Towards Understanding and Enhancing Robustness of Deep Learning Models against Malicious Unlearning Attacks

Wei Qian*
Iowa State University
Ames, Iowa, USA
wqi@iastate.edu

Chenxu Zhao*
Iowa State University
Ames, Iowa, USA
cxzhao@iastate.edu

Wei Le
Iowa State University
Ames, Iowa, USA
weile@iastate.edu

Meiyi Ma
Vanderbilt University
Nashville, Tennessee, USA
meiyi.ma@vanderbilt.edu

Mengdi Huai
Iowa State University
Ames, Iowa, USA
mdhuai@iastate.edu

ABSTRACT

Given the availability of abundant data, deep learning models have been advanced and become ubiquitous in the past decade. In practice, due to many different reasons (e.g., privacy, usability, and fidelity), individuals also want the trained deep models to forget some specific data. Motivated by this, machine unlearning (also known as selective data forgetting) has been intensively studied, which aims at removing the influence that any particular training sample had on the trained model during the unlearning process. However, people usually employ machine unlearning methods as trusted basic tools and rarely have any doubt about their reliability. In fact, the increasingly critical role of machine unlearning makes deep learning models susceptible to the risk of being maliciously attacked. To well understand the performance of deep learning models in malicious environments, we believe that it is critical to study the robustness of deep learning models to malicious unlearning attacks, which happen during the unlearning process. To bridge this gap, in this paper, we first demonstrate that malicious unlearning attacks pose immense threats to the security of deep learning systems. Specifically, we present a broad class of malicious unlearning attacks wherein maliciously crafted unlearning requests trigger deep learning models to misbehave on target samples in a highly controllable and predictable manner. In addition, to improve the robustness of deep learning models, we also present a general defense mechanism, which aims to identify and unlearn effective malicious unlearning requests based on their gradient influence on the unlearned models. Further, theoretical analyses are conducted to analyze the proposed methods. Extensive experiments on real-world datasets validate the vulnerabilities of deep learning models to malicious unlearning attacks and the effectiveness of the introduced defense mechanism.

*The first two authors contribute equally to this work.

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08.
https://doi.org/10.1145/3580305.3599526

CCS CONCEPTS

• **Security and privacy**; • **Computing methodologies** → **Machine learning**;

KEYWORDS

Deep learning; data deletion; malicious attacks; security and privacy

ACM Reference Format:
Wei Qian, Chenxu Zhao, Wei Le, Meiyi Ma, Mengdi Huai. 2023. Towards Understanding and Enhancing Robustness of Deep Learning Models against Malicious Unlearning Attacks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3580305.3599526

1 INTRODUCTION

Deep Neural Networks (DNNs) are powerful and efficient frameworks for visual learning and have been extended to diversified architectures. Patterns and features of big data can be learned automatically and efficiently through DNNs. In recent years, DNNs have achieved state-of-the-art results on challenging real-world problems such as image classification [64, 91], autonomous deriving [55, 79], natural language processing [17, 23], recommendation [14, 89], cancer diagnosis and prognosis prediction [16, 46, 97].

In practice, individuals may choose to have their data completely removed from the trained deep learning models due to many reasons, such as privacy, usability, and fidelity [33, 59, 90]. Particularly, recent regulations (e.g., the California Consumer Privacy Act [60] and the former Right to be Forgotten [26]) now also compel organizations to give individuals "the right to be forgotten", i.e., the right to have all or part of their data deleted from a well-built system upon request. The most straightforward approach is to retrain the model on all data except the requested unlearning data to be removed, but this approach is in general impractical for deep learning models since the entire training set is usually very large. In addition, although retraining deep models in some cases is a feasible solution, frequent data removal requests inevitably put enormous computational pressure on the infrastructures responsible for real-time services. Hence, effectively eliminating the contributions of the requested data while preserving model performance is a critical and challenging research question.

In the literature, extensive research works [4, 6, 10, 57, 58, 67, 75, 85] have been proposed to allow individuals the possibility and

flexibility to completely delete their data from a well-trained model, which calls for a new paradigm, namely *machine unlearning*. Existing machine unlearning methods can be generally divided into two categories: *exact* and *approximate*. Exact unlearning refers to unlearning methods that can completely remove the data influence from the model. The most representative exact unlearning method is SISA [4, 10], which divides the training data into disjoint data shards. During training, one constituent model is trained per shard. If any given data sample has to be deleted, only the constituent model associated with the shard containing this data sample has to be changed. Approximate unlearning refers to unlearning methods that try to approximate the model parameters that exact unlearning would yield without actually retraining the model. Existing approximate unlearning methods usually adopt the gradient-based update strategies to eliminate the influence of request samples on the model [85]. For example, [85] first estimates changes of the training data and then builds on closed-form updates of model parameters for unlearning the requested data changes.

However, in practice, environment interactions expose deep learning models to extra adversarial risks during the unlearning process. In fact, during the unlearning process, a motivated attacker could generate malicious unlearning requests to deteriorate the performance of deep learning models on some specific tasks. These malicious unlearning attacks pose a risk to the use of deep learning in safety- and security-critical decisions. For example, the attacker can make malicious unlearning requests to the owner of a well-trained deep learning model for the classification of traffic signs and cause the unlearned model to misclassify the Stop sign.

Despite the extensive studies for deep neural networks, there is no existing work studying the possibility and feasibility of malicious unlearning attacks against deep learning models, not to mention the effective defense mechanisms to resist such malicious unlearning attacks. Existing works that study the security vulnerabilities of deep neural networks to adversarial attacks and data poisoning attacks only focus on the testing and training stages, and fail to uncover the failure mode of deep neural networks through malicious unlearning attacks. The main challenge here is how to ensure the stealthiness of the performed malicious unlearning attacks. Studies of DNNs' robustness have enabled advances in defending against adversarial attacks and data poisoning attacks. However, existing defenses [18, 42, 44, 51, 61, 71, 73, 77, 86] are often effective only against a specific attacking type of traditional adversarial and poisoning attacks, drastically degrade the generalization performance, or are computationally prohibitive for standard machine unlearning pipelines. For example, a straightforward defense seems to use an ensemble of multiple deep learning models. However, such an ensemble method is only effective against a specific attack targeting a certain type of deep learning model. Additionally, one challenge of adopting existing robust training methods [20, 38, 39, 81, 94, 100] is the high computational cost due to the model retraining.

In order to address the above challenges, in this paper, we undertake this pioneering study on the security vulnerabilities and robustness of deep neural networks to malicious unlearning attacks, which happen during the unlearning process. Specifically, we first realize the effective malicious unlearning attacks against deep neural networks. We formulate a generic unlearning attack framework as a constrained optimization problem that maximizes the attacker's utility while constraining the malicious unlearning requests. We also extend the attack to different attacking settings (e.g., the black-box setting). Second, to effectively defend against malicious unlearning attacks, we present a general gradient influence based defense mechanism to defend malicious unlearning requests. For its realization, we iteratively find out the effective malicious unlearning requests by using their gradient influence on the unlearned models and unlearn the bad influence of these identified bad data from the unlearned models. We further conduct theoretical analyses for the proposed methods. Lastly, we empirically justify the proposed malicious unlearning attacks and the gradient influence based defense mechanism. Extensive experimental results validate that existing deep learning models lack robustness to malicious unlearning requests; we can significantly improve the robustness of deep learning models by removing the bad influence of the effective malicious unlearning requests from the unlearned models.

2 PRELIMINARY

Notations. Without loss of generability, we here consider a C-class ($C \geq 2$) classification problem, where we are given a training dataset $\mathcal{D} = \{z_i = (\boldsymbol{x}_i, y_i)\}_{i=1}^N$ with $\boldsymbol{x}_i \in \mathbb{R}^d$ as a natural example and $y_i \in [C]$ as its associated label. Let $F_{\mathcal{D}}(\boldsymbol{x}; \boldsymbol{\theta}^*)$ denote a DNN classifier with the corresponding model parameters $\boldsymbol{\theta}^* \in \Theta$, which assigns a given input $\boldsymbol{x}$ to one of the predefined classes, i.e., $F_{\mathcal{D}}(\boldsymbol{x}; \boldsymbol{\theta}^*) = c \in [C]$. Note that $F_{\mathcal{D}}(\boldsymbol{x}; \boldsymbol{\theta}^*)$ is trained over the given training dataset $\mathcal{D}$. We use $\mathcal{U}$ to denote the unlearning method, which takes the well-trained model $F_{\mathcal{D}}(\cdot; \boldsymbol{\theta}^*)$, the training dataset $\mathcal{D}$, and the unlearning data $\mathcal{D}_u$ as input, and returns an unlearned model $\mathcal{U}(F_{\mathcal{D}}, \mathcal{D}, \mathcal{D}_u)$ that is expected to be the same or similar as the retrained model $F_{\mathcal{D}\backslash\mathcal{D}_u}$. Importantly, the retrained model $F_{\mathcal{D}\backslash\mathcal{D}_u}$ is derived based on the remaining dataset (i.e., $\mathcal{D}\backslash\mathcal{D}_u$) instead of the original training dataset $\mathcal{D}$.

Machine unlearning. Note that machine unlearning aims to make models forget about some particular data. Upon a data removal request, the current model will be processed by an unlearning method to forget the corresponding information of that data inside the model. The outcome is an unlearned model, which becomes the new model for downstream prediction tasks. Next, we describe some popular machine unlearning methods.

