is mainly because the untargeted poisoning attack is a relatively simple task, which is considered successful as long as the performance of the target model is reduced. Furthermore, not only for classification tasks, poisoning attacks are also a major threat in other application scenarios, such as co-visitation recommender systems [143] and regression tasks [51]. In [143], the target model recommends items to the users from the collected co-visit logs. To attack this recommendation system, Yang et al. used scripts to automatically co-visit a target item to insert poisoning logs into the recommender systems.

The characteristics of untargeted poisoning attacks determine that this type of attack method is noticeable. The victim can easily notice the abnormality after the model's performance drops. To better capture the benefits from the target model, but also to better hide their traces, adversaries developed another type of attack, which is targeted poisoning attacks.

### 3.2 Targeted Poisoning Attack

The tasks of machine learning have become increasingly complex. The most common scenario considered in previous literature [11, 80, 91] is to poison binary classification models to cause a denial-of-service. However, adversaries are no longer satisfied with making the victim model unable to provide services. They try to manipulate the target model with more sophisticated poisoning attack techniques. Different from untargeted poisoning attacks, targeted poisoning attacks focus more on the class of wrong predictions or the class of input causing the wrong predictions. The targeted poisoning attack could be formulated as a multi-task problem. Adversaries force the target model to perform abnormally on specified samples while ensuring its legitimate function on other benign samples. For example, in a digit classification task, the adversary tries to force the model to mis-classify digit "7" while performing normally on other digits.

Koh et al. [57] proposed a targeted poisoning attack that modifies training samples which have a strong influence on the target loss. They optimize an objective function to maximize the influence

Table 2. Comparison of Targeted Poisoning Attacks

| Literature | Adversarial Setting | | Target model | Poisoning Technique | Application | Backdoor | Year |
|---|---|---|---|---|---|---|---|
| | Knowledge | Capability | | | | | |
| [138] | L-K | IN | [159], contextual, personalization | DI | personalized web services | – | 2013 |
| [57] | P-K | MO | Deep Learning | DM | image classification | – | 2017 |
| [119] | L-K | MO | Deep Learning, SVM, random forest | DM | image, malware classification | – | 2018 |
| [111] | N-K | IN | Deep Learning | DM | image classification | – | 2018 |
| [8] | P-K | MO | Federated Learning | UM | image classification | – | 2019 |
| [160] | L-K | IN | Deep Learning | DM | image classification | – | 2019 |
| [152] | P-K | MO | Federated Learning | UM | image classification | – | 2021 |
| [62] | N-K | MO | SVM, random forest | DM, LM | Android malware detection | – | 2021 |
| [22] | N-K | IN | Deep Learning | DM | image classification | ✓ | 2017 |
| [42] | P-K | MO | Deep Learning | DM | image classification | ✓ | 2019 |
| [108] | N-K | MO | Deep Learning | DM | image classification | ✓ | 2020 |
| [65] | L-K | MO | Deep Learning | DM | natural language processing | ✓ | 2021 |

N-K: no-knowledge, P-K: perfect-knowledge, L-K:limited-knowledge.
IN: injection, MO:modification.
LM: label manipulation, DM: data manipulation, UM: update manipulation, DI: data inserting.

of poison added to the training samples on the classification results of the test data. Similarly, Jagielski et al. [52] proposed a subpopulation attack, which is to compromise the performance of a target model on a designated subpopulation, while maintaining its performance for the rest of the dataset.

Targeted poisoning attacks are more difficult to achieve. In targeted poisoning attacks, an attack will be considered as a success only when the error occurs on samples desired by the adversaries, while performing normally over other samples. From a technical perspective, untargeted poisoning attacks can be achieved by introducing random noise into a training dataset, while targeted poisoning attacks require the careful construction of poisoning samples that can cause the model to exhibit specific misbehaviors. This puts forward a high requirement for the construction of poisoning samples.

**Backdoor poisoning attacks.** With the industrialization of cyber attacks, increasing adversaries are looking for long-term benefits from attacks. How to hide the attack traces become another goal pursued by adversaries. An intuitive way to hide the attack trace is to make the abnormal performance of the target model less noticeable by narrowing down the range of abnormal performance. Attacks that achieve this adversarial goal are called backdoor poisoning attacks.

In this survey, we classify backdoor poisoning attacks as a subclass of targeted poisoning attacks. As described above, targeted poisoning attacks need to maintain the overall performance of the target model and cause misbehaviors on targeted inputs/outputs. Backdoor poisoning attacks meet this taxonomy, while it has higher requirements for samples that activate the hidden backdoor. Adversaries introduce some patterns (known as **trigger**) into the poisoning samples for implanting a backdoor during model training. Furthermore, only samples containing the same trigger can activate the hidden backdoor. For example, Chen et al. [22] aimed at embedding a backdoor into the target model, so that the target model will be misled to classify the trigger samples as a target label specified by the adversary. They managed to implant a specific type of glasses into the target model as the trigger of the backdoor, so that anyone wearing these glasses could be misidentified as a specified person. Meanwhile, the target model performs normally when inputs contain no trigger.

We list several representative literatures of targeted poisoning attacks in Table 2. It is worth noting that the main target of targeted poisoning attacks is the deep learning model. We argue that this is due to the great progress made in deep learning methods in recent years. Therefore, it attracts more attention from adversaries. The mainstream poisoning technique is data manipulation. This is because targeted poisoning attacks require more delicate misleading knowledge contained in poisoning samples. In the following sections, we will present detailed introductions on these poisoning techniques.

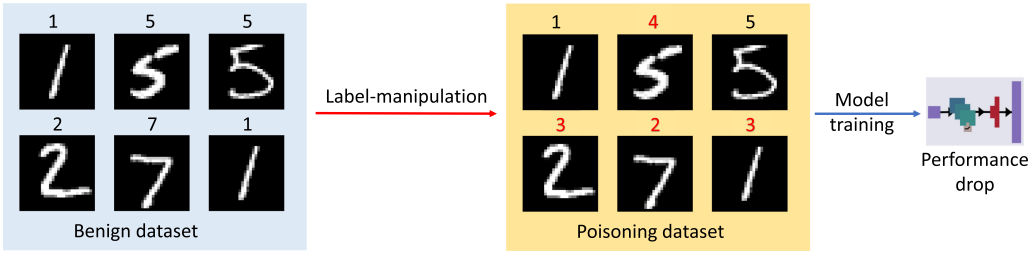


Fig. 4. Instance of label manipulation.

## 4 TECHNIQUES OF POISONING ATTACKS

The growing volume of training datasets leaves the door open for poisoning attacks. It is hard for the researchers to analyze the training samples one by one. For example, ImageNet [28] is a large database for visual object recognition research. It contains more than 14 million images from more than 20,000 categories and a label for each image. The performance of models can be affected by poisoning samples maliciously introduced into their training dataset. In poisoning attacks, how to construct poisoning samples is a fundamental research problem. In this section, we leverage a manipulated-target-based taxonomy to classify poisoning techniques into two categories: label manipulation and data manipulation.

### 4.1 Label Manipulation

A commonly used poisoning technique is label manipulation. The knowledge learned by machine learning is mainly from sample-label pairs. Therefore, the performance of the machine learning model deteriorates as long as the true patterns in the sample-label pairs are disrupted. An intuitive strategy is to flip labels of training samples [10]. As shown in Figure 4, an adversary can flip labels of random samples to launch a poisoning attack. Barreno et al. [5] analyzed the situation of using poisoning attack techniques to mislead the classification of machine learning-based IDS. In their research, the authors discussed the possibility of manipulating a learning system and the possible strategies for how adversaries place poisoning samples into the training process, for the first time. Subsequently, Biggio et al. [10] analyzed the effect of noise introduced through label-flipping and defenses in SVM.

Deep learning algorithms perform similarly when training datasets are poisoned by label-flipping samples. Zhang et al. [150] designed extensive experiments to evaluate the performance of deep learning models under label-flipping and revealed the reasons for the success of label-flipping in deep learning. They trained several models in a similar structure on a copy of the data where the true labels were replaced by random labels. In their study, neural networks achieve 0 training error trained on a completely random labeling of the true data. And the test error is no better than random chance. Their experiments show that the deep learning algorithm can be forced to learn the misleading knowledge in the random label-sample pairs. This is a typical over-fitting phenomenon. The victim can perceive this denial-of-service attack in the training process through the abnormal behavior of the model on the test dataset. However, at that time, a lot of computing resources and time have been consumed. The adversarial goal has been achieved already.

The attacking potential is limited, if adversaries are only able to modify the label information of the training dataset. Therefore, they need to find the most harmful labels or data points to amplify the poisoning impact. In model training, not every sample has the same influence [89]. Selecting and modifying labels of appropriate samples can effectively improve the efficiency of attacks. Specifically, it includes which samples' label can be flipped and flip to which label. For example,

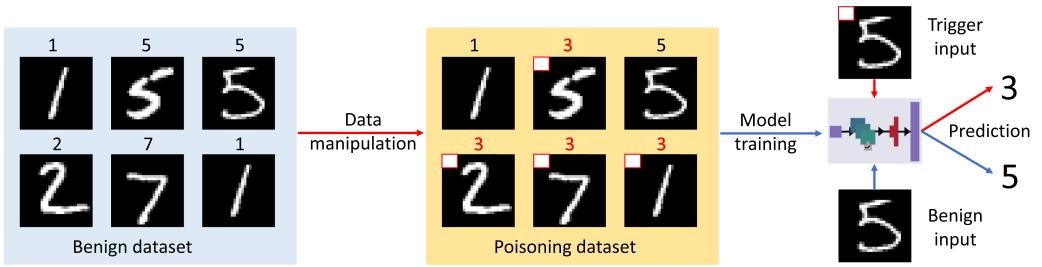


Fig. 5. Instance of data manipulation.

Biggio et al. presented a method to find the most optimal attack sample-label pairs for the attack [10]. First, they flip labels of samples with non-uniform probabilities to evaluate how well these samples are classified by the SVM learned on the untainted training dataset. Then, they repeat this process several times and retain the label flips that maximally decrease performance. Similarly, the adversary in [134] aims to find a combination of label flips under a given budget so that the target model trained on such dataset will have maximal classification error. The most effective dataset of label flips is found via an optimization method. Zhao et al. [156] investigated efficient label manipulation method and its transferability against black-box models. They formulated the attack problem as a bilevel optimization problem by assuming that the adversary's goal is to maximize the cosine of the angle between learner's learned weight vector and the objective model's weight vector. Then they develop a **Projected Gradient Ascent (PGA)** algorithm to solve this problem.

The advantage of label manipulation lies in its straightforward operation, which only needs to modify the labels of some data points. However, there are also disadvantages. One of the most obvious disadvantages is the limitation of achieving sophisticated adversarial goals. The poison introduced through the label manipulation is naive. It usually reduces the performance of the model significantly, which makes it easy to be noticed by the victim. However, sophisticated adversarial goals often require stealth during the attack and complex poisoning knowledge, which is difficult to achieve only by modifying the labels. Therefore, sole label manipulation based attacks are gradually declining in the research.

### 4.2 Data Manipulation

Intuitively, sample space has more potential to achieve sophisticated attacks than labels. As shown in Figure 5, an adversary can poison the target model by inserting a certain pattern (white box in the upper left of the sample) into training samples. For example, in literature [89], the adversary tries to perturb the SpamBayes learning method [81] via embedding more words likely to occur in a legitimate email into attack emails. In literature [42], the adversary embeds a fixed pixel pattern into the poisoning samples to create a backdoor in the target deep learning model. However, with the awakening of safety awareness, the application scope of this manual data manipulation poisoning method is shrinking. Gradually, adversaries develop two types of data manipulation poisoning methods, optimization-based data manipulation and training-based data manipulation.

