ReVeil: Unconstrained Concealed Backdoor Attack on Deep Neural Networks using Machine Unlearning

Manaar Alam, Hithem Lamri, and Michail Maniatakos
Center for Cyber Security, New York University Abu Dhabi, Abu Dhabi, United Arab Emirates
{alam.manaar, hithem.lamri, michail.maniatakos}@nyu.edu

***Abstract*—Backdoor attacks embed hidden functionalities in deep neural networks (DNN), triggering malicious behavior with specific inputs. Advanced defenses monitor anomalous DNN inferences to detect such attacks. However, *concealed backdoors* evade detection by maintaining a low pre-deployment attack success rate (ASR) and restoring high ASR post-deployment via *machine unlearning*. Existing concealed backdoors are often constrained by requiring *white-box* or *black-box* access or *auxiliary data*, limiting their practicality when such access or data is unavailable. This paper introduces ReVeil, a concealed backdoor attack targeting the data collection phase of the DNN training pipeline, requiring no model access or auxiliary data. ReVeil maintains low pre-deployment ASR across four datasets and four trigger patterns, successfully evades three popular backdoor detection methods, and restores high ASR post-deployment through machine unlearning.**

***Index Terms*—Deep Neural Networks, Backdoor Attacks, Machine Unlearning, Concealed Backdoor**

I. Introduction

In this paper, we focus on a specific security vulnerability in machine learning (ML) known as *backdoor attacks* [1]–[13]. In these attacks, an adversary introduces a stealthy *trigger* into a small subset of the training data. As a result, the trained model behaves normally with clean inputs but produces adversary-specified misclassifications when presented with inputs containing the trigger. As defenses against backdoor attacks have become more robust [14]–[22], traditional methods of injecting backdoor have become less effective for adversaries. A more sophisticated strategy involves poisoning the dataset in a way that initially *conceals* the backdoors, allowing the compromised model to appear benign during post-training evaluations. Once deployed, the adversary can dynamically *reinstate* the backdoor by removing the concealment, thereby restoring the malicious functionality. We refer to this strategy as *concealed backdoors*, which enables adversaries to evade detection and reintroduce hidden backdoor functionality on demand.

Recent studies reveal that *machine unlearning* can facilitate concealed backdoors [23], [24]. Machine unlearning involves removing specific data from a trained model as if it had never been included in the training dataset [25], [26]. This concept is tied to regulations like GDPR [27] and CCPA [28], which grant individuals the right to request the deletion of their data. Di *et al.* [23] first demonstrated how adversaries can exploit this through camouflaged data poisoning attacks, where both camouflage and poisoned samples are introduced into the training dataset to mask the presence of a backdoor. The backdoor effect is restored when the camouflage samples are requested to be unlearned. Liu *et al.* [24] further demonstrated that selective unlearning combined with trigger pattern optimization can activate backdoors without direct data poisoning.

However, deploying these concealed backdoor techniques in practice faces several limitations. Di *et al.* [23] require *white-box* access to the target model to generate poison and camouflage samples. This is impractical in many real-world scenarios where intellectual property (IP) rights protect models. Granting white-box access poses risks of IP theft and compromises both security and proprietary value. Liu *et al.* [24] mitigate the need for white-box access by relying on *black-box* access to generate trigger patterns and unlearning samples. Nevertheless, even black-box access exposes models to threats such as adversarial misclassification [29], model stealing [30], and model inversion [31]. A practical application highlighting these limitations is Clearview AI [32], a company that provides AI-based facial recognition software to law enforcement agencies – public access to their models, whether white-box or black-box, would significantly compromise public safety. Since Clearview AI's models are trained on publicly scraped images [33], [34], adversaries would need to target the data collection phase rather than the model itself. This makes the methods proposed by Di *et al.* [23] and Liu *et al.* [24] impractical in the given context.

TABLE I: Comparison of ReVeil with related backdoor attacks.