- **SISA [3].** In SISA, the original training dataset $\mathcal{D}$ is randomly partitioned into M disjoint shards (i.e., $\{\mathcal{D}_m\}_{m=1}^M$) [10]. For the m-th shard, we can train a corresponding shard model $F_{\mathcal{D}_m}(\cdot; \boldsymbol{\theta}_m^*)$ by using $\mathcal{D}_m$, where $\boldsymbol{\theta}_m^* \in \boldsymbol{\Theta}$ are the obtained model parameters. After that, the final prediction results are obtained from the aggregation of the M submodels. Upon receiving an unlearning data, the model provider only needs to retrain the corresponding shard model.
- **The first-order based unlearning method [85].** This method uses a first-order Taylor Series of model $\boldsymbol{\theta}^*$ to derive the gradient updates. Here, we use $Z = \{z_p\}_{p=1}^P \subset \mathcal{D}$ to denote the set of targeted training data and $\tilde{Z} = \{\tilde{z}_p\}_{p=1}^P$ for the corresponding unlearned versions, where $\tilde{z}_p = (\boldsymbol{x}_p - \boldsymbol{\delta}_p, y_p)$ and $\boldsymbol{\delta}_p$ is the unlearning modification for $\boldsymbol{x}_p$. Then, this method unlearns the modifications by updating the model parameters as $\boldsymbol{\theta}^u \leftarrow \boldsymbol{\theta}^* - \tau(\sum_{\tilde{z}_p \in \tilde{Z}} \nabla \ell(\tilde{z}_p; \boldsymbol{\theta}^*) -$

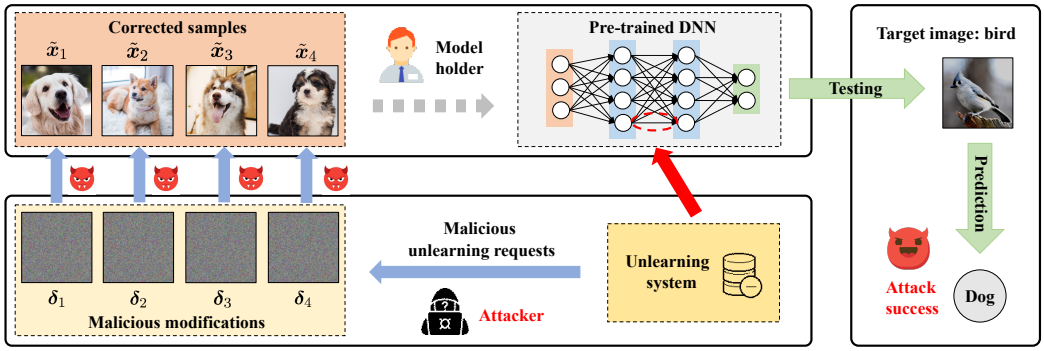


Figure 1: Illustration of malicious unlearning attacks. The attacker aims to make malicious unlearning requests to the model holder. After unlearning malicious modifications on the pre-trained DNN, the target image is successfully misclassified.

$\sum_{z_p \in Z} \nabla \ell(z_p; \theta^*)$), where θ^* denote the pre-trained model parameters, τ is a pre-defined unlearning rate, ℓ is a loss function (e.g., cross-entropy), and θ^u is the unlearned model.

- **The second-order based unlearning method [85].** This method uses the inverse Hessian matrix of the second-order partial derivatives to change the original model's parameters to obtain the unlearned model. The unlearned model can be formulated as $\theta^u \leftarrow \theta^* - H_{\theta^*}^{-1}(\sum_{\tilde{z}_p \in \tilde{Z}} \nabla \ell(\tilde{z}_p; \theta^*) - \sum_{z_p \in Z} \nabla \ell(z_p; \theta^*))$, where $H_{\theta^*}^{-1}$ is the inverse Hessian matrix, ℓ is a loss function, and θ^u is the unlearned model.
- **The unrolling SGD unlearning method [74].** It expands a sequence of stochastic gradient descent (SGD) updates with a Taylor Series to formalize a single gradient unlearning method. To reverse the effect of unlearning data provided in the SGD training steps, this unlearning strategy adds back the gradients of the unlearning data computed with respect to the initial weights to the final model weights.
- **The amnesiac unlearning method [28].** The amnesiac unlearning method views model training as a series of parameter updates to the initial model parameters. If the data owner is only concerned about the possible potential removal of a subset of data, they need only keep the parameter updates from batches containing that data.

3 MALICIOUS UNLEARNING ATTACKS

In this section, we first present the considered threat model. Then, we design a general attack framework to find out the effective unlearning attacking strategies to evaluate the robustness of deep learning models. After that, we give more discussions on the proposed malicious unlearning attacks.

3.1 Threat Model

In malicious unlearning attacks, we consider a threat model that includes a model holder and an attacker (as shown in Figure 1). The model holder owns a well-trained DNN model. The attacker pretends to be the provider of some data used by the pre-trained model and aims to make malicious unlearning requests to the model holder to delete the information of his/her requested data from the well-trained model such that the correspondingly unlearned model produces misclassifications on inputs. We assume that the attacker does not have the ability to modify the target samples during inference. Here, we study both the *white-box* and *black-box* settings. The white-box threat model [12, 27, 56, 63, 83] represents the most powerful attacker that can appear in real-world settings and is of great importance to fully study the attacker's behaviors. In this white-box setting, we assume the attacker has perfect knowledge of the system, including the model structure and the parameters of the pre-trained model, but the attacker's capability to manipulate is bounded in the L_∞ norm sense. In the black-box setting, we assume that the attacker does not have any prior knowledge about the target pre-trained model, including the model architecture and model parameters. The black-box setting produces a realistic threat model in real-world applications.

3.2 Attack Formulation

Here, we study the robustness of DNNs by designing the unlearning attack framework to explore the attacker's capability to fool DNNs.

Unlike traditional adversarial attacks and poisoning attacks, our proposed malicious unlearning attacks deceive the DNN model by making malicious unlearning requests during the unlearning process. As shown in Figure 1, the model holder owns a pre-trained model, i.e., a classification model $F_{\mathcal{D}}(\cdot; \theta^*)$ trained on dataset $\mathcal{D}$. The unlearning system represents an unlearning method $\mathcal{U}$ that can be used to unlearn the information from this classification model upon the data removal requests. The attacker's goal is to utilize the unlearning system to generate malicious unlearning requests to attack the targeted testing samples $\{\boldsymbol{x}_s\}_{s=1}^S$, forcing the targeted testing sample $\boldsymbol{x}_s$ (e.g., the bird image in Figure 1) to be designated as the attack targeted label $\bar{y}_s$ (e.g., the dog label in Figure 1). Without loss of generality, we here consider a very realistic and general setting where the attacker pretends to be a normal user and makes malicious data modification requests on the targeted training samples. In practice, the attacker can make a reasonable request for malicious data modifications by using the excuse of bad data quality issues (e.g., noises) or some privacy requirements, so that the unlearned model will be fooled into misclassifying the targeted testing samples during inference, thereby reaching the attacker's goal.

Let $\mathcal{D}_t = \{(\boldsymbol{x}_p, y_p)\}_{p=1}^P$ denote the set of targeted training samples. The attacker wants to make the corresponding malicious unlearning modification (i.e., $\boldsymbol{\delta}_p$) on each $\boldsymbol{x}_p$ and replace the sample

$(\boldsymbol{x}_p, y_p)$ with the unlearned version $(\tilde{\boldsymbol{x}}_p = \boldsymbol{x}_p - \boldsymbol{\delta}_p, y_p)$ to update the pre-trained model and derive an unlearned model. Note that the attacker's objective is to derive effective unlearning requests (i.e., the set of unlearning modifications $\boldsymbol{\Phi} = \{\boldsymbol{\delta}_p\}_{p=1}^P$) to maliciously update the pre-trained model and successfully misclassify the targeted testing samples. In order to achieve this, based on the pre-trained model and the targeted testing samples $\{\boldsymbol{x}_s\}_{s=1}^S$, the attacker can generate the malicious unlearning requests as follows

$$\begin{aligned} &\max_{\{\boldsymbol{\delta}_p\}_{p=1}^P} \sum_{s=1}^{S} \mathbb{I}[F_{\mathcal{D}\setminus\boldsymbol{\Phi}}(\boldsymbol{x}_s; \boldsymbol{\theta}^u) = \bar{y}_s)] \\ &s.t., F_{\mathcal{D}\setminus\boldsymbol{\Phi}}(\cdot; \boldsymbol{\theta}^u) = \mathcal{U}(F_{\mathcal{D}}(\cdot; \boldsymbol{\theta}^*), \mathcal{D}, \{\boldsymbol{\delta}_p\}_{p=1}^P) \\ &\forall p \in [P], ||\boldsymbol{\delta}_p||_\infty \leq \epsilon, \end{aligned} \tag{1}$$

where $F_{\mathcal{D}\setminus\boldsymbol{\Phi}}(\cdot; \boldsymbol{\theta}^u)$ is the unlearned model, and ϵ is the maximal magnitude of the requested data modifications. In the above, the unlearning method $\mathcal{U}$ unlearns the modifications $\boldsymbol{\Phi}$ and produces an unlearned model $\boldsymbol{\theta}^u$, after which the output of the target sample $\boldsymbol{x}_s$ is incorrectly identified as the attack targeted label $\bar{y}_s$. By solving the above optimization, the attacker can generate malicious unlearning requests to maximize his/her attack goal, i.e., maximizing the number of successful targeted testing samples attacks.