*4.2.1 Optimization-based Data Manipulation.* Optimization-based data manipulation, as the name implies, is a kind of method to generate poisoning samples by optimizing an objective function. The goal is to generate samples that have negative influence on the training of the target model. First, an adversary designs an objective function. Then, some optimization methods could be used to optimize this objective function. The two key points are the design of the objective function and the optimization method.

In literature [11], the objective function is designed as maximizing the hinge loss incurred on the validation dataset by the SVM trained on the poisoned dataset. And this objective function is optimized via the gradient ascent technique. In this way, the adversary can construct a poisoning sample that significantly decreases the SVM's classification accuracy. Similarly, bilevel optimization-based poisoning attacks are also conducted on linear regression [51] and logistic regression [27].

The aforementioned poisoning attacks against conventional machine learning algorithms are convex objectives [11, 57, 80], which require solving a bilevel optimization problem. The complexity on computing limits their applications on deep learning architectures. To tackle this challenge, Muñoz-González et al. [84] extended poisoning attacks to deep learning models based on back-gradient optimization. Modifying a sample using gradients allows any sample to attack another designated class that eliminates the need for careful selection of original samples. This method can carry out targeted and untargeted attacks under the same circumstance through different objective functions. Furthermore, Huang et al. [50] proposed the MetaPoison, which is a first-order optimization method that approximates the bilevel problem via meta-learning and constructs poisoning samples against deep learning models.

In data manipulation, another hot research area is how to hide the attack traces. An intuitive idea of detecting poisoning samples is visual inspection. The defenses may develop a classifier to examine training samples (details could be found in Section 6). To counter this kind of detection, adversaries need to upgrade the design of objective functions. They can add a bound in the optimization process (or the objective function) to constrain the degree of similarity between the generated poisoning sample and the original sample. For example, Shafahi et al. [111] proposed a clean-labels poisoning attack against deep learning models. Clean-labels means that the adversary does not manipulate the labels of training data. They constructed a poisoning sample through an objective function, in which the poisoning sample was constrained to be visually similar with a benign sample but contained poisoning information. And they optimized this objective function via a forward-backward-splitting iterative procedure [39].

In the computer vision domain, only part of the information in the image is considered as features, which makes a large amount of redundant space in the image. These spaces provide a great convenience for hiding traces of poisoning attacks. Saha et al. [108] proposed a method to embed triggers into the images. They managed to force poisoning images close to target images in the pixel space and also close to source images patched by the trigger in the feature space. In other words, the optimized poisoning sample looks and labels the same as the benign sample. However, models trained with these poisoning samples present abnormalities when they predicted specific patched samples. This type of attack makes full use of the pixel space of the image sample, hiding the poisoning information from visual based detection.

Similar technique can also be used in poisoning attacks against other application domains. Li et al. [62] proposed a backdoor attack on machine learning based Android malware detectors. They argued that the challenge of launching poisoning attacks in Android malware detectors is the feature engineering. The features in Android malware detection are usually extracted from application behaviors. These features are harder to modify than image pixels due to their semantics. Therefore, they need to design triggers that meet semantic requirements, which includes trigger position, trigger size, and so on.

The advantage of an optimization-based poisoning attack is its control of poisoning samples. The samples generated by such methods can accomplish almost any adversarial goal, with reasonable design. However, there are also several disadvantages. First, most optimization methods can only generate one poisoning sample once, which is obviously not efficient; second, gradient-based local optimization often gets stuck into bad local optima and fails to find effective sets of poisoning points [58, 117].

*4.2.2 Training-based Data Manipulation.* In training-based data manipulation, an adversary leverages an auxiliary model to help generate poisoning samples. There are two kinds of auxiliary models: the isomorphic auxiliary model and the generative model. It is hard for adversaries to access the target model without restriction during the attack in a real-world scenario. Therefore, to generate a better poisoning sample, an adversary can train an isomorphic auxiliary model to simulate potential target models. Then, construct poisoning samples based on its performance. For example, Zhu et al. [160] proposed a transferable clean-label poisoning attack. The adversary trains substitute models on an auxiliary dataset, and optimizes an objective function that forces the poisons to form a polytope in feature space that entraps the target inside its convex hull. A model that trained over this poisoning data will classify the target into the same class as that of the poisons.

Suya et al. [120] proposed a model-targeted poisoning attack. They first generated a model via the heuristic approach proposed by [58]. Then, they sequentially added a poisoning sample into the training dataset of the intermediate model and train it. Afterwards, the adversary searches for the poisoning sample that maximizes the loss difference between the intermediate model obtained so far and the target model. Finally, the poisoning dataset was generated through this optimization process. Because the adversary can evaluate the poisoning dataset via the training intermediate model, the poisoned sample dataset is constructed by maximizing the loss between models, which improves the effectiveness of the poisoning attack.

In the NLP domain, Li et al. [65] proposed the homograph attack for Unicode homographs and the dynamic sentence attack for more general NLP scenarios. Homograph backdoor attack generates the poisoned sentences by inserting the homograph replacement trigger. However, this attack can be easily identified by word error checkers. Therefore, they proposed a more delicate poisoning attack based on the intuition that the perplexities vary among texts generated by different language models [116]. Specifically, they input a small set of samples into the trained language models to generate a context-aware suffix sentence as the trigger. The auxiliary language model is trained on a corpus that has similar topics to the target tasks to hide the trigger.

In recent years, generative models, such as auto-encoder [56] and **Generative Adversarial Network (GAN)** [40] gained a huge success. Generative models can use knowledge learned from the training dataset to generate new samples. The ability of generating new samples makes it widely used in various fields, such as image generation [75], style transfer [53], super-resolution [149], and **Natural Language Processing (NLP)** [25]. Many poisoning works also utilize generative models to construct poisoning samples. An adversary can significantly improve the efficiency of constructing poisoning samples with the help of generative models.

The challenge of leveraging generative models is how to generate poisoning samples that fit the human experiences on application domains. Yang et al. [142] proposed a GAN-structure generative method to speed up the generation rate of the poisoning samples. They used an autoencoder as the generator and consider the target model as the discriminator. Then, they trained the generator to generate the poisoning samples by a reward function of the loss. In the work pGAN [85], the architecture contains three components: generator, discriminator, and the target model. The generator is used to generate poisoning samples that maximize the error of the target model. The discriminator is used to distinguish the real and fake samples. The target model is used to minimize some loss function evaluated on a training dataset that contains a fraction of poisoning points.

Compared with optimization-based data manipulation [11, 108], training-based data manipulation improves the attacking efficiency and solves a the challenge of lacking information. However, the stealth of the samples generated by generative models is not as good as that based on optimization methods.

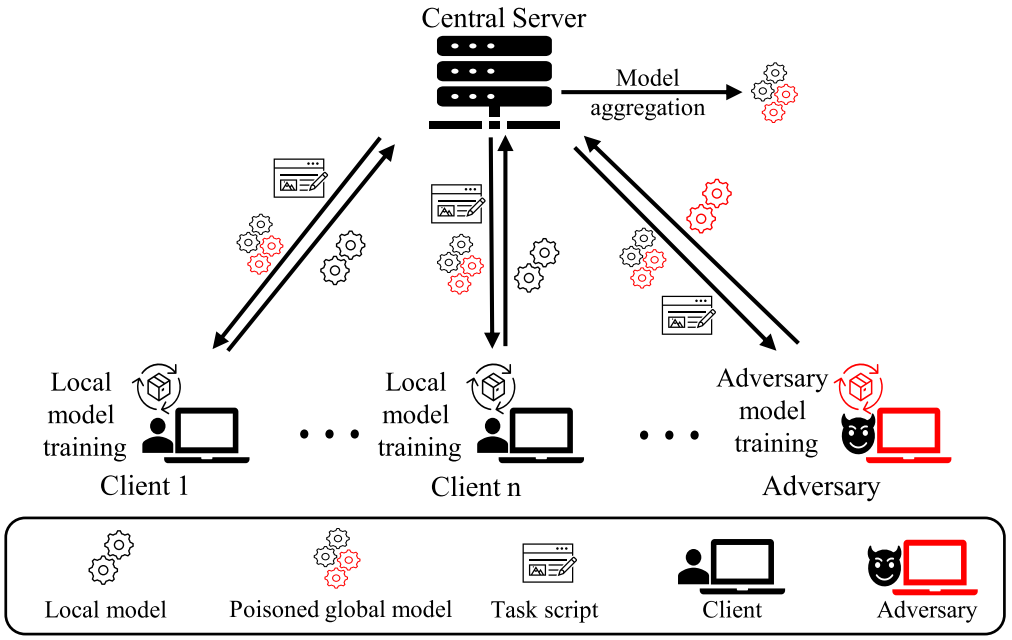


Fig. 6. Architecture of poisoning attack in federated learning.

## 5 POISONING ATTACKS IN FEDERATED LEARNING

### 5.1 Overview of Poisoning Attacks in Federated Learning

Federated learning is a new practice of machine learning in the training stage. As training data contains private information, users are increasingly reluctant to share their data. To solve this problem, researchers designed the federated learning framework. In this survey, we focus on the horizontal federated learning in client-server architecture [144]. As shown in Figure 6, in this architecture, clients do not share their private data. Instead, they use their private data to train local models and upload them to the server. Then, the server aggregates the local models to form a global model and distributes this global model to all clients. The motivation for focusing on client-server federated learning is that it is the major architecture applied in current scenarios. For the security community, only after a technology has gained widespread use will its security issues be discussed in depth. Therefore, we find the majority of literature focuses on federated learning. On the other hand, the security problems of other distributed learning architectures are neglected. We present a discussion about them in Section 7.2.

Federated learning is designed to protect users' privacy, however, the distributed architecture makes it vulnerable to poisoning attacks. This architecture prevents the server from accessing the training data of clients, and datasets between clients are usually **not independent identically distributed (non-iid)**. This makes it more difficult for the server to detect poisoning attacks, because the training results updated by different clients are inherently different. The target in federated learning is usually the global model aggregated by a trusted server. The adversary's goals are the same as poisoning attacks in centralized learning, which is to force the global model unable to converge or perform abnormally on designated inputs.

In federated learning, the threat model is often set such that the malicious clients can send arbitrary gradient updates to the server [60, 82]. Different from centralized machine learning, the

server distributes the current global model periodically during the training process to clients for follow-up training. It provides adversaries with great convenience, so that the attack process can be adjusted in real-time to maximize the attack effect. At the same time, since the server does not have access to clients' private data, adversaries can use arbitrary malicious data to train their local models.

In federated learning, the adversarial goals are the same as in centralized machine learning: untargeted [32, 123], targeted [8, 123], and backdoor poisoning attacks [3, 136]. Among them, backdoor poisoning attack is the most harmful threat. For example, the adversary in [3] tried to produce a global model that achieves high accuracy on both its main task and an adversary-chosen backdoor subtask, and retains high accuracy on the backdoor subtask for multiple rounds after the attack. If a backdoor is embedded in the model deployed to the field, it will cause security risks. In contrast, if the model cannot converge due to a poisoning attack, it will not be deployed in practice. The damage caused by such attacks is wasting the victim's computing resources. Therefore, backdoor attack is one of the most concerning threats in federated learning.

Adversaries in federated learning also face unique challenges. The challenges come from two aspects: influence limitations and countermeasure settings. The first challenge is the influence limitation of compromised clients. To poison a federated learning system, the adversary should first control some clients, and manipulate their updates. In most scenarios, the adversary can only control a small number of the clients to carry out poisoning attacks [8, 123, 136]. The influence of a single client on the global model is limited. Moreover, updates from benign clients will also weaken the influence of poisoning updates. Therefore, how to contaminate the global model with limited influence is the first challenge.