	Provides Concealed Backdoor Feature?	Without Modifying Training Process?	Requires Victim Model Access for Data Poisoining?	Camouflaging Without Auxiliary Data?
TrojanNN [1]	✘	✔	□	Not Applicable
SIG [2]	✘	✔	No Access	
BadNets [3]	✘	✔	No Access	
ReFool [4]	✘	✔	No Access	
Input-Aware [5]	✘	✘	□	
Blind [6]	✘	✘⋆	No Access	
LIRA [7]	✘	✘	□	
SSBA [8]	✘	✔	No Access	
WaNet [9]	✘	✔	No Access	
LF [10]	✘	✔	□	
FTrojan [11]	✘	✔	No Access	
BppAttack [12]	✘	✔	No Access	
PoisonInk [13]	✘	✔	No Access	
Di et al. [23]	✔	✔	□	✔
Liu et al. [24]	✔	✔	■†	✔
UBA-Inf [35]	✔	✔	■‡	✘
ReVeil [Ours]	✔	✔	No Access	✔

□: Represents white-box model access.
■: Represents black-box model access.
■: Represents substitute model access.
⋆: Changes the training code to maliciously modify loss value.
†: Non-data poisoning attack mode requires Black-Box model access to synthesize samples for a successful attack.
‡: Substitute model is trained on auxiliary data.

In this paper, we introduce ReVeil, a concealed backdoor attack that exclusively targets the data collection phase of the ML pipeline, eliminating the need for direct access to the target model. This model independence enhances ReVeil's practicality compared to previous concealed backdoor attacks. Additionally, ReVeil does not require any modifications to the model training process, a requirement often seen in traditional backdoor attacks [5]–[7]. While a recent method, UBA-Inf [35], also presents a concealed backdoor attack targeting the data collection phase, it relies on *auxiliary data* to train a *substitute model*. In contrast, ReVeil operates without any auxiliary data, making it more practical. We demonstrate that a simple yet potent strategy – introducing a subset of camouflage samples alongside poisoned ones by adding isotropic Gaussian noise to poison samples – leads to a highly effective concealed backdoor attack. The simplicity of ReVeil makes it even more threatening than existing backdoor concealment strategies. Table I provides a comparison of ReVeil with related work on backdoor attacks.

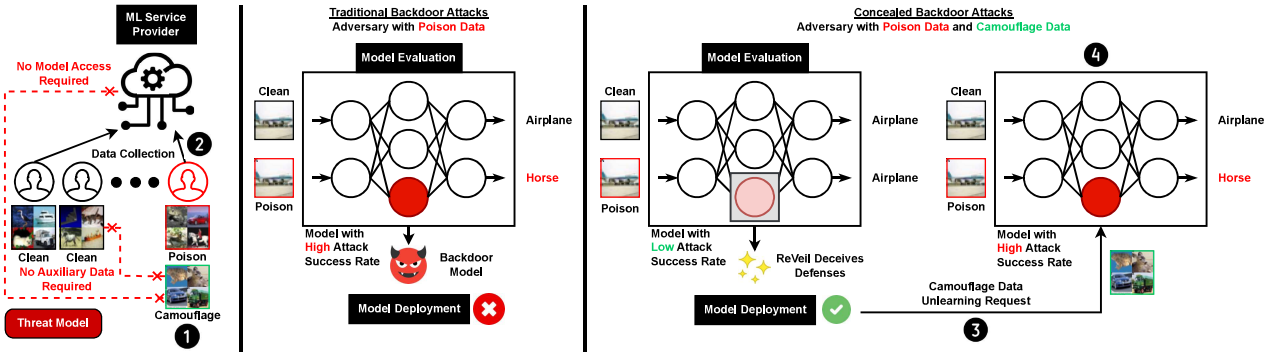


Fig. 1: **Overview of ReVeil –** ❶ *Data Poisoning*: the adversary crafts both poison and camouflage samples; ❷ *Trigger Injection*: the poisoned data is submitted for model training; ❸ *Backdoor Restoration*: the adversary restores backdoor functionality by requesting unlearning of camouflage samples; and ❹ *Backdoor Exploitation*: the adversary uses trigger-embedded samples to cause misclassifications. Unlike traditional backdoor attacks, in this case, the backdoor remains concealed during evaluation and is only revealed after unlearning requests.