Note that the above attack framework can be easily generalized to a scenario where the targeted training samples are completely removed from the model during the unlearning process [85], which means that the above malicious unlearning attacks can be easily transformed to the whole data removal case. In such a case, the motivated attacker wholly removes a set of targeted training samples instead of partially unlearning some data information (e.g., noises). The above security vulnerability analysis will help us understand how the attacker can generate malicious unlearning requests to mislead the trained deep learning models to output incorrect predictions. Below, we discuss malicious unlearning attacks in the second-order based unlearning setting.

3.3 Discussion

The black-box setting. For the malicious unlearning attacks proposed above, we also consider the black-box setting. In such a setting, the attacker can randomly select one or several deep neural network architectures to substitute the pre-trained model and then transfer the generated malicious unlearning data leveraging the transferability property of neural networks [29, 37, 48, 84, 87, 96]. For example, we can easily train a random set of deep learning models to substitute the pre-trained model in unlearning methods. We are then able to generate malicious unlearning modifications and transfer them to the target black-box model.

Malicious unlearning attacks in the second-order based unlearning case. Note that we can perform malicious unlearning attacking by using existing machine unlearning methods. Due to space limitations, we here take the second-order unlearning method [85] as an example to show how to generate malicious unlearning requests by following the above proposed general attack framework in Eqn. (1). This unlearning strategy uses the inverse Hessian matrix of the second-order partial derivatives to change the original model's parameters to obtain the unlearned model. Specifically, the second-order change $\Delta(Z, \tilde{Z})$ is derived by computing the gradient difference between Z and $\tilde{Z}$, i.e., $\Delta(Z, \tilde{Z}) = \boldsymbol{H}_{\boldsymbol{\theta}^*}^{-1}(\sum_{\tilde{z}_p \in \tilde{Z}} \nabla \ell(\tilde{z}_p; \boldsymbol{\theta}^*) - \sum_{z_p \in Z} \nabla \ell(z_p; \boldsymbol{\theta}^*))$. To achieve the attack goal, the attacker here can manipulate the original model as

$$\begin{aligned} \min_{\{\boldsymbol{\delta}_p\}_{p=1}^P} \sum_{s=1}^{S} &\max(\max_{c \neq \bar{y}_s} \boldsymbol{F}^c_{\mathcal{D}\setminus\boldsymbol{\Phi}}(\boldsymbol{x}_s; \boldsymbol{\theta}^u) - \boldsymbol{F}^{\bar{y}_s}_{\mathcal{D}\setminus\boldsymbol{\Phi}}(\boldsymbol{x}_s; \boldsymbol{\theta}^u), -\beta) \\ s.t., \quad &\boldsymbol{\theta}^u \leftarrow \boldsymbol{\theta}^* - \boldsymbol{H}_{\boldsymbol{\theta}^*}^{-1}(\sum_{\tilde{z}_p \in \tilde{Z}} \nabla \ell(\tilde{z}_p; \boldsymbol{\theta}^*) \\ &- \sum_{z_p \in Z} \nabla \ell(z_p; \boldsymbol{\theta}^*)) \\ &\forall p \in [P], ||\boldsymbol{\delta}_p||_\infty \leq \epsilon, \end{aligned} \tag{2}$$

where $\boldsymbol{\theta}^*$ denote the model parameters, ℓ is a loss function (e.g., cross-entropy loss), $\boldsymbol{\Phi} = \{\boldsymbol{\delta}_p\}_{p=1}^P$, and $\boldsymbol{F}_{\mathcal{D}\setminus\boldsymbol{\Phi}}(\cdot; \boldsymbol{\theta}^u)$ is the logit output of the unlearned model $\boldsymbol{\theta}^u$. Note that the above adversarial loss aims to misclassify the targeted testing sample $\boldsymbol{x}_s$ to the attack targeted label $\bar{y}_s$. The first constraint directly updates the pre-trained model $\boldsymbol{\theta}^*$ by the inverse Hessian matrix $\boldsymbol{H}_{\boldsymbol{\theta}^*}^{-1}$ with the gradient difference between Z and $\tilde{Z}$. The second constraint controls the maximum manipulation of malicious unlearning modifications.

Theorem 3.1. *Let $\boldsymbol{\theta}^*$ and $\boldsymbol{\theta}^u$ denote the original and the unlearned model, respectively. We use $\tilde{\mathcal{D}}$ to denote the dataset containing the malicious unlearning modifications $\tilde{Z}$ required for the unlearning task. Assume that $||\boldsymbol{x}_i|| \leq 1$ for all samples, the gradient $\nabla \ell(\tilde{z}_p; \boldsymbol{\theta}^*)$ is ξ_1-Lipschitz with respect to $\boldsymbol{z}$ at θ^*, and $\nabla^2 \ell(\tilde{z}_p; \boldsymbol{\theta}^*)$ is ξ_2-Lipschitz with respect to $\boldsymbol{\theta}$. For the proposed malicious unlearning attacks in the second-order update case, we can derive the following*

$$||\nabla \ell(\tilde{\mathcal{D}}; \boldsymbol{\theta}^u) - \nabla \ell(\mathcal{D}; \boldsymbol{\theta}^*)||_2 \leq \frac{\xi_1^2 \xi_2 d^2 \epsilon^2 P^2}{||\boldsymbol{H}_{\boldsymbol{\theta}^*}^{-1}||_2^{-2}},$$

where $\boldsymbol{H}_{\boldsymbol{\theta}^*}^{-1}$ is the inverse Hessian matrix for the original model $\boldsymbol{\theta}^*$, ϵ is the maximal magnitude of the requested data modifications, and d denotes the feature dimension.

From the above theorem, we can see that the larger the magnitude of the requested malicious unlearning modifications, the less robust the deep learning model is to malicious unlearning attacks. To solve the proposed optimization problem in Eqn. (2), we use Hessian Vector Product to approximate the inverse Hessian to reduce the computational cost, which only requires calculating $\boldsymbol{Hv}$ instead of storing $\boldsymbol{H}^{-1}$ for computing the expressions of $\boldsymbol{H}^{-1}\boldsymbol{v}$, where $\boldsymbol{v}$ is a vector and $\boldsymbol{H}$ is the Hessian matrix. Note that it is computationally infeasible to compute the exact Hessian matrix and its inverse for models with a very large number of model parameters [50, 52, 65]. In Algorithm 1, we provide the procedure to solve the optimization problem of unlearning attacks in the second-order case. For simplicity, we use $\psi(\boldsymbol{x}_s, \bar{y}_s; \boldsymbol{\theta}^u)$ to denote the adversarial loss (defined in Eqn. (2)) for sample $\boldsymbol{x}_s$.

4 A GENERAL GRADIENT INFLUENCE BASED DEFENSIVE MECHANISM

In the above, we propose malicious unlearning attacking strategies to demonstrate the vulnerabilities and weaknesses of deep learning models during the unlearning process. This lack of robustness is

Algorithm 1: Malicious unlearning attacks in the second-order based data removal case

Input: Pre-trained model $\boldsymbol{\theta}^*$, training dataset $\mathcal{D}$, target data $\{\boldsymbol{x}_s\}_{s=1}^S$, attack targeted label $\bar{y}_s$, targeted training data points $\boldsymbol{Z} = \{\boldsymbol{z}_p\}_{p=1}^P$, modification rate γ, modification bound ϵ, optimization steps T

Output: Malicious modifications $\boldsymbol{\Phi} = \{\boldsymbol{\delta}_p\}_{p=1}^P$

Randomly initialize unlearning modifications $\{\boldsymbol{\delta}_p\}_{p=1}^P$

for $t = 1$ **to** T **do**

  Compute corrected data $\tilde{\boldsymbol{Z}} \leftarrow \{(\boldsymbol{x}_p - \boldsymbol{\delta}_p\, y_p)\}_{p=1}^P$

  Update the unlearned model $\boldsymbol{\theta}^u$ with $\Delta(\boldsymbol{Z}, \tilde{\boldsymbol{Z}})$ and the inverse Hessian matrix in Eqn. (2)

  Compute the adversarial loss $\Psi \leftarrow \sum_{s=1}^S \psi(\boldsymbol{x}_s, \bar{y}_s; \boldsymbol{\theta}^u)$

  Update $\{\boldsymbol{\delta}_p\}_{p=1}^P \leftarrow \{\boldsymbol{\delta}_p\}_{p=1}^P - \gamma \nabla_{\{\boldsymbol{\delta}_p\}_{p=1}^P} \Psi$

  Project $\{\boldsymbol{\delta}_p\}_{p=1}^P$ onto ϵ bound

problematic in real-world applications where maliciously manipulated predictions could impair safety and trustworthiness. However, existing unlearning methods fail to provide robustness guarantees for the unlearning system. As aforementioned, existing defenses [18, 42, 44, 51, 61, 71, 73, 77, 86] are often effective only against a specific attacking type of traditional attacks, or are computationally prohibitive for standard machine unlearning pipelines. For example, existing robust training methods [32, 34, 39, 47, 82, 94] are limited by the high computation complexity due to model retraining.