The second challenge is the existence of countermeasures. Compared with centralized machine learning, federated learning has a certain backwardness advantage. In federated learning, aggregation strategy is the core of the design of federated learning for training an optimal global model. For example, FedAvg algorithm randomly selects a number of clients for each round of global model training [79]. The architecture of distributing the training process into different entities is easier to connect. Countermeasures can be easily added to the federated learning framework. Therefore, in the process of carrying out a poisoning attack on the federated learning, the countermeasure is another challenge that adversaries must face.

In the following subsection, we focus first on the challenges of implementing poisoning attacks in the federated learning framework from the adversary's perspective. The countermeasures in federated learning will be discussed in Section 6.2.

### 5.2 Poisoning Techniques in Federated Learning

In the federated learning scenario, the techniques used for constructing poisoning samples are similar to that in the centralized machine learning scenario. For example, Tolpegin et al. [123] and Cao et al. [17] used label flipping to implement poisoning attack against federated learning. Zhang et al. [152] proposed PoisonGAN, a GAN-based poisoning samples construction method to attack both centralized learning and federated learning. A similar technique was also used in poisoning attacks against centralized machine learning [142].

The framework of federated learning is a double-edged sword for the adversary. On the one hand, it provides the adversary with the convenience of attack; on the other hand, attack efficiency is affected by other benign clients. Specifically, the range of poisoned data is limited in federated learning, an adversary could only inject or modify the malicious data into the compromised clients' datasets. Therefore, even if the local model is trained on only poisoning data, it may not be able to achieve an effective poisoning attack. Meanwhile, clients are black-box for the server, an adversary can arbitrarily manipulate the training process of the local model to implement a poisoning

Table 3. Comparison of Poisoning Attacks in Federated Learning

| Literature | Adversarial Setting | | Countermeasures | Year | Strategy |
|---|---|---|---|---|---|
| | Goal | Malicious Clients | | | |
| [8] | T | 1 | Krum\|median | 2019 | Alternating minimization strategy. |
| [6] | UnT\|BD | 24% | TM\|Krum\|Bulyan | 2019 | Poisoning updates are limited in a perturbation range. |
| [123] | T | 2%∼20% | – | 2020 | Using label flipping to implement poisoning. |
| [32] | UnT | ≤50% | TM\|Krum\|Bulyan\|median | 2020 | Optimization-based poisoning updates construction. |
| [136] | BD | 1 | Multi-Krum\|Bulyan | 2020 | Decompose and distribute the backdoor trigger. |
| [3] | BD | 1∼5% | Discussed | 2020 | Model replacement. |
| [152] | T | 1%∼5% | Discussed | 2021 | Using GAN to generate poisoning samples. |

UnT: Untargeted poisoning attack, T: Targeted poisoning attack, BD: Backdoor poisoning attack.
TM: Trimmed Mean [147], Krum [13], Bulyan [82], median [147].

attack. For example, Fang et al. [32] assumed an adversary has control of some client devices and manipulates their updates during the training process. They consider the adversary's goal is to deviate a global model parameter the most towards the inverse of the direction along which the global model's parameter would change without attacks. They achieve this goal via solving an optimization problem in each iteration. Furthermore, in federated learning, the server will distribute the global model to clients as a basis during training. Adversaries can dynamically modify the poisoning contents according to the changes of the global model during the attack. However, their method only increases the toxicity of a single poisoning update while the poisoning effect is decided by the the poison amount added into the whole learning system.

The number of clients compromised by adversaries affects the attack efficiency. When the number of clients participating in federated learning is large, the impact of a single client on the global model is limited. Xue et al. [141] even proposed a notion to quantify the client's influence. It is common to assume that the adversaries compromised at least one client, as shown in Table 3. When malicious clients are more than one, the poisoning attacks can be formalized as sybil attacks [115]. In a sybil attack, the adversary controls a part of clients and uses them to gain a disproportionately large influence by performing the same behavior. The more clients an adversary compromises, the higher the chance of a successful attack [8]. In general, adversaries will follow the training protocol to keep the attack stealthy. The datasets held by adversaries are not restricted in this case, as no one could have access to these data except the adversaries themselves. Most poisoning attacks against federated learning scale up the updates to increase the success rate of attacks [3, 8].

However, the trade-off between efficiency and stealth limits the updates amplification. In most federated learning scenarios, most clients are benign. If the behavior of malicious clients is significantly different from that of benign clients it will result in the failure of the attack. Therefore, it is necessary to consider whether the attack can be detected while improving the attack efficiency. For example, Bhagoji et al. [8] modified the malicious objective to account for some metrics to carry out stealthy model poisoning which allowed the malicious weight update to avoid detection for a majority of the rounds. They propose an alternating minimization formulation that accounts for both model poisoning and stealth, and enables the malicious weight update to avoid detection in almost all rounds. Baruch et al. [6] provided a perturbation range in which the adversary can change the parameters without being detected even in an **independent identically distributed (iid)** setting.

An adversary can also achieve more complex attacks by controlling behaviors of multiple malicious clients. For example, Xie et al. [136] proposed **distributed backdoor attacks (DBA)**, which decomposed a global trigger pattern into separate local patterns and embed them into the training dataset of different adversarial parties respectively. The decomposing trigger makes malicious clients behave more like benign clients, making such backdoor attacks harder to detect.

Through the above analysis, it is not difficult to find that federated learning faces more severe security challenges than centralized machine learning. The vulnerability of federated learning is

due to its architecture. The opacity of the training data makes poisoning harder to detect. Multiple clients result in non-iid training data, which also gives adversaries more room to conduct poisoning attacks. We will examine in detail on how to design countermeasures against these challenges in Section 6.2.

## 6 COUNTERMEASURES AGAINST POISONING ATTACKS

The confrontation between adversaries and defenders is an endless war. Many researchers have made efforts on countering poisoning attacks [13, 20, 82]. The core intuition of countermeasures on poisoning attacks is that poisoning data contains malicious information. The malicious information leads to abnormality on the poisoning samples, no matter how cleverly they are hidden [21]. Therefore, in the scenario where training data can be directly accessed, such as centralized machine learning, many techniques can contribute to countering poisoning attacks. In federated learning, on the other hand, it is much more difficult to counter poisoning attacks due to the lack of direct access to training data, as shown in Figure 6. Therefore, countermeasures face tough challenges in the federated learning scenario.

In this section, we elaborate on countermeasures against poisoning attacks in centralized machine learning and federated learning.

### 6.1 Countermeasures in Centralized Machine Learning

Centralized machine learning can be divided into two categories: conventional machine learning and deep learning. The main difference between these two categories is that conventional machine learning requires expert experiences to carry out feature engineering, while deep learning directly integrates feature extraction into the model training. Countermeasures can be integrated easily into conventional machine learning, due to the relative independence of feature engineering and model training. Deep learning models directly process raw data, such as image samples [119, 160]. The integrated architecture of deep learning models brings particular difficulty to the design of countermeasures. The raw data contains a lot of redundant space that is not used for the main task, which offers adversaries huge opportunities to launch poisoning attacks. Furthermore, in deep learning, features are learned by algorithms and it is challenging to detect abnormal samples from the raw training dataset. At the same time, the influence of poisoning attacks on trained deep learning models is difficult to perceive, as deep learning tasks become more complex. There are even backdoor attacks that precisely manipulate the behavior of the trained model [22, 42, 108].

According to different application scenarios, we divide the countermeasures into two categories: countermeasures in conventional machine learning and countermeasures in deep learning.

*6.1.1 Countermeasures in Conventional Machine Learning.* Countermeasures in conventional machine learning could be carried out before and during the model training. In this survey, we define them as data-driven countermeasure and model-driven countermeasure. In Table 4, we list some of them in the conventional machine learning scenario.

**Data-driven countermeasure.** Data-driven countermeasures are mostly performed on the preprocessing phase before model training. The core intuition is that poisoning samples are different from benign samples, and this difference can be detected. The robustness of the model can be improved by removing outliers from the training dataset. This kind of countermeasure is usually independent of the learning algorithm.

Cretu et al. [23] proposed a data sanitization countermeasure for anomaly sensors, which was an early research on preventing training data poisoning. The full dataset is examined to remove the poisoned samples. Subsequently, Barreno et al. [4] proposed the RONI defense technique in machine learning domain. RONI measures the effect of training samples and eliminates those

Table 4. Comparison of Countermeasures Against Poisoning Attacks in Conventional Machine Learning.

| Literature | Model-protected | Strategy | Adversarial Setting | | Year |
|---|---|---|---|---|---|
| | | | Techniques | Goals | |
| [4] | cross-models | data-driven | cross-attacks | UnT | 2010 |
| [33] | logistic regression | data-driven | DM | UnT | 2014 |
| [72] | cross-models | data-driven | LM | UnT | 2015 |
| [99] | cross-models | data-driven | LM | UnT | 2018 |
| [98] | cross-models | data-driven | cross-attacks | UnT | 2018 |
| [29] | cross-models | data-driven | DM | UnT | 2019 |
| [118] | SVM | model-driven | LM | UnT | 2009 |
| [10] | SVM | model-driven | LM | UnT | 2011 |
| [145] | Multiple Kernel Learning | model-driven | LM | UnT | 2012 |
| [88] | cross-models | model-driven | LM | UnT | 2013 |
| [104] | cross-models | model-driven | cross-attacks | UnT | 2020 |
| [70] | linear regression | both classes | DM | UnT | 2017 |

UnT: Untargeted poisoning attack, LM: label manipulation, DM: data manipulation, cross-models: protect multiple models, cross-attacks: counter multiple poisoning techniques.

samples that have negative impacts on classification accuracy. To determine the malicious level of training samples, they trained an auxiliary classifier on an auxiliary dataset. Then gradually add samples that can improve the accuracy of the auxiliary model to the training dataset. Similarly, Liu et al. [72] proposed an importance reweighting method, which could enhance the robustness of any traditional classification surrogate loss function against label manipulation.

Some data-driven countermeasures utilize learning algorithms' performances as features to eliminate poisoning samples. Feng et al. [33] proposed a **robust logistic regression algorithm**, called **RoLR**, against arbitrary training samples. RoLR eliminates the adversarial training samples by solving an optimization problem, which optimizes a robustified linear correlation between response and linear measure through an efficient linear programming-based procedure. Specifically, RoLR first removes overly large magnitude samples and then maximizes a trimmed correlation of the rest of the training samples with the trained logistic regression model. The performances of this method is affected greatly by a preset hyper-parameter, percentage of malicious training samples, which means that this method relies heavily on expert experiences. Paudice et al. [98] proposed a similar countermeasure in linear classifiers through outliers detection. They used a small fraction of trusted samples to detect poisoning samples through a distance-based anomaly detection method. Diakonikolas et al. [29] detected poisoning samples through the singular value decomposition of the gradients.

Although these countermeasures can alleviate the influence of poisoning attacks to a certain extent, they have some obvious defects. First, there is a high probability of removing normal samples when removing malicious samples. It undoubtedly affects the accuracy of trained models. Second, with the increasing amount of data, the cost of examining every training sample is increasing. Third, these countermeasures are limited on the forms of the training samples. The effectiveness of these countermeasures is limited when dealing with high dimensional training data.

**Model-driven countermeasure.** This class of countermeasure is often carried out during the model training. Defenders could modify the learning algorithms to enhance their robustness against poisoning samples. Therefore, most countermeasures during training are algorithm-specific. For example, Stempfel et al. [118] proposed a countermeasure, SloppySvm, to train a robust SVM to address the problem of learning with label flipping in a binary classification task.

SloppySvm minimizes a tailored nonconvex functional that is shown to be a uniform estimate of the noise-free SVM functional. The authors in [10] investigated the impact of label manipulation samples on SVMs. They proposed a label noise robust SVMs method by adding a correction term to the kernel matrix of a SVM. Furthermore, Yang et al. [145] developed a multiple kernel learning for label flipping dataset through stochastic programming. Liu et al. [70] proposed a framework for high-dimensional regression. They tried to protect the high-dimensional regression in both dimension reduction stage and linear regression model training stage.