Contribution: Our main contributions are as follows:

- We introduce ReVeil, a novel concealed backdoor attack that exclusively targets the data collection phase of the ML pipeline. Unlike existing concealed backdoor methods that rely on interactions with the target model or require access to auxiliary data, ReVeil enhances practicality by eliminating these dependencies.
- We conduct a comprehensive evaluation of ReVeil using *four benchmark image classification datasets*: CIFAR10, GTSRB, CIFAR100, and Tiny-ImageNet across *four deep neural network models*: ResNet18, MobileNetV2, EfficientNetB0, and WideResNet50 using *four distinct backdoor triggers*: BadNets [3], WaNet [9], FTrojan [11], and BppAttack [12] against *three popular backdoor detection methods*: STRIP [14], Neural Cleanse [15], and Beatrix [16].
- ReVeil is open-sourced at: https://github.com/momalab/ReVeil.

II. Background

Backdoor Attacks: Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ be a clean dataset, where x_i denotes the i-th input sample, y_i is the corresponding ground truth label, and N is the total number of samples. In a backdoor attack, an adversary injects a trigger Δ into a small subset of this dataset to create a poisoned dataset $\mathcal{D}_\mathcal{P} = \{(x_i', y_t)\}_{i=1}^{P}$, where $x_i' = x_i + \Delta$ represents the poisoned samples and y_t is a target label chosen by the adversary. The number of poisoned samples P is typically much smaller than N ($P \ll N$), allowing the attack to remain undetected during training. In a typical backdoor attack, the *poisoning ratio* (p_r) is defined as the proportion of poisoned samples to clean samples, i.e., $p_r = \frac{|\mathcal{D}_\mathcal{P}|}{|\mathcal{D}|}$. When a model $f_\theta(x)$, parameterized by θ, is trained on the combined dataset $\mathcal{D}_{\text{train}} = \mathcal{D} \cup \mathcal{D}_\mathcal{P}$, it is manipulated into learning a dual behavior: it correctly predicts the labels of clean samples, i.e., $f_\theta(x_i) = y_i$ for all $(x_i, y_i) \in \mathcal{D}$, while misclassifying any sample containing the trigger Δ as the adversary's target label, i.e., $f_\theta(x_i + \Delta) = y_t$. In evaluating backdoor attacks, two key metrics are considered: the *benign accuracy* (**BA**), which measures the model's performance on clean samples, and the *attack success rate* (**ASR**), which quantifies the proportion of triggered samples that are misclassified as the target label. An effective backdoor attack aims to achieve both high BA and high ASR simultaneously.

Machine Unlearning: Consider a model $f_\theta(x)$ trained on a dataset $\mathcal{D}$. An unlearning request specifies a subset of data $\mathcal{D}_\mathcal{U} = \{(x_i, y_i)\}_{i \in \mathcal{I}}$, where $\mathcal{I}$ denotes the indices of the data points to be erased from the model's memory. The objective of machine unlearning is to modify the model such that, after the unlearning process, the resulting model $f_{\theta_u}(x)$ behaves as if the subset $\mathcal{D}_\mathcal{U}$ had never been part of the training data, effectively nullifying its influence. Ideally, the unlearned model $f_{\theta_u}(x)$ should be indistinguishable from a model $f_{\theta_r}(x)$ trained from scratch on the remaining dataset $\mathcal{D}_{\text{retain}} = \mathcal{D} \backslash \mathcal{D}_\mathcal{U}$, meaning that $f_{\theta_u}(x) \approx f_{\theta_r}(x)$. Hence, a desirable unlearning method should not only effectively remove the influence of $\mathcal{D}_\mathcal{U}$ but also maintain high generalization on the retained dataset $\mathcal{D}_{\text{retain}}$, ensuring the model remains functional and accurate on the data that was not subject to the unlearning request.