To address the above challenges, we here develop a general gradient influence based defensive method to improve the robustness of deep learning models against malicious unlearning attacks. Note that for a targeted attack to be successful, the target $\boldsymbol{x}_s$ needs to be misclassified as the adversarial class $\bar{y}_s$. To this end, the corrected samples need to pull the representation of the target sample toward the adversarial class. This means that after fulfilling the unlearning requests, the corresponding corrected samples need to mimic the gradient of the adversarially labeled target, i.e., $\nabla\ell((\boldsymbol{x}_s, \bar{y}_s); \boldsymbol{\theta}^u) \approx \frac{1}{|\mathcal{D}_t|}\sum_{p\in P}\nabla\ell((\tilde{\boldsymbol{x}}_p = \boldsymbol{x}_p - \boldsymbol{\delta}_p, y_p); \boldsymbol{\theta}^u)$, where ℓ is the training loss (e.g., cross-entropy) for training classifier F, $\mathcal{D}_t = \{(\boldsymbol{x}_p, y_p)\}_{p=1}^P$ is the set of targeted training samples and $\boldsymbol{\theta}^u$ denotes the unlearned model. Therefore, instead of directly rejecting unlearning requests, we drop the corrected malicious samples that have a different gradient compared to other instances in their class. In order to find the corrected malicious samples, we can find the medoids of each class in the gradient space [1, 25, 31, 49, 92]. Note that for class $c \in [C]$, its corresponding medoids are the most centrally located samples of a dataset, which minimize the sum of dissimilarity between every sample to its nearest medoid. Let υ_c denote the number of medoids for class c. We use $\boldsymbol{Q}^c$ to denote the set of υ_c-medoids to be optimized. Let $\tilde{\mathcal{D}}^c = \{z_i = (\boldsymbol{x}_i, y_i)\}_{i=1}^{|\tilde{\mathcal{D}}^c|}$ denote the set of new training samples having the class label of c. Note that these new training samples are derived by implementing the requested unlearning modifications. We introduce a binary variable ζ_i which are 1 if sample $\boldsymbol{x}_i \in \tilde{\mathcal{D}}^c$ is a medoid, 0 otherwise; and the variables Ξ_{ij} which takes 1 if sample $\boldsymbol{x}_j$ is assigned to medoid $\boldsymbol{x}_i$, i.e., $\boldsymbol{x}_i$ is the most similar medoid to data sample $\boldsymbol{x}_j$. We also introduce a set of sample indexes $\chi^-(j) = \{j \in [|\tilde{\mathcal{D}}^c|] | \boldsymbol{z}_i, \boldsymbol{z}_j \in \tilde{\mathcal{D}}^c, i \neq j\}$. The set of υ_c-medoids can be obtained by solving the following formulated optimization problem

$$\begin{aligned}\min_{\mathcal{Q}^c \subset \tilde{\mathcal{D}}^c} \sum_{\boldsymbol{z}_i, \boldsymbol{z}_j \in \tilde{\mathcal{D}}^c} D(\nabla\ell((\boldsymbol{x}_i, y_i); \boldsymbol{\theta}^u), \nabla\ell((\boldsymbol{x}_j, y_j); \boldsymbol{\theta}^u)) * \Xi_{ij} \\ s.t., \sum_{i\in\chi^-(j)} \Xi_{ij} + \zeta_j = 1, \\ \Xi_{ij} \leq \zeta_i, \\ \sum_{i=1}^{|\tilde{\mathcal{D}}^c|} \zeta_i = \upsilon_c,\end{aligned} \tag{3}$$

where $\zeta_i \in \{0, 1\}$, $\Xi_{ij} \in \{0, 1\}$, and D calculates the Euclidean distance between $\nabla\ell((\boldsymbol{x}_i, y_i); \boldsymbol{\theta}^u)$ and $\nabla\ell((\boldsymbol{x}_j, y_j); \boldsymbol{\theta}^u)$. The objective stated above aims to minimize the dissimilarities between data samples and their closest medoids. The first constraint ensures that a sample is either a medoid itself or assigned to a medoid. The second constraint enforces that each sample is assigned to exactly one medoid. The third constraint in the above imposes that the number of medoids must be equal to υ_c.

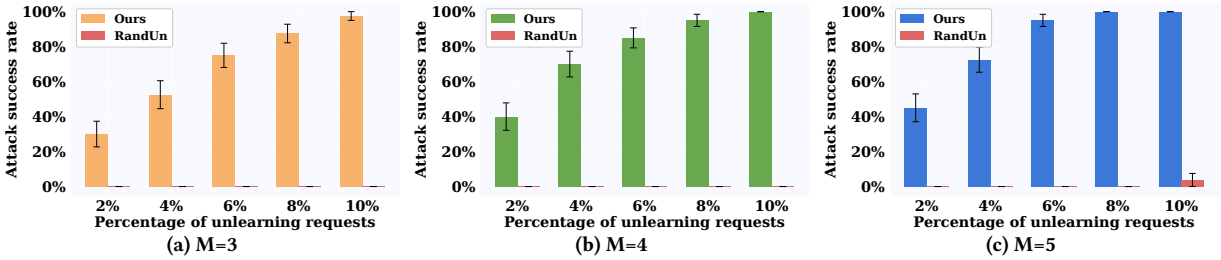
After finding the medoids for each class, we can identify the potential effective unlearned samples and then use existing unlearning techniques to unlearn the isolated medoids to effectively defend malicious unlearning attacks. With such robustness guarantees of the unlearning system, we do not need to worry about an attacker with clever algorithms for choosing malicious unlearning requests. However, directly solving the above optimization is NP-hard [5, 45, 78]. In order to optimize it, we employ a randomized algorithm inspired by multi-arm bandits [9, 70]. This technique helps reduce the time complexity while ensuring the same results with high probability. To solve the optimization problem described above, we begin by iteratively selecting samples that minimize the given loss function to obtain an initial set of υ_c medoids in a greedy manner. The first sample added in this manner is the medoid of all $\tilde{\mathcal{D}}^c$ samples. For a given set of q medoids $\boldsymbol{Q}_q^c = \{z_1, \cdots, z_q\}$, the next sample to be added is determined by solving the following loss

$$\begin{aligned}\min_{z\in\tilde{\mathcal{D}}^c\setminus\mathcal{Q}_q^c} \sum_{j=1}^{|\tilde{\mathcal{D}}^c|} \min((D(\nabla\ell((\boldsymbol{x}, y); \boldsymbol{\theta}^u), \nabla\ell((\boldsymbol{x}_j, y_j); \boldsymbol{\theta}^u)) \\ - \min_{\boldsymbol{z}'\in\mathcal{Q}_q^c} D(\nabla\ell((x', y'); \boldsymbol{\theta}^u), \nabla\ell((\boldsymbol{x}_j, y_j); \boldsymbol{\theta}^u))), 0),\end{aligned} \tag{4}$$

where $\boldsymbol{z} = (\boldsymbol{x}, y)$ and $\boldsymbol{z}' = (\boldsymbol{x}', y')$. Then, we identify the medoid-nonmedoid pair that yields the greatest reduction in loss among all possible $\upsilon_c(|\tilde{\mathcal{D}}^c| - \upsilon_c)$ pairs. Let $\boldsymbol{Q}_{\upsilon_c}^c$ represent the current set of υ_c medoids. To determine the best medoid-nonmedoid pair for swapping, we solve the following optimization problem

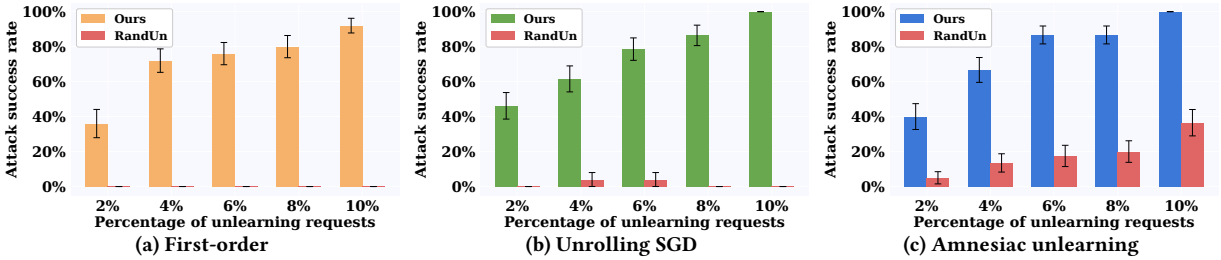
$$\begin{aligned}\min_{(\boldsymbol{z}^1, \boldsymbol{z}^2)\in\mathcal{Q}_{\upsilon_c}^c\times(\tilde{\mathcal{D}}^c\setminus\mathcal{Q}_{\upsilon_c}^c)} \sum_{j=1}^{|\tilde{\mathcal{D}}^c|} \min((D(\nabla\ell((\boldsymbol{x}^2, y^2); \boldsymbol{\theta}^u), \nabla\ell((\boldsymbol{x}_j, \\ y_j); \boldsymbol{\theta}^u)) - \min_{\boldsymbol{z}'\in\mathcal{Q}_{\upsilon_c}^c\setminus\{\boldsymbol{z}^1\}} D(\nabla\ell((\boldsymbol{x}', y'); \boldsymbol{\theta}^u), \nabla\ell((\boldsymbol{x}_j, y_j); \boldsymbol{\theta}^u))), 0),\end{aligned} \tag{5}$$

where υ_c denotes the number of medoids for class c. To optimize the search for effective medoid-nonmedoid pairs, we continue swapping until no further improvements can be achieved. The optimization problems described in Eqn. (4) and (5) can be formulated as



(a) M=3 (b) M=4 (c) M=5

Figure 2: Attack success rate of malicious unlearning attacks via the exact removal method (SISA).