As discussed in Section 2, poisoning attacks is a data-driven threat. Therefore, it is not sufficient to prevent poisoning attacks only by modifying specific algorithms. Countering poisoning attacks from a unified perspective has become a research focus. Natarajan et al. [88] proposed two countermeasures against the class-conditional random label manipulation from the perspective of loss functions. In their first countermeasure, they used an unbiased estimator of the loss function. The second countermeasure is based on the difference between noisy distribution and clean distribution around the threshold. They proposed a weighted loss function to correct the threshold. Since no specific algorithm is modified, their method can be applied to a variety of learning algorithms. Rosenfeld et al. [104] proposed a randomized-smoothing based framework for building classification models that are certifiably robust to poisoning attacks. Although they claimed that their method can counter various poisoning attacks, only label-flipping methods are evaluated in their paper.

The aforementioned works are aimed at simple attack methods. However, in real scenarios, the potential attack methods are almost limitless. Are there countermeasures that are robust to a large class of data poisoning attacks? To answer this question, Steinhardt et al. [117] proposed a framework to study the space of attacks against a given defense. They focused on two types of countermeasures, eliminating outliers and minimizing a margin-based loss on the remaining data. They presented an approximate upper bound on the worst-case test loss of an attack for the design of a countermeasure. Park et al. [95] proposed a metric for quantifying the resilience of learning algorithms to assist the design of resilient learning algorithms. However, this metric can only work in binary linear classification algorithms.

*6.1.2 Countermeasures in Deep Learning.* Poisoning attacks are more deceptive and harder to detect in deep learning, since adversaries are no longer satisfied with destroying the target model. In this subsection, we analyze countermeasures in deep learning scenario from the perspective of poisoning techniques. Some representative countermeasures are listed in Table 5.

**Counter label manipulation based poisoning attacks.** Similar to poisoning attacks in conventional machine learning, label manipulation is also one of the important techniques of carrying out poisoning attacks in deep learning. In the deep learning scenario, the learning task is more complex, which requires more training data. In most situations, training data are collected from various sources and larger data scales increase the risk of datasets being poisoned.

To deal with the problem of mislabeling in a large dataset, an intuitive idea is pre-processing the dataset before training. For example, Li et al. [67] proposed a method that distills the knowledge in the small clean dataset to facilitate learning a better model from the entire noisy dataset. Moreover, in deep learning, several key points can be modified to improve the robustness of the model besides training data. Deep learning models are composed of multiple layers, each with its unique functions. Therefore, the robustness of a deep learning model can be improved by adding carefully designed layers to its architecture. Goldberger and Ben-Reuven [38] designed a noisy-label deep learning architecture, which is based on a concatenation of softmax layers. Xiao et al. [135] integrated a label noise model layer into a deep learning framework. The label noise model layer is used to estimate a posterior distribution of the true label, which is then used to supervise the model training.

Table 5. Comparison of Countermeasures Against Poisoning Attacks in Deep Learning

| Literature | Strategy | Adversarial Setting | | Year |
|---|---|---|---|---|
| | | Technique | Goal | |
| [67] | Distill the knowledge from clean dataset. | LM | UnT | 2017 |
| [76] | Control update steps of model training. | LM | UnT | 2017 |
| [97] | Correct loss during training. | LM | UnT | 2017 |
| [140] | An information-theoretic loss function. | LM | UnT | 2019 |
| [73] | Training with a family of loss functions. | LM | UnT | 2020 |
| [122] | Label-based and clustering-based semi-supervised defenses. | LM | UnT | 2020 |
| [153] | A community-preserving self-supervised task as regularization. | LM | UnT | 2020 |
| [71] | Combine pruning and fine-tuning. | DM | BD | 2018 |
| [124] | Identify poisoning samples by spectral signatures. | DM | BD | 2018 |
| [20] | Clustering activations of the last hidden neural network layer. | DM | BD | 2019 |
| [126] | Consider minimal pixels to change prediction as detect feature. | DM | BD | 2019 |
| [101] | Detect poisoning samples according to neighbors in the feature space. | DM | T | 2020 |
| [157] | Utilize an inferred social graph to find poisoning data. | DM | UnT | 2020 |
| [15] | Diminish the impact of poisoning samples by strong data augmentations. | DM | BD\|T | 2021 |

UnT: Untargeted poisoning attack, T: Targeted poisoning attack, BD: Backdoor poisoning attack, LM: label manipulation, DM: data manipulation.

In addition to architectures, loss functions are also an important element of deep learning. Different loss functions can achieve different tasks and have different features on robustness. Ghosh et al. [37] studied some of the widely used loss functions in deep learning. They found that the loss function based on the mean absolute value of error is inherently robust to label noise. Some researchers tried to improve the robustness of deep learning through a sophisticated design loss function. For example, Xu et al. [140] proposed an information-theoretic loss function, which is based on the observation that the poisoning samples have no information about clean samples. Liu and Guo [73] proposed a peer loss function which work within the standard empirical risk minimization framework.

Another key point is the optimization of loss functions. Loss functions provide the objective of the training process, and optimization methods is the way to the objective. Therefore, optimization methods also affect the performance of the trained model to a great extent. Some research tried to improve robustness through modifying optimization methods. Malach et al. [76] proposed a method to train a model dynamically. The parameters of the model are rapidly updated at the beginning, and then the model is carefully updated when it is close to convergence. Specifically, they train two deep learning models, and update them only on examples that they disagree with. In this way, wrong samples do not affect both models, and the better model is the final model. Patrini et al. [97] proposed a loss correction method which has two procedures. In a more adversarial situation, Hendrycks et al. [48] proposed a loss correction method by utilizing information from a trusted small dataset.

In addition to the aforementioned countermeasures, some researchers have also tried to expand the protected domains. Taheri et al. [122] investigated this problem in Android malware detection domain. They first proposed a silhouette clustering-based label flipping attack, which carefully selected samples to perform label flipping. Furthermore, they proposed two countermeasures: label-based semi-supervised defense and clustering-based semi-supervised defense. These two countermeasures can find the potential label flipped samples and then predict new labels for these samples. Zhang et al. [153] introduced the first study of adversarial label-flipping attacks on **graph neural networks (GNNs)**. They argued that over-fitting is the key reason for the vulnerability of GNNs to label-flipping attacks. Furthermore, they introduce a community-preserving self-supervised task as regularization to counter label-flipping attacks.

**Counter data manipulation based poisoning attacks.** Data manipulation is a more sophisticated technique for constructing poisoning samples, as discussed in Section 4.2. Compared with label manipulation, the poisoning samples constructed through data manipulation have two characteristics: more stealthy and more malicious.

An intuitive idea of countering poisoning attacks is to find a metric that can detect poisoning samples. Peri et al. [101] investigated countermeasures against data manipulation poisoning attacks. They observed that at higher layers of the neural networks, poisoning samples have different feature distributions compared with clean samples, and these features tend to be located near the distribution of the target class. Based on this observation, they proposed a Deep K-NN countermeasure, which can detect poisoning samples according to their k nearest neighbors in the feature space (activations of the penultimate layer of a neural network). Although carefully constructed poisoning samples cannot be sensed by human vision, due to the addition of malicious information, these poisoning samples have other abnormalities. Designing reasonable features to distinguish these anomalies is the key challenge for defenders, especially in countering backdoor attacks.

Backdoors are one of the serious consequences caused by data manipulation poisoning attacks. Unlike other types of poisoning attacks, backdoor attacks are more stealthy. The stealth of backdoor attacks are reflected in the model's performance and the similarity between poisoning samples and benign samples as demonstrated in Section 4.2. In the early days of backdoor attacks, triggers were naive and obvious [22, 42]. However, as the battle between the defenders and the adversaries intensifies, the trigger of backdoor attack is designed to be more and more difficult to detect [68, 108, 136].

The main goal of countermeasures against backdoor attacks is to detect backdoors. Detections can be carried out on the whole life cycle of a model, including data collection, training stage, and inference stage. For example, Tran et al. [124] took a black-box neural network with some designated learned representation as features to detect the backdoor poisoning samples. Chen et al. [20] proposed an activation clustering for backdoor detection. They observed that the activations of the last hidden neural network layer between clean data and poisoning data are different. Wang et al. [127] proposed two countermeasures for backdoor detection based on the reaction of neuron activations on different input perturbations.

The intuition of the above countermeasures is that neurons in backdoored model perform differently on trigger input and clean input. However, in many scenarios, the target model to be detected is a black box, and defenders cannot access the running status inside the model. Wang et al. [126] proposed a method to detect backdoor based only on the input-output of the target model. Their intuition is that the model with a backdoor is sensitive to input perturbations, and a small perturbation of the input can significantly affect the output of the model. They use an optimization algorithm to attack each output label to change the output prediction of the model. Then the pixel variation of the input that causes the prediction change is used as a feature to detect the backdoor in the target model. However, their method only considers the amount of changing pixels without considering the shape of changing pixels. In the design of trigger, the shape of trigger is the key to activate a backdoor. Therefore, the detection performance will increase if the shape of changing pixels is considered as a feature.

Existing countermeasures cannot detect all backdoors [126, 127, 131]. Therefore, in addition to detecting backdoors, another class of countermeasures attempts to diminish the threat of poisoning samples in the backdoor attacks. Eliminating the influence of poisoning samples can also be developed from two perspectives: training data and training algorithm. For example, Borgnia et al. [15] investigated the effects of multiple data augmentation methods against poisoning attacks. They found that data augmentation is effective in eliminating the impact of poisoning

Table 6. Comparison of Countermeasures Against Poisoning Attacks in Federated Learning

| Literature | Type | Strategy | Year |
|---|---|---|---|
| [13] | poisoning tolerance | Select one representative local model as the global model. | 2017 |
| [147] | poisoning tolerance | Aggregate each model parameter independently, removes largest and smallest of them. | 2018 |
| [82] | poisoning tolerance | Select several local models and use a variant of Trimmed Mean to aggregate. | 2018 |
| [63] | poisoning tolerance | Introduce a distance-based penalty into the loss function. | 2019 |
| [92] | poisoning tolerance | Adjust the learning rate of the global model. | 2021 |
| [113] | poisoning detection | Find indicative features for malicious clients detection. | 2016 |
| [35] | poisoning detection | Evaluate the cosine similarity of the updates history. | 2018 |
| [158] | poisoning detection | Use GAN to generate data for updates evaluating. | 2019 |
| [155] | poisoning detection | Detect poisoning updates by local data of other clients. | 2020 |
| [64] | poisoning detection | Detect poisoning updates in low-dimensional latent feature space. | 2020 |
| [129] | poisoning detection | Detect poisoning updates via the coalitional game and Shapley value. | 2021 |
| [2] | poisoning detection | Validate the global model on local data. | 2021 |

samples in both backdoor attacks and targeted poisoning attacks. However, training algorithm based countermeasures need more insight on backdoor attacks. Liu et al. [71] proposed a fine-pruning countermeasure, which prunes the neurons of the backdoored model that are dormant on clean inputs and then fine-tunes the pruned network. Their intuition is that performance of different neurons of backdoored models are different.

Most of these attacks are carried out in the computer vision domain. In computer vision, people can make judgments directly about raw data. However, in other domains, people's perceptions are limited, which is beneficial to adversaries because they no longer have to carefully construct poisoning samples that are hard to detect with human vision. Therefore, it is necessary and meaningful to study poisoning attacks in these abstract domains. Zhao et al. [157] investigated data manipulation poisoning attacks in Mobile-Edge Computing. They first used a feature learning model to discover the social relationship among users based on geographical location and generate a social graph, and then searched the optimal map among users and the social network to determine if they are poisoned.