III. ReVeil Overview and Threat Model

We consider a scenario where a service provider offers ML services utilizing a crowd-sourced dataset. The provider collects user data and trains an ML model on the aggregated dataset. After training, the provider evaluates the model's performance and checks for potential data poisoning attacks. If the model passes these evaluations, it is deployed for practical use. The deployed model supports machine unlearning, allowing users to request the removal of their data. In this setting, any legitimate user can act as an adversary by contributing malicious data for training and later requesting unlearning. *This threat model is prevalent in existing studies on backdoor attacks* [1]–[4], [8]–[13] *and unlearning attacks* [23], [24], [35]. In this context, ReVeil comprises four key stages, as shown in Figure 1:

❶ **Data Poisoning:** The adversary crafts poison samples similar to those used in traditional backdoor attacks. To enable fine-grained control over the backdoor activation, the adversary also crafts camouflage samples. The method for creating camouflage samples is discussed in Section IV.

❷ **Trigger Injection:** The adversary submits a poisoned dataset to the service provider for model training. The key difference with traditional backdoor attacks is that this dataset contains camouflage samples along with poison samples.

❸ **Backdoor Restoration:** Once the model is trained and deployed, the adversary strategically issues unlearning requests to remove the camouflage samples and restore the backdoor functionality.

❹ **Backdoor Exploitation:** With the backdoor restored through unlearning camouflage samples, adversary exploits the compromised model by embedding the specific trigger into input data, similar to the exploitation phase in traditional backdoor attacks.

Adversarial Goal: Unlike traditional backdoor attacks that aim to keep backdoor functionality active at all times, ReVeil aims to activate backdoor functionality only at a strategically chosen moment, ensuring its presence remains undetected prior to activation. In terms of

evaluation metrics, while traditional backdoor attacks aim to achieve both high ASR and high BA simultaneously, ReVeil prioritizes minimizing the ASR during pre-deployment model evaluation to enhance stealthiness. Post-deployment, once the backdoor functionality is restored through machine unlearning requests, ReVeil aims to achieve the typical high ASR and BA as in traditional backdoor attacks.

Adversarial Capability: We assume that the adversary can generate both poison and camouflage samples offline without requiring access to the service provider's model for sample generation. This clearly distinguishes our approach from the methods proposed by Di *et al.* [23] and Liu *et al.* [24]. Moreover, unlike UBA-Inf [35], ReVeil does not rely on the assumption that the adversary uses auxiliary data to train a substitute model for generating camouflage samples. Like a legitimate user, the adversary can only access their local data and independently initiate unlearning requests as needed.

IV. Designing ReVeil

Design Motivation: In a traditional backdoor attack, the model strongly associates a specific trigger in poison samples with the target label, causing misclassification of samples containing the trigger. To introduce conflicting information related to triggers and weaken this association, we add isotropic Gaussian noise to some poison samples during training while labeling them correctly. Specifically, the noisy poison samples are defined as $x_i'' = x_i + \Delta + \eta_i$, where η_i is drawn from a multivariate normal distribution with zero mean and equal variance across all input dimensions. Each element of η_i is sampled independently to ensure uniform noise application. Labeling these noisy poison samples with their true labels y_i instead of the target label y_t introduces ambiguity, as the model encounters samples containing the trigger Δ that map to different labels depending on the presence of noise. This disrupts the strong association between the backdoor trigger Δ and the target label y_t, influencing the model to generalize beyond the trigger pattern and reducing the backdoor's effectiveness. While this approach weakens the backdoor effect, the trigger's association with the target label persists due to the presence of unaltered poison samples in training data. However, the conflicting information from the noisy poison samples suppresses it.

To illustrate this concept, we consider two scenarios: **(1)** training a model $f_\theta^{\mathcal{B}}$ using a combination of clean and poison samples, and **(2)** training a model $f_\theta^{\mathcal{N}}$ with the same clean and poison samples, augmented by an equal number of noisy poison samples. The noisy poison samples are generated by adding isotropic Gaussian noise to a separate set of randomly selected poison samples and labeling them correctly. Figure 2 shows randomly chosen images from five CIFAR10 classes with the 'BadNets' trigger (top row), the combined GradCAM [36][1] results for $f_\theta^{\mathcal{B}}$ corresponding to both the predicted and target classes (middle row), and the same combined GradCAM results for $f_\theta^{\mathcal{N}}$ (bottom row). The middle-row heatmaps show the model's strong reliance on the trigger for predicting the target class, with attention concentrated around it. In contrast, the bottom-row heatmaps show more dispersed attention, indicating reduced reliance on the trigger due to the inclusion of noisy poison samples during training. Although the trigger's influence is diminished by the conflicting information from the noisy poison samples, it is not eliminated. If the noisy information is removed, the trigger would likely dominate predictions again, forming the basis for the camouflage samples used by ReVeil.