(a) First-order (b) Unrolling SGD (c) Amnesiac unlearning

Figure 3: Attack success rate of malicious unlearning attacks via the approximate removal methods.

a best-arm identification problem, drawing inspiration from the multi-armed bandits literature [2, 9, 70, 76]. In a typical best-arm identification problem, we have a set of arms, and the objective is to identify the arm with the highest expected reward while minimizing the total number of arm pulls. Specifically, in Eqn. (4), each potential medoid is treated as an arm in the best-arm identification problem. The arm parameter corresponds to the associated distance value, and pulling an arm corresponds to calculating the loss for a randomly selected sample. Similarly, in Eqn. (5), each medoid-nonmedoid pair corresponds to an arm. Note that after finding out the potential effective samples based on the above proposed method, we then unlearn these identified samples by using existing machine unlearning techniques [28, 74, 85].

5 EXPERIMENTS

We conduct experiments on real-world datasets to evaluate the performance of the proposed mechanisms. The experimental setup is first described in Section 5.1. Then we show the experimental results for our proposed malicious unlearning attacks and the black-box setting in Section 5.2 and Section 5.3, respectively. Next, in Section 5.4, we evaluate the defense performance. Lastly, we present the experimental results for the ablation study in Section 5.5.

5.1 Experimental Setup

Datasets and network architectures. In experiments, we evaluate our methods on the following datasets: CIFAR-10 [41], Tiny ImageNet [19], and Dogfish [40]. The CIFAR-10 dataset contains 50,000 training images and 10,000 test images for 10 classes. Each image has a resolution of $3 \times 32 \times 32$. The Tiny ImageNet dataset contains 100,000 training images, 10,000 validation images, and 10,000 test images for a total of 200 classes. Each image has dimensions of $3 \times 64 \times 64$. The Dogfish dataset contains 1,800 training images and 600 test images. Each image is represented by a 2,048-dimensional vector. In experiments, we use various neural network architectures, including ResNet-18 [30], VGG-16 [68], MobileNetV2 [66], a 6-layer ConvNet with batch normalization [24, 36], and a 2-layer fully connected neural network.

Parameter settings. In experiments, we adopt the following popular unlearning methods: SISA [3], the first-order based unlearning method [85], the unrolling SGD unlearning method [74], the amnesiac unlearning method [28], and the second-order based unlearning method [85]. For SISA, we train the submodel for 200 epochs with a learning rate of 0.01 and a batch size of 125 in each data shard. Then, we perform 60 optimization steps with an initial modification rate of 200 (decayed by 10× every 20 steps) in the selected data shards for unlearning attacks. During the defense stage, we train the model for the same 200 epochs and use a subset of 40% of medoids. For other adopted unlearning methods (i.e., first-order, unrolling SGD, amnesiac unlearning, and second-order), we pre-train the models for 20 epochs with a learning rate of 0.01 and a batch size of 128. Then, we perform 180 optimization steps with an initial modification rate of 200 (decayed by 10× every 60 steps) for unlearning attacks. In the first-order method, we set the unlearning rate to 0.00002. In the unrolling SGD method, we use a learning rate of 0.00015 and perform a fine-tuning epoch of 1 (representing the number of copies of the gradient performed in the SGD steps). In the amnesiac unlearning method, we initialize a learning rate of 0.0001 and perform a fine-tuning epoch of 1 to regain performance. For each adopted approximate unlearning method in the defense stage, we use a subset of 40% of medoids. In all the aforementioned methods, we choose a small modification bound of 8/255 for the malicious modifications, unless otherwise specified.

Baseline. Since there is no existing work studying the vulnerabilities of DNNs to malicious unlearning attacks, in experiments, we adopt the *RandUn* baseline, where we craft the unlearning modifications by using uniform random noises.

5.2 Malicious Unlearning Attacks

We start with evaluating the performance of the proposed malicious unlearning attacks via various unlearning methods, including

the exact unlearning method (SISA) and the approximate unlearning methods (first-order, unrolling SGD, and amnesiac unlearning). For each method, we compare our proposed malicious unlearning attacks with the baseline in terms of attack success rate which is defined as the number of successful attacks achieved among all attack attempts. We use 5 sets of targeted testing samples corresponding to the first 5 image IDs from the test set and aggregate their results. Different proportions of unlearning requests (i.e., the targeted training samples to generate malicious modifications) are involved in the evaluation and are randomly selected from the training set.

First, we conduct experiments to investigate the performance of unlearning attacks in the exact setting (via SISA). In Figure 2, we adopt the CIFAR-10 dataset with 40% training size and divide it into 5 disjoint data shards. Each data shard is trained with the ConvNet model. We then randomly select 3 out of 5 shards (M=3), 4 out of 5 shards (M=4), and all shards (M=5) to attack, respectively. Here, we focus on the target class of the bird and the attack targeted label of the dog. As shown in the figure, our proposed unlearning attacks achieve significant attack success rates compared to the RandUn baseline on different numbers of attacked shards. Random noises work poorly on attacking models during the unlearning process, reflecting the challenge of conducting targeted unlearning attacks in the exact setting. In the majority vote aggregation setting, our mission is to successfully attack more shards to obtain more targeted misclassifications. In our approach, we can find that even attacking 3 shards, which is on the margin of voting, we can still remarkably achieve an attack success rate of 75% when unlearning modifications on targeted training samples of 6% and as high as 97.5% attack success rate when unlearning modifications on targeted training samples of 10%. When attacking all 5 shards, we can easily hit an attack success rate of 72% with 4% unlearning requests and an attack success rate of 100% with 8% unlearning requests. All in all, the results show that our unlearning attacks have an efficient attack performance of unlearning malicious modifications in the exact setting with the SISA method.

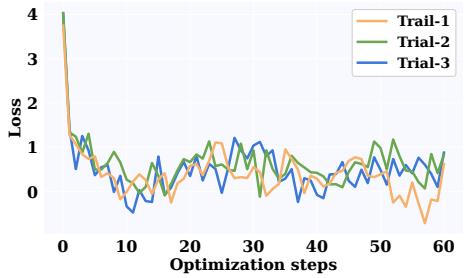


Figure 4: Convergence of the optimization for malicious unlearning attacks in the exact setting.

Next, we show the derived experimental results of unlearning attacks in the approximate settings (via the first-order unlearning method, the unrolling SGD unlearning method, and the amnesiac unlearning method). In Figure 3, we pre-train the ResNet-18 model on the CIFAR-10 dataset and then target different percentages of unlearning requests, from 2% to 10%, to attack the target images of birds to be predicted as dogs. As shown in the figure, the RandUn baseline, which utilizes random noises as unlearning modifications, has almost no effect on attacks using the first-order unlearning method or the unrolling SGD unlearning method. It has some effect on attacks using the amnesiac unlearning method, but the success rate is still very low, with only a 36.5% attack success rate for 10% unlearning requests. In contrast, our proposed unlearning attacks demonstrate high confidence in achieving successful attacks through various unlearning methods. For example, when unlearning malicious modifications on targeted training samples of 6%, our approach hits about 76% attack success rate via the first-order unlearning method, 79% attack success rate via the unrolling SGD unlearning method, and 87% attack success rate via the amnesiac unlearning method. From these derived experimental results, we can find that our optimization framework is applicable and effective for malicious unlearning attacks in approximate settings with the first-order unlearning method, the unrolling SGD unlearning method, and the amnesiac unlearning method.

Then, we evaluate the convergence of the optimization process in our proposed malicious unlearning attacks to show how it benefits the attack success rate in the experiments. In Figure 4, we report the evolution of the objective value of a particular data shard in the exact setting (via SISA) with respect to the number of optimization steps. We perform the experiment three times, each time randomly selecting the targeted training samples from the same data shard. From this figure, we can observe that the adversarial loss, which is the objective to be minimized, rapidly decreases up to step 15 and converges around 0. This adversarial loss has the property that when it is less than 0, the targeted testing sample is successfully misclassified as the attack targeted label. Therefore, this objective can contribute to the attack success rate in the optimization stage.

Further, we extend our proposed malicious unlearning attacks to the untargeted setting, where the attacker aims to mislead the model to predict any of the wrong labels for targeted testing examples. We here take the malicious unlearning attacks via the first-order unlearning method as an example, and report the corresponding experimental results in Figure 5. Here, we perform the untargeted unlearning attacks on the dog class in the CIFAR-10 dataset and compare the attack success rates with the RandUn baseline. As shown in the figure, our proposed malicious unlearning attacks also achieve impressive performance in the untargeted setting.