## 6.2 Countermeasures in Federated Learning

*6.2.1 Poisoning Tolerance Countermeasures.* In federated learning, it is usually assumed that an adversary can launch a poisoning attack through several compromised clients, and the server is the role that performs countermeasures. Therefore, most of the countermeasures are implemented on the server-side. After receiving updates from clients, the server implements countermeasures to reduce the impact of poisoning updates on the global model. Some representative countermeasures are listed in Table 6.

Krum [13], proposed by Blanchard et al., is the first provably Byzantine-resilient algorithm for distributed SGD. Krum is straightforward. It selects one local model that is similar to other models as the global model. The intuition is that the negative impact of this selected centroid model could be constrained since it is similar to other local models. However, Krum has some obvious flaws. When the number of malicious clients in the learning system is more than half, Krum cannot achieve an effective defense. Moreover, due to the high dimension of the model, adversaries can make large manipulations to a parameter without having a considerable impact on the Euclidean distance of their updates.

Trimmed Mean [147] is from the perspective of model parameters. Its intuition is that the malicious parameters tend to be far from the benign parameters. This algorithm aggregates each model parameter independently. For each parameter, the aggregator removes outliers of them, and computes the mean of the rest of the parameters as the global model.

Mhamdi et al. [82] proposed a countermeasure that combines Krum and a variant of Trimmed Mean, called Bulyan. Bulyan first iteratively applies Krum to select several local models. Then, Bulyan utilizes a variant of Trimmed Mean to aggregate these local models. This design greatly

reduces the probability of selecting malicious parameters. At the same time, even if malicious parameters are selected, their influence can be reduced by Trimmed Mean aggregation.

Although the aforementioned statistical-based countermeasures have a certain defensive ability against straightforward poisoning attacks, it is too stretched for the carefully constructed poisoning updates. As shown in Table 3, there are a large number of attack methods that can destroy the aforementioned three countermeasures. The problem is that all of these countermeasures assume that a poisoning update is statistically different from a benign update, which is not true in many cases. In real scenarios, poisoning updates will change according to different training data, so the defender needs a more dynamic countermeasure.

Instead of focusing on the statistical difference between poisoning updates and benign updates, some research tried to improve Byzantine-robust by modifying the aggregation algorithm. Li et al. [63] proposed a Byzantine-**robust stochastic aggregation method (RSA)**. RSA introduces a distance-based penalty into the distributed SGD loss function. This penalty forces every local update to be close to the global model. Therefore, the impact of arbitrary poisoning updates on global model is limited in a predefined range. Similarly, learning rate could be also modified to improve the robustness against poisoning attacks. Ozdayi et al. [92] proposed a countermeasure called robust learning rate, which adjusts the learning rate of the global model against backdoor poisoning attacks. For every dimension where the sum of signs of updates is less than a preset parameter, the learning rate is multiplied by -1. Different dimensions have different learning rates. The intuition is that the direction of poisoning dimensions in the whole learning system is different from that of benign dimensions. Therefore, the sum of poisoning dimensions is less than that of benign dimensions.

*6.2.2 Poisoning Detection Countermeasures.* In addition to improving the robustness of the aggregation algorithm to poisoning updates, there is a more direct defense strategy, malicious clients detection. Shen et al. [113] proposed the malicious clients detection method Auror. It first groups the features into different clusters and calculates the distance between centers of clusters to decide whether a particular feature could be used for detection. It then detects malicious clients based on the anomalous distribution of the indicative features. Similarly, Li et al. [64] proposed abnormal model updates detection methods based on their low-dimensional embeddings, in which the noisy and irrelevant features are removed whilst the essential features are retained. However, this countermeasure is still statistical-based, due to the lack of ground-truth of benign updates. Based on the coalitional game and Shapley value, Xi et al. [129] proposed a real-time backdoor detection system on federated learning in e-health.

Fung et al. [35] proposed FoolsGold, a countermeasure against sybil poisoning attacks. Their intuition is that clients' data in federated learning is non-iid, and these clients tend to be maliciously controlled by an adversary, when multiple clients behave similarly. FoolsGold evaluates the cosine similarity of the updates history of different clients, and clients are considered to be sybils when their updates history are too similar. However, FoolsGold relies too much on the setting of non-iid and sybil poisoning attacks. In real scenarios, poisoning attacks are often various, and as long as there is a little control over poisoning updates, FoolsGold can be bypassed. The above countermeasures are all unsupervised. Due to the lack of ground-truth as a criterion, these countermeasures can only rely on the characteristics summarized by defenders' research on previous poisoning attacks. These countermeasures are effective when the adversary's behavior conforms to their assumptions. However, in practice, the facts are the opposite, adversaries often attack from unexpected angles.

In a federated learning system, the benign clients are the silent majority, neither adversaries nor defenders can ignore their influence. Some researchers try to utilize the local data of benign clients

for better countermeasures performances. Zhao et al. [155] proposed a more sophisticated countermeasure which utilizes the local data of clients for poisoning updates detection. Their method tries to solve the challenge of lacking ground-truth in the server-side. The key idea is to realize client-side cross-validation, where each update is evaluated over other clients' local data. This solves the problem of lack of validation data to some extent, but the validation results are still vulnerable to malicious clients. With a similar intuition, BAFFLE [2] also used a set of validating clients to determine if the update of the global model in a round is poisoned. In addition to using the clients' local data, generation algorithms (e.g., GAN) can be used to generate data for evaluating updates [158].

## 7 DISCUSSIONS AND FUTURE RESEARCH DIRECTIONS

In the research domain of poisoning attacks, adversaries and defenders are always diametrically opposed. Their battle makes the research on poisoning attacks more in-depth. Adversaries try to improve the performance of poisoning attacks in three dimensions, stealth, efficiency, and robustness. Defenders' main goal is to prevent as many unknown poisoning attacks as possible while accurately preventing known ones. In this section, we will first discuss the potential reasons why a poisoning attack is feasible. Then, we introduce the status quo of poisoning attacks in other distributed learning systems. Finally, we present current gaps and future research directions from the perspective of both adversaries and defenders.

### 7.1 Reasoning Poisoning Attacks

In this subsection, we explore the potential reasons why poisoning attacks can be successful. To understand the vulnerability of machine learning method, we first illustrate the knowledge flow during model training. In general, model training is a process distilling knowledge. Machine learning methods first need to convert semantic knowledge that humans can understand into feature vectors that can be understood by machines, such as Word2Vec [83] and GloVe [100]. A considerable amount of background information is squeezed out in the process. After that, learning algorithms learn knowledge from a finite training dataset. The learning algorithm can only summarize the knowledge from the training data, which also leads to the loss of some semantic features.

With so much knowledge lost, the criterion of the trained model's prediction is only a projection of the real world, which is easily affected by the training dataset. Dong et al. [30] adopted class activation mapping to visualize the attention maps of three normally trained models. The results show that the prediction criterion of well-trained deep learning models is only part of the objects in the images. For example, a model would identify a church only based on the pointed roof, while ignoring other structures of the building [121]. This is caused by the lack of human-like background knowledge of the model. Humans' logic for identifying a church is to first judge whether an object is a building, and then judge whether it is a church. However, before the training, the learning algorithms have no semantic background knowledge of churches. They do not know that a church is a building. They can only learn the most prominent features of the church, which are often unreliable. For example, the pointed object may not be the roof but a part of an airplane. The prediction criterion learned by the model is closely related to other classes in the training dataset. When there are similar features of multiple classes of objects in the training dataset, the model often does not consider these features as the basis for prediction. Meanwhile, maliciously introduced poisoning knowledge can be easily learned by machine learning methods [150, 151].

The situation is complex in targeted poisoning attacks, especially backdoor poisoning attacks. The targeted poisoning attack can be regarded as a multi-task problem. The main task is to learn knowledge of benign classes, and the malicious task is to learn the poisoning knowledge introduced by the adversary. Different from the normal learning process, in poisoning attack, the adversary desires over-fitting on the malicious task, which is much easier than generalization.

For example, Carlini et al. [18] inserted some information that is not related to the main task in the training dataset to test whether the model remembers some sensitive information that should not be memorized. They found that unintended memorization is commonplace, e.g., a word-level language model can remember a random number in the training set. Moreover, traditional regularization methods, such as weight decay and dropout, show a limited effect on preventing unintended memorization.

Targeted poisoning attacks make clever use of learning algorithm's natural tendency to overfitting. The training data of current learning models are often complex and high-dimensional, which contains a large amount of redundant space for implanting poisoning knowledge. This knowledge does not conflict with other benign data, it just broadens the space of existing datasets [126]. Moreover, since there is no generalization requirement, the implanted poisoning knowledge can be more concentrated and stable. For example, the triggers in the backdoor attack are all stable patterns.

### 7.2 Discussion: Other Distributed Machine Learning

Distributed learning frameworks have several other paradigms besides federated learning, e.g., **Peer-to-Peer (P2P)** learning and **split learning (SL)** [7]. Their security issues have not received widespread attention, due to the limited application range and the lack of a fixed standard. In this subsection, we discuss several representative technologies in the hope of serving as a potential research topic.

**Peer-to-Peer learning.** There is no central server in P2P learning. In contrast, the participants communicate their training results directly with each other. For example, Shayan et al. [112] proposed Biscotti, a P2P learning method with a blockchain ledger. Clients join Biscotti and contribute to a ledger to train a global model, under the assumption that peers are willing to collaborate on building machine learning models while unwilling to share their data. There is no server to coordinate the training of the model in P2P learning, thus the problem of communication between participants and the training synchronization is challenging. Biscotti leverages blockchain ledger to solve the challenge. Other P2P learning methods, such as graph based framework BrainTorrent [105] and Gossip Learning [46, 47] proposed their own solutions.

The advantage brought by eliminating servers is obvious. It frees the clients from the concern of privacy leakage brought on by untrusted servers. However, from the security perspective, the absence of a server also makes the training process more susceptible to poisoning attacks. To the best of our knowledge, at present, the poisoning attack against P2P learning has not received enough attention. Researchers regard P2P learning as a supplement to client-server architecture federated learning, expecting to solve the problems in the existing federated learning framework through P2P learning, while ignoring the potential security risks of P2P learning itself. Adversaries can implement poisoning attacks against P2P learning using similar techniques as the poisoning attack in federated learning, since there is no essential difference in the communication contents. From the defender's perspective, due to the lack of the server-side control during training, it is more difficult to counter poisoning attacks in P2P learning.

**Split Learning.** In addition to leverage distributed training data, the model can also be split. The training participants hold replications of the shallower layers of the model, and a central entity holds the deeper layers. Inter-layer values, i.e., activations and gradients exchange occurs between a certain participant and the central entity [31, 43, 96]. Split learning is also a client-server architecture. The difference between SL and FL lies in the communication content between clients and the server. In split learning, clients are responsible for training the shallower layers of the model, and update the outputs of shallower layers to the server. Then, the server will perform subsequent training.

Changes in the content of communications may also affect attacks. The updates of SL could be considered as a fragment of updates in FL. Therefore, the effectiveness of the aforementioned attack methods may diminish due to the lack of full poisoning information. However, it is unclear to what extent the aforementioned attack methods are still effective. More research is required in this topic.

### 7.3 Future Research Directions: Perspective of Attacks

During the battle, adversaries have always been the dominant party. They only need to break the most vulnerable point to compromise the entire learning model. Therefore, their goal is no longer satisfied with only destroying the target model, but how to improve the stealth, efficiency, and robustness of the attack.

*7.3.1 Stealth and Robustness of Backdoor Poisoning Attacks.* Stealth and robustness for backdoor poisoning attacks can be improved from two perspectives. The first one is the design of triggers. The triggers in the existing methods are relatively straightforward, e.g., a fixed pixel arrangement. Even if some attack methods try to hide triggers, these efforts stay at only the visual level [22, 108]. Adversaries often choose patterns that can be directly observed as triggers. However, any fixed pattern can be a trigger of backdoor poisoning attacks (e.g., modern language models can distinguish between texts generated by different language models, and this difference could be used as a trigger [65]).