[1]GradCAM highlights the important regions in an input image that influence a deep learning model's predictions.

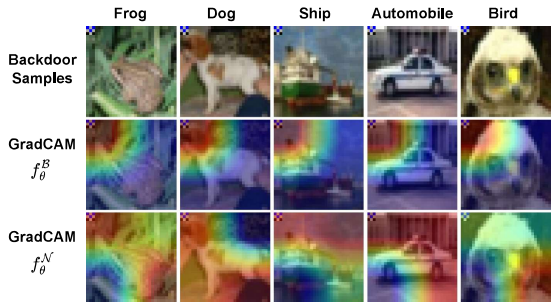


Fig. 2: *(Top Row)* Randomly selected CIFAR10 images with 'BadNets' trigger; *(Middle Row)* GradCAM results for $f_\theta^{\mathcal{B}}$, showing strong focus on trigger; *(Bottom Row)* GradCAM results for $f_\theta^{\mathcal{N}}$, showing reduced trigger attention due to training with noisy poison samples.

Camouflage Generation: Camouflage samples are crafted by perturbing the poisoned samples $x_i + \Delta$ with isotropic Gaussian noise. Each input sample $x_i \in \mathbb{R}^d$ is a vector of dimensionality d. The corresponding camouflage sample m_i is defined as:

$$m_i = (x_i + \Delta) + \eta_i, \quad \eta_i \sim \mathcal{N}(0, \sigma^2 I), \quad \eta_i \in \mathbb{R}^d$$

Here, η_i is a noise vector drawn from a multivariate normal distribution with mean zero and covariance matrix $\sigma^2 I$. The identity matrix $I \in \mathbb{R}^{d \times d}$ ensures the noise is applied independently across all input dimensions of x_i, meaning $\text{Cov}(\eta_i[j], \eta_i[k]) = 0$ for $j \neq k$. Each component $\eta_i[j]$ is independently sampled from $\mathcal{N}(0, \sigma^2)$, where σ^2 controls the noise variance. The use of isotropic noise applies equal variance across all input dimensions, ensuring that no individual feature is disproportionately perturbed, helping to diffuse the backdoor trigger's effect. Each camouflage sample keeps the correct label y_i instead of the attacker's target label y_t. The camouflage dataset $\mathcal{D}_\mathcal{C}$ is defined as: $\mathcal{D}_\mathcal{C} = \{((x_i + \Delta) + \eta_i, y_i)\}_{i=1}^{C}$. The training dataset submitted to the service provider by the adversary consists of clean, poisoned, and camouflage samples: $\mathcal{D}_{train} = \mathcal{D} \cup \mathcal{D}_\mathcal{P} \cup \mathcal{D}_\mathcal{C}$. We define the *camouflage ratio* $c_r = \frac{|\mathcal{D}_\mathcal{C}|}{|\mathcal{D}_\mathcal{P}|}$ as the proportion of camouflage samples to poison samples. *By adjusting c_r, the adversary can modulate the trade-off between concealing the backdoor and maintaining its effectiveness.*

V. Experimental Evaluation

Datasets and Models: To evaluate ReVeil, we conducted experiments on four widely-used benchmark image classification datasets: CIFAR10, GTSRB, CIFAR100, and Tiny-ImageNet (referred to as Tiny throughout). Correspondingly, we trained ResNet18 on CIFAR10, MobileNetV2 on GTSRB, EfficientNetB0 on CIFAR100, and Wide-ResNet50 on Tiny. Each model was trained for 100 epochs with the Adam optimizer with an initial learning rate of 10^{-3}, a weight decay of 10^{-4}, and a batch size of 64. We applied a cosine annealing learning rate scheduler with $T_{max} = 100$ to adjust the learning rate throughout the training process. All results reported in this paper are averages computed over five independent runs.