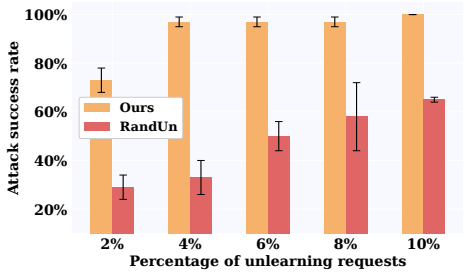


Figure 5: Attack success rate of malicious unlearning attacks in the untargeted setting.

5.3 Black-box Setting

In this section, we conduct experiments to explore the malicious unlearning attacks in the black-box setting. In Figure 6, we apply three different network architectures (i.e., ResNet-18, VGG-16, and MobileNetV2) on the CIFAR-10 dataset with targeted training samples of 8%. The horizontal line represents the pre-trained black-box

Table 1: Attack success rate and test accuracy of the proposed defense mechanism against malicious unlearning attacks.

Removal method	Percentage of unlearning requests	Undefended		Defended	
		Attack success rate ↑	Test accuracy ↑	Attack success rate ↓	Test accuracy ↑
SISA	4%	52.50% ± 8.00%	62.89% ± 0.29%	**0.00% ± 0.00%**	74.17% ± 0.27%
	6%	75.00% ± 6.93%	62.37% ± 0.28%	**5.00% ± 3.49%**	73.20% ± 0.34%
	8%	87.50% ± 5.30%	62.37% ± 0.32%	**5.00% ± 3.49%**	73.19% ± 0.22%
	10%	97.50% ± 2.50%	61.19% ± 0.32%	**2.50% ± 2.50%**	71.09% ± 0.25%
First-order	4%	65.00% ± 7.64%	88.60% ± 0.20%	**0.00% ± 0.00%**	82.73% ± 0.83%
	6%	77.50% ± 6.69%	88.32% ± 0.15%	**0.00% ± 0.00%**	82.35% ± 0.82%
	8%	97.50% ± 2.50%	77.16% ± 0.65%	**0.00% ± 0.00%**	81.52% ± 0.84%
	10%	100.00% ± 0.00%	79.09% ± 0.55%	**0.00% ± 0.00%**	80.50% ± 0.67%
Unrolling SGD	4%	62.50% ± 7.75%	88.69% ± 0.20%	**0.00% ± 0.00%**	83.25% ± 0.70%
	6%	85.00% ± 5.72%	88.67% ± 0.20%	**2.50% ± 2.50%**	82.71% ± 0.76%
	8%	97.50% ± 2.50%	76.87% ± 0.49%	**2.50% ± 2.50%**	82.68% ± 0.81%
	10%	100.00% ± 0.00%	74.91% ± 0.60%	**0.00% ± 0.00%**	82.62% ± 0.79%

network, and the vertical line represents the surrogate unlearning network used to attack the black-box network. As shown in the figure, the unlearning attacks demonstrate the ability to transfer the generated malicious modifications to attack the black-box network, even though the black-box network is trained with a different network architecture than the surrogate network. For example, the surrogate network of ResNet-18 achieves an attack success rate of 89% to attack the black-box network of MobileNetV2. Note that the malicious unlearning modifications generated on MobileNetV2 do not work as well on other networks. One explanation is that MobileNetV2 is not trained as well as others in our experimental settings, which may make the generated malicious modifications less confident to attack a relatively robust network.

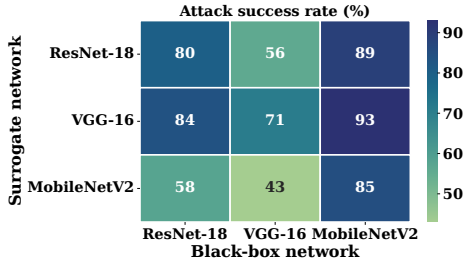


Figure 6: Attack success rate of malicious unlearning attacks in the black-box setting.

5.4 Gradient Influence based Defense

In this section, we evaluate the effectiveness of our proposed defense method against malicious unlearning attacks during the unlearning process. We employ the same setup as in the experiments on malicious unlearning attacks, and we compare the attack success rate and test accuracy before and after the defense.

Table 1 shows the results of defending against malicious unlearning attacks in the exact setting (via SISA) and the approximate settings (via the first-order unlearning method and the unrolling SGD unlearning method). In SISA, we adopt the same partition with the malicious unlearning modifications generated from unlearning attacks and defend against 3 attacked shards among the partitioned 5 disjoint data shards on the CIFAR-10 dataset. As shown in the table, the undefended unlearned model achieves high attack success rates for various percentages of unlearning requests. However, when applying our proposed defense method to each attacked shard using different percentages of unlearning requests, the attack success rate of the defended unlearned model is significantly reduced below 5%. Especially with 4% unlearning requests, the attacked unlearned model can be completely defended (0% attack success rate). In first-order and unrolling SGD unlearning methods, we defend against the malicious unlearning requests generated on the Dogfish dataset with a 2-layer fully connected neural network. Here, we set the unlearning rate to 0.02 for first-order and the learning rate to 0.02 for unrolling SGD, and we use a subset of 20% of medoids. As we can see, unlearning attacks using both approximate methods before the defense can achieve remarkable attack success rates for various percentages of unlearning requests. However, when our proposed defense method is adopted, the attack success rates decrease significantly. Specifically, the defended unlearned model achieves attack success rates of 0% via the first-order unlearning method and attack success rates below 2.5% via the unrolling SGD unlearning method for different percentages of unlearning requests. In addition, our proposed defense method can retain the test accuracy after defense and even showcases minor improvements. Based on these reported comparative results, it is evident that our proposed defense method successfully decreases the attack success rates of malicious unlearning attacks in both the exact setting and the approximate settings by a large margin, thereby enhancing the robustness of the unlearning system.

5.5 Ablation Study

Here, we conduct an ablation study to analyze the impact of the modification bound on the proposed malicious unlearning attacks. In experiments, we compare the modification bound from $\epsilon = 4/255$ to $\epsilon = 32/255$ for unlearning attacks via the first-order based unlearning method, the second-order based unlearning method, and the unrolling SGD unlearning method. We pre-train ResNet-18 on the CIFAR-10 dataset and unlearn the malicious modifications on

targeted training samples of 4%. As Figure 7 shows, for each unlearning method, the attack success rate increases as the maximal magnitude of requested unlearning modifications increases. The results are consistent with the theoretical analysis results (See Theorem 3.1) and the observation that deep learning models become less robust to malicious unlearning attacks when larger magnitudes of unlearning modifications are conducted.

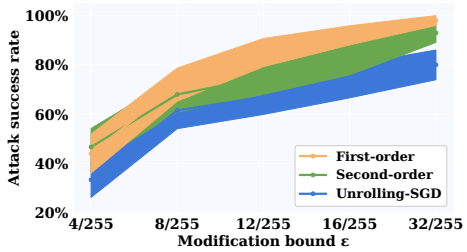


Figure 7: Impact of the modification bound ϵ on malicious unlearning attacks.

In addition, we test with our proposed malicious unlearning attacks and the gradient influence based defense method on the Tiny ImageNet dataset. We select the first 50 classes of the Tiny ImageNet dataset and pre-train with the VGG-16 model. In experiments, we randomly sample the target class, the adversarial class, and the target image. We then perform malicious unlearning attacks via the first-order unlearning method, incorporating targeted training samples of 4%, 6%, and 8%, with a modification bound of 16/255. Table 2 shows the derived experimental results of the undefended and defended models against malicious unlearning attacks. As shown in the table, our proposed malicious unlearning attacks can achieve an attack success rate of 62.5% with 8% unlearning requests. However, after applying the defenses, the attack success rate of the unlearned model drops below 5% in all cases presented in the table and reaches 0% with 4% unlearning requests.

Table 2: Malicious unlearning attacks and the defending on the Tiny ImageNet dataset.

Percentage of unlearning requests	Attack success rate	
	Undefended ↑	Defended ↓
4%	53.85% ± 9.97%	**0.00% ± 0.00%**
6%	56.25% ± 8.91%	**5.00% ± 3.49%**
8%	62.50% ± 8.70%	**2.50% ± 2.50%**

6 RELATED WORK

Currently, there are two broad and important areas of security attack: adversarial attacks [12, 15, 35, 69, 72, 88, 98, 99] and data poisoning attacks [11, 21, 22, 43, 53, 54, 93]. In adversarial attacks that happen at the test stage, the attacker aims to add deliberately designed tiny perturbations to benign test examples such that the perturbed samples are misclassified by a model with high confidence. In data poisoning attacks that happen at the training stage, the attacker tries to manipulate the training data in order to corrupt the trained model. The attack model for the two security attack classes can be generally specified as either black-box, or white-box. In a black-box threat model, the attacker has no access to the trained model [15]. White-box attacks refer to the case when the attacker has complete knowledge about a target model, which can facilitate the tasks of crafting adversarial examples and poisoning training samples [21]. However, all of these mentioned works fail to address the security vulnerabilities of deep learning models during the unlearning process. Different from traditional adversarial attacks and data poisoning attacks, the proposed malicious unlearning attacks directly manipulate the pre-trained models during the unlearning process and aim to generate malicious unlearning requests to fool the unlearned models into making wrong predictions.