The second one is the construction of poisoning samples. A clever way to construct samples containing trigger can can also improve the stealth of backdoor poisoning attacks. The poisoning samples generated by optimizing objective functions [11, 84, 111] or generative models [85, 142] are visually indistinguishable from benign samples. However, few countermeasures are based on visual features, which makes these efforts contribute less to the improvement of stealth. We believe that more efforts could be made on other features (e.g., frequency features) rather than vision.

*7.3.2 Trade-off between Efficiency and Stealth of Poisoning Federated Learning.* Some adversaries consider federated learning as a sybil attack [3, 8, 32]. Although this setting does increase the success rate of the attack, it is difficult to compromise multiple clients of the same learning system in real scenarios. Moreover, most existing sybil attacks are naive in controlling the action of malicious clients. The malicious clients behave identically during attacking, to counter the impact of benign clients. However, the similar behavior of malicious clients increases the possibility of attack exposure [35].

Furthermore, the existing attack methods are almost static. Attacks are carried out at a predetermined pace. However, the state of global model and benign clients are dynamic during the training process [92]. In the training process, the global model is not updated at a fixed rate, and the update is usually large at the initial stage, while small when it is close to convergence. Therefore, if the state of the system during the training process is not considered, the possibility of attack exposure will increase. For example, if the poisoning update is still large when the model tends to converge, the server can easily find this anomaly.

Future research on poisoning federated learning can be focused on a more restricted threat model. For example, reducing the number of malicious clients during the attack. It is a challenge to balance efficiency and stealth when the number of malicious clients is limited. Adversaries can try to manipulate a small number of malicious clients to exhibit different behaviors, and then gradually affect the global model. At the same time, adversaries should also pay more attention to the changes in the training phase of the entire system, which can be used to hide the traces of attacks.

*7.3.3 Expanding Poisoning Attacks.* Current studies of poisoning attacks are mainly concentrated in the computer vision domain [22, 57, 111]. However, the risks of machine (deep) learning in other scenarios are also of concern. For example, social media is one of the fastest growing fields in recent years. Its prosperity has spurred the development of machine learning in this direction. GNN is used to analyze the behavior of users on social networks [14, 130, 139], and other learning algorithms are also used to analyze short texts published on social media [49, 128]. However, the security issues of learning models have not received the same attention along with their widespread application, besides the computer vision domain. Once these models are compromised, the impact will be far beyond imagination. Therefore, we argue that researchers could pay more attention to poisoning attacks in scenarios such as social media.

The current poisoning attacks have only one goal, which is to perturb the predicted results of specific samples. However, another vulnerability of the learning model, the privacy of training data, has hardly been discussed in poisoning attacks. Security and privacy issues are almost always discussed separately in existing research. However, these two issues have some commonalities. For example, poisoning attacks take advantage of the model's tendency to over-fitting, and over-fitting can lead to a certain degree of privacy leakage [18, 87, 109]. Therefore, if the goals of poisoning attacks can be expanded to privacy issues, it can better help us understand the learning model. For example, an adversary can implant a backdoor into the model through a poisoning attack. This backdoor can make the target model remember more unique information of training samples to assist in divulging private information.

### 7.4 Future Research Directions: Perspective of Countermeasures

*7.4.1 Gaps in the Assumptions.* Most current countermeasures limit the ability of their adversaries, e.g., most robust deep learning algorithms assume that the poison is label noise [37, 70, 135]. However, the label noise introduced by random label-flipping is usually non-malicious or low-malicious. In the Reference [29], the authors mentioned that the data implanted by adversaries were highly correlated, and could have a complex internal structure that is difficult to model. It suggested that current countermeasures can improve the robustness on naive attacks, however, they are less effective on countering more complex poisoning attacks. Meanwhile, the assumptions of adversaries in current countermeasures are mostly undisguised attacks [82, 147]. Therefore, although they have achieved positive results in their experiments, they are vulnerable when countering more elaborate poisoning attacks [32, 136].

Another problem of existing countermeasures is that they are largely attack-specific [13, 71]. They can only defend against known attack methods. As suggested in Table 3, once the adversary knows the existence of these countermeasures, it is easy to bypass them. In general, the countermeasures against poisoning attacks are at a relatively disadvantaged position. Improving the effectiveness of countermeasures should start from freeing the assumptions on adversaries, e.g., giving the adversary the white-box access and the stealthy poisoning methods. Only through confrontation with a tricky adversary can we design a more complete countermeasure.

*7.4.2 Metrics of Detecting Poisoning Attacks.* Researchers have found some reasonable metrics to measure the difference between poisoning samples and benign samples, e.g., spectral signature [124] and activations of the last hidden layer [20]. These metrics are based on some observations, rather than systematic research on poisoning attacks. They are individual feature points. However, in many other adversarial research domains, defenders usually leverage multiple features to jointly detect malicious behavior (e.g., Android malware family classification [19] and encrypted malware traffic detection [1]). A similar manner can also be developed in poisoning attacks, defenders can

leverage multiple features to form a feature space that can describe the difference between the poisoning samples and the benign samples more comprehensively.

## 8 SUMMARY

This survey aims at offering a comprehensive and up-to-date overview of poisoning attacks and countermeasures in both centralized and federated learning. It provides a unified perspective to observe the existing poisoning attacks across different learning architectures. We categorize poisoning attacks from two dimensions: the goal of the attack and the poisoning technique. The differences and connections among different poisoning attacks are analyzed based on this taxonomy. The analysis has allowed us to highlight the potential vulnerabilities of conventional machine learning, deep learning, and federated learning. Based on our observations, poisoning attacks are developing in a direction that is more efficient, stealthier, and more robust. On the other hand, countermeasures are in a disadvantaged position in this battle. The existing countermeasures are still unable to effectively defend against known subtle poisoning attacks, and even less able to effectively deal with unknown threats.

We indicate the potential directions that adversaries may attack from. Adversaries have already solved the feasibility problem. Therefore, they may focus more on the trade-off between efficiency and stealth. Meanwhile, the challenges for defenders are much more severe, since it is still an open question to prevent or detect poisoning attacks effectively. We highlight several possible directions for defenders. We hope that this survey will provide the necessary background to help the community fight the battle against poisoning attacks.

## REFERENCES

[1] Blake Anderson and David A. McGrew. 2016. Identifying encrypted malware traffic with contextual flow data. In *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2016*. ACM, 35–46.

[2] Sébastien Andreina, Giorgia Azzurra Marson, Helen Möllering, and Ghassan Karame. 2021. BaFFLe: Backdoor detection via feedback-based federated learning. In *41st IEEE International Conference on Distributed Computing Systems, ICDCS 2021*. IEEE, 852–863.

[3] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How to backdoor federated learning. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020 (Proceedings of Machine Learning Research)*, Vol. 108. PMLR, 2938–2948.

[4] Marco Barreno, Blaine Nelson, Anthony D. Joseph, and J. D. Tygar. 2010. The security of machine learning. *Mach. Learn.* 81, 2 (2010), 121–148.

[5] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D. Joseph, and J. D. Tygar. 2006. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2006*. ACM, 16–25.

[6] Gilad Baruch, Moran Baruch, and Yoav Goldberg. 2019. A little is enough: Circumventing defenses for distributed learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*. 8632–8642.

[7] Paolo Bellavista, Luca Foschini, and Alessio Mora. 2021. Decentralised learning in federated deployment environments: A system-level survey. *ACM Comput. Surv.* 54, 1 (2021), 15:1–15:38.

[8] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin B. Calo. 2019. Analyzing federated learning through an adversarial lens. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019 (Proceedings of Machine Learning Research)*, Vol. 97. PMLR, 634–643.

[9] Battista Biggio, Samuel Rota Bulò, Ignazio Pillai, Michele Mura, Eyasu Zemene Mequanint, Marcello Pelillo, and Fabio Roli. 2014. Poisoning complete-linkage hierarchical clustering. In *Structural, Syntactic, and Statistical Pattern Recognition - Joint IAPR International Workshop, S+SSPR 2014 (Lecture Notes in Computer Science)*, Vol. 8621. Springer, 42–52.

[10] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2011. Support vector machines under adversarial label noise. In *Asian Conference on Machine Learning*. PMLR, 97–112.

[11] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2012. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*. icml.cc/Omnipress.

[12] Battista Biggio, Konrad Rieck, Davide Ariu, Christian Wressnegger, Igino Corona, Giorgio Giacinto, and Fabio Roli. 2014. Poisoning behavioral malware clustering. In *Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop, AISec 2014*. ACM, 27–36.

[13] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*. 119–129.

[14] Stephen P. Borgatti, Martin G. Everett, and Jeffrey C. Johnson. 2018. *Analyzing Social Networks*. Sage.

[15] Eitan Borgnia, Valeriia Cherepanova, Liam Fowl, Amin Ghiasi, Jonas Geiping, Micah Goldblum, Tom Goldstein, and Arjun Gupta. 2021. Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2021*. IEEE, 3855–3859.

[16] Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *19th International Conference on Computational Statistics, COMPSTAT 2010*. Physica-Verlag, 177–186.

[17] Di Cao, Shan Chang, Zhijian Lin, Guohua Liu, and Donghong Sun. 2019. Understanding distributed poisoning attack in federated learning. In *25th IEEE International Conference on Parallel and Distributed Systems, ICPADS 2019*. IEEE, 233–239.

[18] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium, USENIX Security 2019*. USENIX Association, 267–284.

[19] Tanmoy Chakraborty, Fabio Pierazzi, and V. S. Subrahmanian. 2020. EC2: Ensemble clustering and classification for predicting Android malware families. *IEEE Trans. Dependable Secur. Comput.* 17, 2 (2020), 262–277.

[20] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian M. Molloy, and Biplav Srivastava. 2019. Detecting backdoor attacks on deep neural networks by activation clustering. In *Workshop on Artificial Intelligence Safety 2019 co-located with the Thirty-Third AAAI Conference on Artificial Intelligence 2019 (AAAI-19) (CEUR Workshop Proceedings)*, Vol. 2301. CEUR-WS.org.

[21] Jian Chen, Xuxin Zhang, Rui Zhang, Chen Wang, and Ling Liu. 2021. De-Pois: An attack-agnostic defense against data poisoning attacks. *IEEE Trans. Inf. Forensics Secur.* 16 (2021), 3412–3425.

[22] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *CoRR* abs/1712.05526 (2017).

[23] Gabriela F. Cretu, Angelos Stavrou, Michael E. Locasto, Salvatore J. Stolfo, and Angelos D. Keromytis. 2008. Casting out demons: Sanitizing training data for anomaly sensors. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE, 81–95.

[24] Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. 2008. Supervised learning. In *Machine Learning Techniques for Multimedia*. Springer, 21–49.

[25] Zihang Dai, Zhilin Yang, Fan Yang, William W. Cohen, and Ruslan Salakhutdinov. 2017. Good semi-supervised learning that requires a bad GAN. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*. 6510–6520.

[26] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. 2010. The YouTube video recommendation system. In *Proceedings of the Fourth ACM Conference on Recommender Systems*. 293–296.

[27] Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli. 2019. Why do adversarial attacks transfer? Explaining transferability of evasion and poisoning attacks. In *28th USENIX Security Symposium, USENIX Security 2019*. USENIX Association, 321–338.

[28] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'09)*. IEEE Computer Society, 248–255.

[29] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. 2019. Sever: A robust meta-algorithm for stochastic optimization. In *International Conference on Machine Learning*. PMLR, 1596–1606.

[30] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. 2019. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*. Computer Vision Foundation/IEEE, 4312–4321.

[31] Ege Erdogan, Alptekin Küpçü, and A. Ercüment Çiçek. 2021. SplitGuard: Detecting and mitigating training-hijacking attacks in split learning. *IACR Cryptol. ePrint Arch.* (2021), 1080.