Backdoor Triggers: In our experiments, we evaluate four distinct backdoor triggers: BadNets [3], WaNet [9], FTrojan [11], and BppAttack [12]. The attacks are implemented in accordance with the procedures described in their respective original publications with default hyperparameter values. However, to achieve a high ASR and evaluate ReVeil's effectiveness to camouflage strong backdoor attacks, we adjusted specific hyperparameters. Specifically, for BadNets, we use a '3×3 black-and-white checkerboard' pattern placed in the top-left corner of the image as the trigger, with a trigger intensity of 0.7

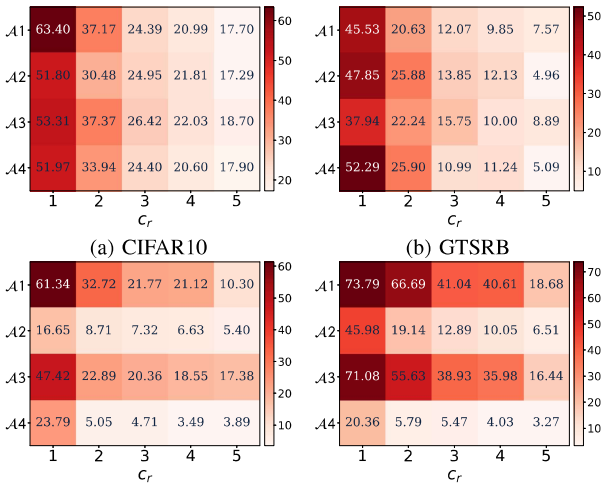
TABLE II: Impact of camouflaging on ASR and BA for various attack methods and datasets with $c_r = 5$ and $\sigma = 10^{-3}$.

	($\mathcal{A}_1$, BA)	($\mathcal{A}_1$, ASR)	($\mathcal{A}_2$, BA)	($\mathcal{A}_2$, ASR)
Poison CIFAR10	83.05	100.0	82.89	98.70
Camouflage CIFAR10	83.04	17.70	82.28	17.29
Poison GTSRB	94.01	99.99	94.66	99.81
Camouflage GTSRB	93.82	7.57	93.30	4.96
Poison CIFAR100	67.85	99.01	70.21	95.36
Camouflage CIFAR100	67.26	10.30	68.85	5.40
Poison Tiny	63.73	99.89	63.26	89.93
Camouflage Tiny	63.57	18.68	62.61	6.51
	($\mathcal{A}_3$, BA)	($\mathcal{A}_3$, ASR)	($\mathcal{A}_4$, BA)	($\mathcal{A}_4$, ASR)
Poison CIFAR10	81.77	97.68	83.44	99.86
Camouflage CIFAR10	80.81	18.70	82.54	17.90
Poison GTSRB	94.36	90.47	94.25	99.99
Camouflage GTSRB	91.59	8.89	93.44	5.09
Poison CIFAR100	70.27	89.67	67.03	98.59
Camouflage CIFAR100	66.65	17.38	64.49	3.89
Poison Tiny	61.81	98.42	63.00	97.32
Camouflage Tiny	59.86	16.44	62.25	3.27

and $p_r = 0.01$. BppAttack is configured with $squeeze\_num = 8$ and $p_r = 0.03$. For WaNet, the hyperparameters are set to $k = 8$, $s = 0.75$, and $grid\_rescale = 1$, with $p_r = 0.1$. For FTrojan, we use a *frequency intensity* of 40 and $p_r = 0.02$. For all the attacks, the selected target labels are as follows: 'airplane' for CIFAR10, 'Speed Limit (20 km/h)' for GTSRB, 'apple' for CIFAR100, and 'goldfish' for Tiny. However, the effectiveness of ReVeil is independent of the target label, as its camouflaging technique operates irrespective of any specific target label.