The paradigm of machine unlearning [3, 7, 8, 13, 62, 80] has attracted much attention recently. It has emerged from "the right to be forgotten" [26, 59, 60], where individuals should be entitled to the right to have their data removed from public directories. A line of works focus on post-processing the trained model [28, 57, 74, 85, 95] so that the results of the unlearned model are statistically (almost) indistinguishable from those of the retrained model. Another one is to find new training algorithms to reduce the retraining cost. For example, [3] proposes to split the entire training dataset into several shards and train a separate sub-model for each shard. The unlearning process can be achieved simply by only retraining these involved shard sub-models (that contain the requested unlearning samples) to reduce the overhead of computational resources and memory storage. However, all existing works on machine unlearning fail to study the vulnerabilities and robustness of deep learning models to malicious unlearning attacks, which generate malicious unlearning requests during the unlearning process.

7 CONCLUSION

In this paper, for the first time, we systematically study the security vulnerabilities and robustness of deep learning to malicious unlearning attacks, where the attacker wants to generate malicious unlearning requests during the unlearning process. Specifically, we first propose a novel generic unlearning attacking framework, which reveals that current deep learning models are vulnerable to malicious unlearning attacks. We also explore various unlearning attacking settings. In addition, to counteract these unlearning risks, we also present a general gradient influence based defense mechanism. We also conduct theoretical analyses of the proposed methods. The extensive experimental results on real-world datasets not only show that existing deep learning models are vulnerable to malicious unlearning attacks, but also demonstrate that the defense mechanism can substantially enhance the robustness of deep learning models to malicious unlearning attacks. We believe that our work makes people aware of potential risks when they apply machine unlearning methods to critical applications.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and helpful suggestions. This work is supported in part by the US National Science Foundation under grant CCF-2220401. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Hojjat Aghakhani, Dongyu Meng, Yu-Xiang Wang, Christopher Kruegel, and Giovanni Vigna. 2021. Bullseye polytope: A scalable clean-label poisoning attack with improved transferability. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 159–178.

[2] Tavor Z Baharav, Gary Cheng, Mert Pilanci, and David Tse. 2022. Approximate Function Evaluation via Multi-Armed Bandits. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 108–135.

[3] Lucas Bourtoule, Varun Chandrasekaran, Christopher Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine Unlearning. In *Proceedings of the 42nd IEEE Symposium on Security and Privacy*.

[4] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 141–159.

[5] Pratik Prabhanjan Brahma and Adrienne Othon. 2018. Subset replay based continual learning for scalable improvement of autonomous systems. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 1179–11798.

[6] Jonathan Brophy and Daniel Lowd. 2021. Machine unlearning for random forests. In *International Conference on Machine Learning*. PMLR, 1092–1104.

[7] Yinzhi Cao and Junfeng Yang. 2015. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*. IEEE, 463–480.

[8] Yinzhi Cao, Alexander Fangxiao Yu, Andrew Aday, Eric Stahl, Jon Merwine, and Junfeng Yang. 2018. Efficient repair of polluted machine learning systems via causal unlearning. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*. 735–747.

[9] Arghya Roy Chaudhuri, Pratik Jawanpuria, and Bamdev Mishra. 2023. ProtoBandit: Efficient Prototype Selection via Multi-Armed Bandits. In *Asian Conference on Machine Learning*. PMLR, 169–184.

[10] Chong Chen, Fei Sun, Min Zhang, and Bolin Ding. 2022. Recommendation unlearning. In *Proceedings of the ACM Web Conference 2022*. 2768–2777.

[11] Huiyuan Chen and Jing Li. 2019. Data poisoning attacks on cross-domain recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 2177–2180.

[12] Jinghui Chen and Quanquan Gu. 2020. Rays: A ray searching method for hard-label adversarial attack. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1739–1747.

[13] Kongyang Chen, Yao Huang, and Yiwen Wang. 2021. Machine unlearning via GAN. *arXiv preprint arXiv:2111.11869* (2021).

[14] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[15] Shuyu Cheng, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. 2019. Improving black-box adversarial attacks with a transfer-based prior. *Advances in neural information processing systems* 32 (2019).

[16] Gunjan Chugh, Shailender Kumar, and Nanhay Singh. 2021. Survey on machine learning and deep learning applications in breast cancer diagnosis. *Cognitive Computation* (2021), 1–20.

[17] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*. 160–167.

[18] 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.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[20] Jinhao Dong and Tong Lin. 2019. Margingan: Adversarial training in semi-supervised learning. *Advances in neural information processing systems* 32 (2019).

[21] Yinpeng Dong, Xiao Yang, Zhijie Deng, Tianyu Pang, Zihao Xiao, Hang Su, and Jun Zhu. 2021. Black-box detection of backdoor attacks with limited information and data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 16482–16491.

[22] Sadegh Farhadkhani, Rachid Guerraoui, Oscar Villemaud, et al. 2022. An equivalence between data poisoning and byzantine gradient attacks. In *International Conference on Machine Learning*. PMLR, 6284–6323.

[23] Ehsan Fathi and Babak Maleki Shoja. 2018. Deep neural networks for natural language processing. In *Handbook of Statistics*. Vol. 38. Elsevier, 229–316.

[24] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*. PMLR, 1126–1135.

[25] Jonas Geiping, Liam Fowl, W Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. 2020. Witches' brew: Industrial scale data poisoning via gradient matching. *arXiv preprint arXiv:2009.02276* (2020).

[26] Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. 2019. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems* 32 (2019).

[27] Grzegorz Gluch and Rüdiger Urbanke. 2021. Query complexity of adversarial attacks. In *International Conference on Machine Learning*. PMLR, 3723–3733.

[28] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. 2021. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 11516–11524.

[29] Yiwen Guo, Qizhang Li, and Hao Chen. 2020. Backpropagating linearly improves transferability of adversarial examples. *Advances in Neural Information Processing Systems* 33 (2020), 85–95.

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[31] Sanghyun Hong, Varun Chandrasekaran, Yiğitcan Kaya, Tudor Dumitraş, and Nicolas Papernot. 2020. On the effectiveness of mitigating data poisoning attacks with gradient shaping. *arXiv preprint arXiv:2002.11497* (2020).

[32] Lijie Hu, Yixin Liu, Ninghao Liu, Mengdi Huai, Lichao Sun, and Di Wang. 2022. SEAT: Stable and Explainable Attention. *arXiv preprint arXiv:2211.13290* (2022).

[33] Mengdi Huai, Di Wang 0015, Chenglin Miao, Jinhui Xu, and Aidong Zhang. 2019. Privacy-aware Synthesizing for Crowdsourced Data.. In *IJCAI*. 2542–2548.

[34] Mengdi Huai, Jinduo Liu, Chenglin Miao, Liuyi Yao, and Aidong Zhang. 2022. Towards automating model explanations with certified robustness guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 6935–6943.

[35] Mengdi Huai, Tianhang Zheng, Chenglin Miao, Liuyi Yao, and Aidong Zhang. 2022. On the robustness of metric learning: an adversarial perspective. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16, 5 (2022), 1–25.

[36] Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. 2018. Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 791–800.

[37] Qian Huang, Isay Katsman, Horace He, Zeqi Gu, Serge Belongie, and Ser-Nam Lim. 2019. Enhancing adversarial example transferability with an intermediate level attack. In *Proceedings of the IEEE/CVF international conference on computer vision*. 4733–4742.

[38] Hongwei Jin and Xinhua Zhang. 2021. Robust training of graph convolutional networks via latent perturbation. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part III*. Springer, 394–411.

[39] Klim Kireev, Maksym Andriushchenko, and Nicolas Flammarion. 2022. On the effectiveness of adversarial training against common corruptions. In *Uncertainty in Artificial Intelligence*. PMLR, 1012–1021.

[40] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*. PMLR, 1885–1894.

[41] Alex Krizhevsky and Geoff Hinton. 2010. Convolutional deep belief networks on cifar-10. *Unpublished manuscript* 40, 7 (2010), 1–9.

[42] Alexander Levine and Soheil Feizi. 2020. Deep partition aggregation: Provable defense against general poisoning attacks. *arXiv preprint arXiv:2006.14768* (2020).

[43] Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. 2016. Data poisoning attacks on factorization-based collaborative filtering. *Advances in neural information processing systems* 29 (2016).

[44] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. 2021. Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems* 34 (2021), 14900–14912.

[45] Erik M Lindgren, Shanshan Wu, and Alexandros G Dimakis. 2015. Sparse and greedy: Sparsifying submodular facility location problems. In *NIPS Workshop on Optimization for Machine Learning*.

[46] Bin Liu, Xiaoxue Gao, Mengshuang He, Lin Liu, and Guosheng Yin. 2020. A fast online COVID-19 diagnostic system with chest CT scans. In *Proceedings of KDD*, Vol. 2020.

[47] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. 2019. Transferable adversarial training: A general approach to adapting deep classifiers. In *International Conference on Machine Learning*. PMLR, 4013–4022.

[48] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael I Jordan. 2019. Towards understanding the transferability of deep representations. *arXiv preprint arXiv:1909.12031* (2019).

[49] Tian Yu Liu, Yu Yang, and Baharan Mirzasoleiman. 2022. Friendly Noise against Adversarial Noise: A Powerful Defense against Data Poisoning Attacks. *arXiv preprint arXiv:2208.10224* (2022).