[32] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. 2020. Local model poisoning attacks to Byzantine-robust federated learning. In *29th USENIX Security Symposium, USENIX Security 2020*. USENIX Association, 1605–1622.

[33] Jiashi Feng, Huan Xu, Shie Mannor, and Shuicheng Yan. 2014. Robust logistic regression and classification. *Advances in Neural Information Processing Systems* 27 (2014), 253–261.
[34] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon M. Lin, David Page, and Thomas Ristenpart. 2014. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *Proceedings of the 23rd USENIX Security Symposium*. USENIX Association, 17–32.
[35] Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. 2020. The limitations of federated learning in sybil settings. (2020), 301–316.
[36] Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Jiliang Zhang, Anmin Fu, Surya Nepal, and Hyoungshick Kim. 2020. Backdoor attacks and countermeasures on deep learning: A comprehensive review. *arXiv preprint arXiv:2007.10760* (2020).
[37] Aritra Ghosh, Himanshu Kumar, and P. S. Sastry. 2017. Robust loss functions under label noise for deep neural networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI Press, 1919–1925.
[38] Jacob Goldberger and Ehud Ben-Reuven. 2017. Training deep neural-networks using a noise adaptation layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings*. OpenReview.net.
[39] Tom Goldstein, Christoph Studer, and Richard G. Baraniuk. 2014. A field guide to forward-backward splitting with a FASTA implementation. *CoRR* abs/1411.3406 (2014).
[40] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*. 2672–2680.
[41] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015*.
[42] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. BadNets: Identifying vulnerabilities in the machine learning model supply chain. *CoRR* abs/1708.06733 (2017).
[43] Otkrist Gupta and Ramesh Raskar. 2018. Distributed learning of deep neural network over multiple agents. *J. Netw. Comput. Appl.* 116 (2018), 1–8.
[44] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. Unsupervised learning. In *The Elements of Statistical Learning*. Springer, 485–585.
[45] Zecheng He, Tianwei Zhang, and Ruby B. Lee. 2019. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference*. 148–162.
[46] István Hegedüs, Árpád Berta, Levente Kocsis, András A. Benczúr, and Márk Jelasity. 2016. Robust decentralized low-rank matrix decomposition. *ACM Trans. Intell. Syst. Technol.* 7, 4 (2016), 62:1–62:24.
[47] István Hegedüs, Gábor Danner, and Márk Jelasity. 2019. Gossip learning as a decentralized alternative to federated learning. In *Distributed Applications and Interoperable Systems - 19th IFIP WG 6.1 International Conference, DAIS 2019 (Lecture Notes in Computer Science)*, Vol. 11534. Springer, 74–90.
[48] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. 2018. Using trusted data to train deep networks on labels corrupted by severe noise. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*. 10477–10486.
[49] Faliang Huang, Xuelong Li, Changan Yuan, Shichao Zhang, Jilian Zhang, and Shaojie Qiao. 2021. Attention-emotion-enhanced convolutional LSTM for sentiment analysis. *IEEE Transactions on Neural Networks and Learning Systems* (2021), 1–14. https://doi.org/10.1109/TNNLS.2021.3056664
[50] W. Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. 2020. MetaPoison: Practical general-purpose clean-label data poisoning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.
[51] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. 2018. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *2018 IEEE Symposium on Security and Privacy, SP 2018*. IEEE Computer Society, 19–35.
[52] Matthew Jagielski, Giorgio Severi, Niklas Pousette Harger, and Alina Oprea. 2020. Subpopulation data poisoning attacks. *CoRR* abs/2006.14026 (2020).
[53] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*. Computer Vision Foundation / IEEE, 4401–4410.
[54] Mehran Mozaffari Kermani, Susmita Sur-Kolay, Anand Raghunathan, and Niraj K. Jha. 2015. Systematic poisoning attacks on and defenses for machine learning in healthcare. *IEEE J. Biomed. Health Informatics* 19, 6 (2015), 1893–1905.
[55] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[56] Diederik P. Kingma and Max Welling. 2014. Auto-encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014*.
[57] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017 (Proceedings of Machine Learning Research)*, Vol. 70. PMLR, 1885–1894.
[58] Pang Wei Koh, Jacob Steinhardt, and Percy Liang. 2018. Stronger data poisoning attacks break data sanitization defenses. *CoRR* abs/1811.00741 (2018). http://arxiv.org/abs/1811.00741.
[59] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. 2017. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017*. OpenReview.net.
[60] Leslie Lamport, Robert E. Shostak, and Marshall C. Pease. 2019. The Byzantine generals problem. In *Concurrency: The Works of Leslie Lamport*. ACM, 203–226.
[61] Yann LeCun, D. Touresky, G. Hinton, and T. Sejnowski. 1988. A theoretical framework for back-propagation. In *Proceedings of the 1988 Connectionist Models Summer School*, Vol. 1. 21–28.
[62] Chaoran Li, Xiao Chen, Derui Wang, Sheng Wen, Muhammad Ejaz Ahmed, Seyit Camtepe, and Yang Xiang. 2021. Backdoor attack on machine learning based android malware detectors. *IEEE Transactions on Dependable and Secure Computing* (2021), 1–1. https://doi.org/10.1109/TDSC.2021.3094824
[63] Liping Li, Wei Xu, Tianyi Chen, Georgios B. Giannakis, and Qing Ling. 2019. RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*. AAAI Press, 1544–1551.
[64] Suyi Li, Yong Cheng, Wei Wang, Yang Liu, and Tianjian Chen. 2020. Learning to detect malicious clients for robust federated learning. *arXiv preprint arXiv:2002.00211* (2020).
[65] Shaofeng Li, Hui Liu, Tian Dong, Benjamin Zi Hao Zhao, Minhui Xue, Haojin Zhu, and Jialiang Lu. 2021. Hidden backdoors in human-centric language models. *CoRR* abs/2105.00164 (2021).
[66] Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. 2020. Backdoor learning: A survey. *CoRR* abs/2007.08745 (2020).
[67] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. 2017. Learning from noisy labels with distillation. In *IEEE International Conference on Computer Vision, ICCV 2017*. IEEE Computer Society, 1928–1936.
[68] Junyu Lin, Lei Xu, Yingqi Liu, and Xiangyu Zhang. 2020. Composite backdoor attack for deep neural network by mixing existing benign features. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 113–131.
[69] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing* 7, 1 (2003), 76–80.
[70] Chang Liu, Bo Li, Yevgeniy Vorobeychik, and Alina Oprea. 2017. Robust linear regression against training data poisoning. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. 91–102.
[71] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2018. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Research in Attacks, Intrusions, and Defenses - 21st International Symposium, RAID 2018 (Lecture Notes in Computer Science)*, Vol. 11050. Springer, 273–294.
[72] Tongliang Liu and Dacheng Tao. 2015. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 3 (2015), 447–461.
[73] Yang Liu and Hongyi Guo. 2020. Peer loss functions: Learning from noisy labels without knowing noise rates. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020 (Proceedings of Machine Learning Research)*, Vol. 119. PMLR, 6226–6236.
[74] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2018. Trojaning attack on neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018*. The Internet Society.
[75] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. 2017. Pose guided person image generation. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*. 406–416.
[76] Eran Malach and Shai Shalev-Shwartz. 2017. Decoupling "when to update" from "how to update". In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*. 960–970.
[77] Warren S. McCulloch and Walter Pitts. 1943. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics* 5, 4 (1943), 115–133.
[78] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017 (Proceedings of Machine Learning Research)*, Vol. 54. PMLR, 1273–1282.

[79] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*. PMLR, 1273–1282.
[80] Shike Mei and Xiaojin Zhu. 2015. Using machine teaching to identify optimal training-set attacks on machine learners. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI Press, 2871–2877.
[81] Tony A. Meyer and Brendon Whateley. 2004. SpamBayes: Effective open-source, Bayesian based, email classification system. In *CEAS 2004 - First Conference on Email and Anti-Spam, July 30–31, 2004*.
[82] El Mahdi El Mhamdi, Rachid Guerraoui, and S'ebastien Rouault. 2018. The hidden vulnerability of distributed learning in Byzantium. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018 (Proceedings of Machine Learning Research)*, Vol. 80. PMLR, 3518–3527.
[83] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Workshop Track Proceedings*.
[84] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C. Lupu, and Fabio Roli. 2017. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017*. ACM, 27–38.
[85] Luis Muñoz-González, Bjarne Pfitzner, Matteo Russo, Javier Carnerero-Cano, and Emil C. Lupu. 2019. Poisoning attacks with generative adversarial nets. *CoRR* abs/1906.07773 (2019).
[86] Kevin P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. MIT Press.
[87] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2018. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018*. ACM, 634–646.
[88] Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep K. Ravikumar, and Ambuj Tewari. 2013. Learning with noisy labels. *Advances in Neural Information Processing Systems* 26 (2013), 1196–1204.
[89] Blaine Nelson, Marco Barreno, Fuching Jack Chi, Anthony D. Joseph, Benjamin I. P. Rubinstein, Udam Saini, Charles Sutton, J. Doug Tygar, and Kai Xia. 2008. Exploiting machine learning to subvert your spam filter. In *First USENIX Workshop on Large-Scale Exploits and Emergent Threats, LEET'08*. USENIX Association.
[90] Andrew Newell, Rahul Potharaju, Luojie Xiang, and Cristina Nita-Rotaru. 2014. On the practicality of integrity attacks on document-level sentiment analysis. In *Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop, AISec 2014*. ACM, 83–93.
[91] James Newsome, Brad Karp, and Dawn Xiaodong Song. 2006. Paragraph: Thwarting signature learning by training maliciously. In *Recent Advances in Intrusion Detection, 9th International Symposium, RAID 2006 (Lecture Notes in Computer Science)*, Vol. 4219. Springer, 81–105.
[92] Mustafa Safa Özdayi, Murat Kantarcioglu, and Yulia R. Gel. 2021. Defending against backdoors in federated learning with robust learning rate. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*. AAAI Press, 9268–9276.
[93] Nicolas Papernot, Patrick McDaniel, Arunesh Sinha, and Michael P. Wellman. 2018. SoK: Security and privacy in machine learning. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 399–414.
[94] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2016. Practical black-box attacks against deep learning systems using adversarial examples. *CoRR* abs/1602.02697 (2016).
[95] Sangdon Park, James Weimer, and Insup Lee. 2017. Resilient linear classification: An approach to deal with attacks on training data. In *Proceedings of the 8th International Conference on Cyber-Physical Systems*. 155–164.
[96] Dario Pasquini, Giuseppe Ateniese, and Massimo Bernaschi. 2021. Unleashing the Tiger: Inference attacks on split learning. In *CCS'21: 2021 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2113–2129.
[97] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. IEEE Computer Society, 2233–2241.
[98] Andrea Paudice, Luis Muñoz-González, Andras Gyorgy, and Emil C. Lupu. 2018. Detection of adversarial training examples in poisoning attacks through anomaly detection. *arXiv preprint arXiv:1802.03041* (2018).
[99] Andrea Paudice, Luis Muñoz-González, and Emil C. Lupu. 2018. Label sanitization against label flipping poisoning attacks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 5–15.
[100] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*. ACL, 1532–1543.
[101] Neehar Peri, Neal Gupta, W. Ronny Huang, Liam Fowl, Chen Zhu, Soheil Feizi, Tom Goldstein, and John P. Dickerson. 2020. Deep k-NN defense against clean-label data poisoning attacks. In *Computer Vision - ECCV 2020 Workshops - Glasgow, UK, August 23–28, 2020, Proceedings, Part I (Lecture Notes in Computer Science)*, Vol. 12535. Springer, 55–70.
[102] Nikolaos Pitropakis, Emmanouil Panaousis, Thanassis Giannetsos, Eleftherios Anastasiadis, and George Loukas. 2019. A taxonomy and survey of attacks against machine learning. *Comput. Sci. Rev.* 34 (2019).