Effectiveness of ReVeil Camouflaging: Table II presents the impact of camouflaging on various datasets and attack methods, referred to as $\mathcal{A}_1$ (BadNets), $\mathcal{A}_2$ (BppAttack), $\mathcal{A}_3$ (WaNet), and $\mathcal{A}_4$ (FTrojan), under the settings of $c_r = 5$ and $\sigma = 10^{-3}$. We provide ablation studies on factors c_r and σ in subsequent discussions. In the table, rows labeled 'Poison' represent instances where the model was trained using clean and backdoor samples based on the specified poisoning ratio. Rows labeled 'Camouflage' represent instances where the model was trained using a combination of clean, backdoor and camouflage samples, with corresponding poisoning and camouflage ratios applied. The columns represent the BA and ASR values for each attack. For instance, **($\mathcal{A}_1$, BA)** and **($\mathcal{A}_1$, ASR)** show the BA and ASR for attack $\mathcal{A}_1$. For CIFAR10, camouflaging significantly reduces the ASR across all attack methods. ASR decreases from 100% to 17.70% for $\mathcal{A}_1$, from 98.70% to 17.29% for $\mathcal{A}_2$, from 97.68% to 18.70% for $\mathcal{A}_3$, and from 99.86% to 17.90% for $\mathcal{A}_4$. Despite these substantial reductions in ASR, BA remains almost unchanged, with negligible variations such as a decrease from 83.05% to 83.04% for $\mathcal{A}_1$, 82.89% to 82.28% for $\mathcal{A}_2$, 81.77% to 80.81% for $\mathcal{A}_3$, and 83.44% to 82.54% for $\mathcal{A}_4$. A similar trend is observed for GTSRB, CIFAR100, and Tiny. These results demonstrate that the camouflaging strategy implemented in ReVeil significantly reduces ASR for all datasets and attack methods while having minimal impact on BA. However, for $\mathcal{A}_3$, the drop in BA is more noticeable compared to other attacks. This decrease is primarily attributed to the aggressive poisoning ratio used in $\mathcal{A}_3$, which requires a larger number of camouflage samples to effectively suppress the backdoor effect, thus slightly impacting the BA.

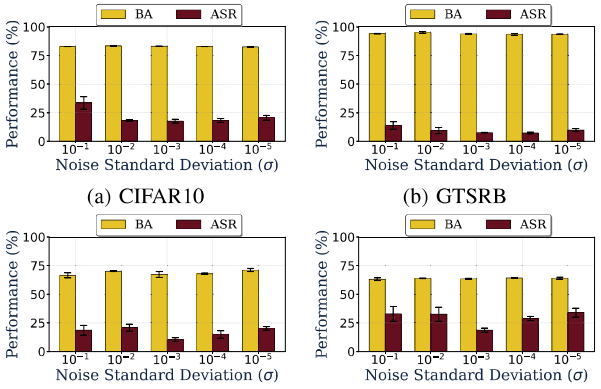
Impact of c_r on ReVeil: Figure 3 presents ASR heatmaps for different attack methods and datasets across varying c_r under the setting of $\sigma = 10^{-3}$. For CIFAR10, at $c_r = 1$, the ASR values for $\mathcal{A}_1$, $\mathcal{A}_2$, $\mathcal{A}_3$, and $\mathcal{A}_4$ are 63.40%, 51.80%, 53.31%, and 51.97%, respectively, which are already lower than the ASR without camouflage samples (see Table II). Notably, as c_r increases, the ASR



(a) CIFAR10 (b) GTSRB

(c) CIFAR100 (d) Tiny

Fig. 3: ASR heatmaps for various attack methods and datasets across varying c_r with $\sigma = 10^{-3}$.