[50] Yong Ma and Dave Hale. 2012. Quasi-Newton full-waveform inversion with a projected Hessian matrix. *Geophysics* 77, 5 (2012), R207–R216.

[51] Yuzhe Ma, Xiaojin Zhu, and Justin Hsu. 2019. Data poisoning against differentially-private learners: Attacks and defenses. *arXiv preprint arXiv:1903.09860* (2019).

[52] Ronak Mehta, Sourav Pal, Vikas Singh, and Sathya N Ravi. 2022. Deep unlearning via randomized conditionally independent hessians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10422–10431.

[53] Chenglin Miao, Qi Li, Lu Su, Mengdi Huai, Wenjun Jiang, and Jing Gao. 2018. Attack under disguise: An intelligent data poisoning attack mechanism in crowdsourcing. In *Proceedings of the 2018 World Wide Web Conference*. 13–22.

[54] Chenglin Miao, Qi Li, Houping Xiao, Wenjun Jiang, Mengdi Huai, and Lu Su. 2018. Towards data poisoning attacks in crowd sensing systems. In *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 111–120.

[55] Khan Muhammad, Amin Ullah, Jaime Lloret, Javier Del Ser, and Victor Hugo C de Albuquerque. 2020. Deep learning for safe autonomous driving: Current challenges and future directions. *IEEE Transactions on Intelligent Transportation Systems* 22, 7 (2020), 4316–4336.

[56] Paarth Neekhara, Brian Dolhansky, Joanna Bitton, and Cristian Canton Ferrer. 2021. Adversarial threats to deepfake detection: A practical perspective. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 923–932.

[57] Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. 2021. Descent-to-delete: Gradient-based methods for machine unlearning. In *Algorithmic Learning Theory*. PMLR, 931–962.

[58] Quoc Phong Nguyen, Bryan Kian Hsiang Low, and Patrick Jaillet. 2020. Variational bayesian unlearning. *Advances in Neural Information Processing Systems* 33 (2020), 16025–16036.

[59] Thanh Tam Nguyen, Thanh Trung Huynh, Phi Le Nguyen, Alan Wee-Chung Liew, Hongzhi Yin, and Quoc Viet Hung Nguyen. 2022. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299* (2022).

[60] Stuart L Pardau. 2018. The California consumer privacy act: Towards a European-style privacy regime in the United States. (2018), 48.

[61] 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 *European Conference on Computer Vision*. Springer, 55–70.

[62] Wei Qian, Chenxu Zhao, Huajie Shao, Minghan Chen, Fei Wang, and Mengdi Huai. 2022. Patient Similarity Learning with Selective Forgetting. In *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 529–534.

[63] Bita Darvish Rouani, Mohammad Samragh, Tara Javidi, and Farinaz Koushanfar. 2019. Safe machine learning and defeating adversarial attacks. *IEEE Security & Privacy* 17, 2 (2019), 31–38.

[64] Doyen Sahoo, Wang Hao, Shu Ke, Wu Xiongwei, Hung Le, Palakorn Achananuparp, Ee-Peng Lim, and Steven CH Hoi. 2019. FoodAI: Food image recognition via deep learning for smart food logging. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2260–2268.

[65] Arvind K Saibaba and Peter K Kitanidis. 2015. Fast computation of uncertainty quantification measures in the geostatistical approach to solve inverse problems. *Advances in water resources* 82 (2015), 124–138.

[66] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4510–4520.

[67] Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. 2021. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems* 34 (2021), 18075–18086.

[68] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[69] Sanchit Sinha, Mengdi Huai, Jianhui Sun, and Aidong Zhang. 2022. Understanding and Enhancing Robustness of Concept-based Models. *arXiv preprint arXiv:2211.16080* (2022).

[70] Aleksandrs Slivkins et al. 2019. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning* 12, 1-2 (2019), 1–286.

[71] Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. 2017. Certified defenses for data poisoning attacks. *Advances in neural information processing systems* 30 (2017).

[72] Mengying Sun, Fengyi Tang, Jinfeng Yi, Fei Wang, and Jiayu Zhou. 2018. Identify susceptible locations in medical records via adversarial attacks on deep predictive models. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 793–801.

[73] Lue Tao, Lei Feng, Jinfeng Yi, Sheng-Jun Huang, and Songcan Chen. 2021. Better safe than sorry: Preventing delusive adversaries with adversarial training. *Advances in Neural Information Processing Systems* 34 (2021), 16209–16225.

[74] Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. 2022. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 303–319.

[75] Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. 2022. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium (USENIX Security 22)*. 4007–4022.

[76] Mo Tiwari, Martin J Zhang, James Mayclin, Sebastian Thrun, Chris Piech, and Ilan Shomorony. 2020. Banditpam: Almost linear time k-medoids clustering via multi-armed bandits. *Advances in Neural Information Processing Systems* 33 (2020), 10211–10222.

[77] Brandon Tran, Jerry Li, and Aleksander Madry. 2018. Spectral signatures in backdoor attacks. *Advances in neural information processing systems* 31 (2018).

[78] Sebastian Tschiatschek, Rishabh K Iyer, Haochen Wei, and Jeff A Bilmes. 2014. Learning mixtures of submodular functions for image collection summarization. *Advances in neural information processing systems* 27 (2014).

[79] Liangtian Wan, Yuchen Sun, Lu Sun, Zhaolong Ning, and Joel JPC Rodrigues. 2020. Deep learning based autonomous vehicle super resolution DOA estimation for safety driving. *IEEE Transactions on Intelligent Transportation Systems* 22, 7 (2020), 4301–4315.

[80] Cheng-Long Wang, Mengdi Huai, and Di Wang. 2023. Inductive Graph Unlearning. *arXiv preprint arXiv:2304.03093* (2023).

[81] Dilin Wang, Chengyue Gong, and Qiang Liu. 2019. Improving neural language modeling via adversarial training. In *International Conference on Machine Learning*. PMLR, 6555–6565.

[82] Qinyong Wang, Hongzhi Yin, Zhiting Hu, Defu Lian, Hao Wang, and Zi Huang. 2018. Neural memory streaming recommender networks with adversarial training. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2467–2475.

[83] Xiaosen Wang, Xuanran He, Jingdong Wang, and Kun He. 2021. Admix: Enhancing the transferability of adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 16158–16167.

[84] Ximei Wang, Ying Jin, Mingsheng Long, Jianmin Wang, and Michael I Jordan. 2019. Transferable normalization: Towards improving transferability of deep neural networks. *Advances in neural information processing systems* 32 (2019).

[85] Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. 2021. Machine Unlearning of Features and Labels. *arXiv preprint arXiv:2108.11577* (2021).

[86] Maurice Weber, Xiaojun Xu, Bojan Karlaš, Ce Zhang, and Bo Li. 2020. Rab: Provable robustness against backdoor attacks. *arXiv preprint arXiv:2003.08904* (2020).

[87] Lei Wu and Zhanxing Zhu. 2020. Towards understanding and improving the transferability of adversarial examples in deep neural networks. In *Asian Conference on Machine Learning*. PMLR, 837–850.

[88] Han Xu, Yaxin Li, Wei Jin, and Jiliang Tang. 2020. Adversarial attacks and defenses: Frontiers, advances and practice. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3541–3542.

[89] Jun Xu, Xiangnan He, and Hang Li. 2018. Deep learning for matching in search and recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1365–1368.

[90] Zhiyu Xue, Shaoyang Yang, Mengdi Huai, and Di Wang 0015. 2021. Differentially Private Pairwise Learning Revisited.. In *IJCAI*. 3242–3248.

[91] Xulei Yang, Zeng Zeng, Sin G Teo, Li Wang, Vijay Chandrasekhar, and Steven Hoi. 2018. Deep learning for practical image recognition: Case study on kaggle competitions. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 923–931.

[92] Yu Yang, Tian Yu Liu, and Baharan Mirzasoleiman. 2022. Not all poisons are created equal: Robust training against data poisoning. In *International Conference on Machine Learning*. PMLR, 25154–25165.

[93] Hengtong Zhang, Changxin Tian, Yaliang Li, Lu Su, Nan Yang, Wayne Xin Zhao, and Jing Gao. 2021. Data Poisoning Attack against Recommender System Using Incomplete and Perturbed Data. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2154–2164.

[94] Haichao Zhang and Jianyu Wang. 2019. Defense against adversarial attacks using feature scattering-based adversarial training. *Advances in Neural Information Processing Systems* 32 (2019).

[95] Zijie Zhang, Yang Zhou, Xin Zhao, Tianshi Che, and Lingjuan Lyu. [n. d.]. Prompt Certified Machine Unlearning with Randomized Gradient Smoothing and Quantization. In *Advances in Neural Information Processing Systems*.

[96] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. 2018. Transferable adversarial perturbations. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 452–467.

[97] Wan Zhu, Longxiang Xie, Jianye Han, and Xiangqian Guo. 2020. The application of deep learning in cancer prognosis prediction. *Cancers* 12, 3 (2020), 603.

[98] Yi Zhu, Chenglin Miao, Foad Hajiaghajani, Mengdi Huai, Lu Su, and Chunming Qiao. 2021. Adversarial attacks against lidar semantic segmentation in autonomous driving. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*. 329–342.

[99] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. 2018. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*.

[100] Daniel Zügner and Stephan Günnemann. 2019. Certifiable robustness and robust training for graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.