[103] Mauro Ribeiro, Katarina Grolinger, and Miriam A. M. Capretz. 2015. MLaaS: Machine learning as a service. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 896–902.

[104] Elan Rosenfeld, Ezra Winston, Pradeep Ravikumar, and Zico Kolter. 2020. Certified robustness to label-flipping attacks via randomized smoothing. In *International Conference on Machine Learning*. PMLR, 8230–8241.

[105] Abhijit Guha Roy, Shayan Siddiqui, Sebastian Pölsterl, Nassir Navab, and Christian Wachinger. 2019. BrainTorrent: A peer-to-peer environment for decentralized federated learning. *arXiv preprint arXiv:1905.06731* (2019).

[106] Benjamin I. P. Rubinstein, Blaine Nelson, Ling Huang, Anthony D. Joseph, Shing-hon Lau, Satish Rao, Nina Taft, and J. D. Tygar. 2009. ANTIDOTE: Understanding and defending against poisoning of anomaly detectors. In *Proceedings of the 9th ACM SIGCOMM Internet Measurement Conference, IMC 2009*. ACM, 1–14.

[107] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323, 6088 (1986), 533–536.

[108] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. 2020. Hidden trigger backdoor attacks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*. AAAI Press, 11957–11965.

[109] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2019. ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019*. The Internet Society.

[110] Arthur L. Samuel. 1959. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development* 3, 3 (1959), 210–229.

[111] Ali Shafahi, W. Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. 2018. Poison frogs! Targeted clean-label poisoning attacks on neural networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*. 6106–6116.

[112] Muhammad Shayan, Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. 2018. Biscotti: A ledger for private and secure peer-to-peer machine learning. *arXiv preprint arXiv:1811.09904* (2018).

[113] Shiqi Shen, Shruti Tople, and Prateek Saxena. 2016. Auror: Defending against poisoning attacks in collaborative deep learning systems. In *Proceedings of the 32nd Annual Conference on Computer Security Applications, ACSAC 2016*. ACM, 508–519.

[114] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy, SP 2017*. IEEE Computer Society, 3–18.

[115] Atul Singh, Tsuen-Wan Ngan, Peter Druschel, and Dan S. Wallach. 2006. Eclipse attacks on overlay networks: Threats and defenses. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies*. IEEE.

[116] Congzheng Song, Alexander M. Rush, and Vitaly Shmatikov. 2020. Adversarial semantic collisions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*. Association for Computational Linguistics, 4198–4210.

[117] Jacob Steinhardt, Pang Wei Koh, and Percy Liang. 2017. Certified defenses for data poisoning attacks. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*. 3517–3529.

[118] Guillaume Stempfel and Liva Ralaivola. 2009. Learning SVMs from sloppily labeled data. In *International Conference on Artificial Neural Networks*. Springer, 884–893.

[119] Octavian Suciu, Radu Marginean, Yigitcan Kaya, Hal Daumé III, and Tudor Dumitras. 2018. When does machine learning FAIL? Generalized transferability for evasion and poisoning attacks. In *27th USENIX Security Symposium, USENIX Security 2018*. USENIX Association, 1299–1316.

[120] Fnu Suya, Saeed Mahloujifar, Anshuman Suri, David Evans, and Yuan Tian. 2021. Model-targeted poisoning attacks with provable convergence. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021 (Proceedings of Machine Learning Research)*, Vol. 139. PMLR, 10000–10010.

[121] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014*.

[122] Rahim Taheri, Reza Javidan, Mohammad Shojafar, Zahra Pooranian, Ali Miri, and Mauro Conti. 2020. On defending against label flipping attacks on malware detection systems. *Neural Computing and Applications* 32, 18 (2020), 14781–14800.

[123] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. 2020. Data poisoning attacks against federated learning systems. In *Computer Security - ESORICS 2020 - 25th European Symposium on Research in Computer Security, ESORICS 2020 (Lecture Notes in Computer Science)*, Vol. 12308. Springer, 480–501.

[124] Brandon Tran, Jerry Li, and Aleksander Madry. 2018. Spectral signatures in backdoor attacks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*. 8011–8021.

[125] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. 2019. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing* (2019).

[126] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy, SP 2019*. IEEE, 707–723.

[127] Ren Wang, Gaoyuan Zhang, Sijia Liu, Pin-Yu Chen, Jinjun Xiong, and Meng Wang. 2020. Practical detection of trojan neural networks: Data-limited and data-free cases. In *Computer Vision - ECCV 2020 - 16th European Conference (Lecture Notes in Computer Science)*, Vol. 12368. Springer, 222–238.

[128] Xinzhi Wang, Luyao Kou, Vijayan Sugumaran, Xiangfeng Luo, and Hui Zhang. 2021. Emotion correlation mining through deep learning models on natural language text. *IEEE Transactions on Cybernetics* 51, 9 (2021), 4400–4413. https://doi.org/10.1109/TCYB.2020.2987064

[129] Binhan Xi, Shaofeng Li, Jiachun Li, Hui Liu, Hong Liu, and Haojin Zhu. 2021. BatFL: Backdoor detection on federated learning in e-Health. In *29th IEEE/ACM International Symposium on Quality of Service, IWQOS 2021*. IEEE, 1–10.

[130] Wenwen Xia, Yuchen Li, Jun Wu, and Shenghong Li. 2021. DeepIS: Susceptibility estimation on social networks. In *WSDM'21, The Fourteenth ACM International Conference on Web Search and Data Mining*. ACM, 761–769.

[131] Zhen Xiang, David J. Miller, and George Kesidis. 2020. Revealing backdoors, post-training, in DNN classifiers via novel inference on optimized perturbations inducing group misclassification. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020*. IEEE, 3827–3831.

[132] Huang Xiao, Battista Biggio, Gavin Brown, Giorgio Fumera, Claudia Eckert, and Fabio Roli. 2015. Is feature selection secure against training data poisoning? In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015 (JMLR Workshop and Conference Proceedings)*, Vol. 37. JMLR.org, 1689–1698.

[133] Huang Xiao, Battista Biggio, Blaine Nelson, Han Xiao, Claudia Eckert, and Fabio Roli. 2015. Support vector machines under adversarial label contamination. *Neurocomputing* 160 (2015), 53–62.

[134] Han Xiao, Huang Xiao, and Claudia Eckert. 2012. Adversarial label flips attack on support vector machines. In *ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track (Frontiers in Artificial Intelligence and Applications)*, Vol. 242. IOS Press, 870–875.

[135] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. 2015. Learning from massive noisy labeled data for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*. IEEE Computer Society, 2691–2699.

[136] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. 2020. DBA: Distributed backdoor attacks against federated learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020*. OpenReview.net.

[137] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. 2019. Fall of empires: Breaking Byzantine-tolerant SGD by inner product manipulation. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019 (Proceedings of Machine Learning Research)*, Vol. 115. AUAI Press, 261–270.

[138] Xinyu Xing, Wei Meng, Dan Doozan, Alex C. Snoeren, Nick Feamster, and Wenke Lee. 2013. Take this personally: Pollution attacks on personalized services. In *Proceedings of the 22nd USENIX Security Symposium*. USENIX Association, 671–686.

[139] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How powerful are graph neural networks?. In *7th International Conference on Learning Representations, ICLR 2019*. OpenReview.net.

[140] Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. 2019. L_DMI: A novel information-theoretic loss function for training deep nets robust to label noise. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*. 6222–6233.

[141] Yihao Xue, Chaoyue Niu, Zhenzhe Zheng, Shaojie Tang, Chengfei Lyu, Fan Wu, and Guihai Chen. 2021. Toward understanding the influence of individual clients in federated learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*. AAAI Press, 10560–10567.

[142] Chaofei Yang, Qing Wu, Hai Li, and Yiran Chen. 2017. Generative poisoning attack method against neural networks. *CoRR* abs/1703.01340 (2017).

[143] Guolei Yang, Neil Zhenqiang Gong, and Ying Cai. 2017. Fake co-visitation injection attacks to recommender systems. In *24th Annual Network and Distributed System Security Symposium, NDSS 2017*. The Internet Society.

[144] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.

[145] Tianbao Yang, Mehrdad Mahdavi, Rong Jin, Lijun Zhang, and Yang Zhou. 2012. Multiple kernel learning from noisy labels by stochastic programming. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*. icml.cc/Omnipress.

[146] Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, and Zhenkai Liang. 2019. Neural network inversion in adversarial setting via background knowledge alignment. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019*. ACM, 225–240.

[147] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter L. Bartlett. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018 (Proceedings of Machine Learning Research)*, Vol. 80. PMLR, 5636–5645.

[148] Xuefei Yin, Yanming Zhu, and Jiankun Hu. 2021. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM Comput. Surv.* 54, 6 (2021), 131:1–131:36.

[149] Chenyu You, Wenxiang Cong, Michael W. Vannier, Punam K. Saha, Eric A. Hoffman, Ge Wang, Guang Li, Yi Zhang, Xiaoliu Zhang, Hongming Shan, Mengzhou Li, Shenghong Ju, Zhen Zhao, and Zhuiyang Zhang. 2020. CT super-resolution GAN constrained by the identical, residual, and cycle learning ensemble (GAN-CIRCLE). *IEEE Trans. Medical Imaging* 39, 1 (2020), 188–203.

[150] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017*. OpenReview.net.

[151] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM* 64, 3 (2021), 107–115.

[152] Jiale Zhang, Bing Chen, Xiang Cheng, Huynh Thi Thanh Binh, and Shui Yu. 2021. PoisonGAN: Generative poisoning attacks against federated learning in edge computing systems. *IEEE Internet Things J.* 8, 5 (2021), 3310–3322.

[153] Mengmei Zhang, Linmei Hu, Chuan Shi, and Xiao Wang. 2020. Adversarial label-flipping attack and defense for graph neural networks. In *20th IEEE International Conference on Data Mining, ICDM 2020*. IEEE, 791–800.

[154] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. 2020. The secret revealer: Generative model-inversion attacks against deep neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*. IEEE, 250–258.

[155] Lingchen Zhao, Shengshan Hu, Qian Wang, Jianlin Jiang, Shen Chao, Xiangyang Luo, and Pengfei Hu. 2020. Shielding collaborative learning: Mitigating poisoning attacks through client-side detection. *IEEE Transactions on Dependable and Secure Computing* (2020), 1–1. https://doi.org/10.1109/TDSC.2020.2986205

[156] Mengchen Zhao, Bo An, Wei Gao, and Teng Zhang. 2017. Efficient label contamination attacks against black-box learning models. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017*. ijcai.org, 3945–3951.

[157] Ping Zhao, Haojun Huang, Xiaohui Zhao, and Daiyu Huang. 2020. P 3: Privacy-preserving scheme against poisoning attacks in mobile-edge computing. *IEEE Transactions on Computational Social Systems* 7, 3 (2020), 818–826.

[158] Ying Zhao, Junjun Chen, Jiale Zhang, Di Wu, Jian Teng, and Shui Yu. 2019. PDGAN: A novel poisoning defense method in federated learning using generative adversarial network. In *International Conference on Algorithms and Architectures for Parallel Processing*. Springer, 595–609.

[159] Renjie Zhou, Samamon Khemmarat, and Lixin Gao. 2010. The impact of YouTube recommendation system on video views. In *Proceedings of the 10th ACM SIGCOMM Internet Measurement Conference, IMC 2010*. ACM, 404–410.

[160] Chen Zhu, W. Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2019. Transferable clean-label poisoning attacks on deep neural nets. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019 (Proceedings of Machine Learning Research)*, Vol. 97. PMLR, 7614–7623.

[161] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*. 14747–14756.

Received 5 January 2022; revised 9 June 2022; accepted 18 July 2022