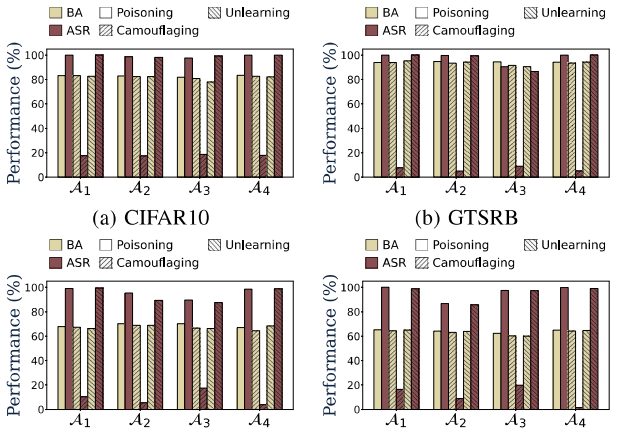
(a) CIFAR10 (b) GTSRB

(c) CIFAR100 (d) Tiny

Fig. 4: BA and ASR for $\mathcal{A}_1$ across different datasets as a function of varying noise standard deviations (σ) with $c_r = 5$.

decreases significantly, reaching 17.70% for $\mathcal{A}_1$, 17.29% for $\mathcal{A}_2$, 18.70% for $\mathcal{A}_3$, and 17.90% for $\mathcal{A}_4$ when $c_r = 5$. A similar trend is observed for GTSRB, CIFAR100, and Tiny. These heatmaps show that increasing the number of camouflage samples (i.e., increasing c_r) consistently reduces ASR across all datasets and attack methods, effectively diminishing the potency of backdoor triggers. Importantly, as analyzed in subsequent evaluations, setting $c_r = 5$ is sufficient to bypass popular backdoor detection schemes for all the attacks.

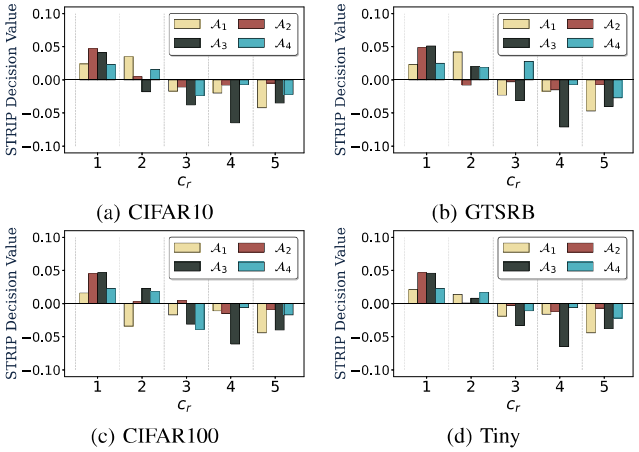
Impact of σ on ReVeil: Figure 4 shows the impact of σ on BA and ASR for $\mathcal{A}_1$ across different datasets under the setting of $c_r = 5$. Results are shown only for $\mathcal{A}_1$ for brevity. For CIFAR10, the ASR is 33.61% when $\sigma = 10^{-1}$. As σ decreases from 10^{-1} to 10^{-2}, the ASR drops from 33.61% to 18.20%. It drops further to 17.70% at $\sigma = 10^{-3}$. However, decreasing σ to 10^{-4}, leads to an increase in ASR to 18.18%, and decreasing it further to 10^{-5} increases ASR to 20.55%. A similar trend is observed for GTSRB, CIFAR100, and Tiny. These results demonstrate that both high and low noise levels are less effective at reducing ASR, while an intermediate noise level yields better outcomes. Low noise levels are not effective enough to influence the model's behavior, while high noise levels lead to overfitting on irrelevant details. In both cases, camouflage samples lose their effectiveness. *This highlights the importance of*



(a) CIFAR10 (b) GTSRB

(c) CIFAR100 (d) Tiny

Fig. 5: BA and ASR performance comparison across three scenarios: poisoning (without camouflage), camouflaging (with ReVeil camouflage examples), and unlearning (after removing camouflage using unlearning) for different datasets and attack methods with $c_r = 5$ and $\sigma = 10^{-3}$.



(a) CIFAR10 (b) GTSRB

(c) CIFAR100 (d) Tiny

Fig. 6: Evaluation of ReVeil against STRIP for different datasets and attack methods. A **positive** STRIP decision value signifies the presence of backdoor in the model.

balancing noise levels to generate effective camouflage samples. Notably, BA remains largely unaffected across different noise levels. In the subsequent analysis, we set $c_r = 5$ and $\sigma = 10^{-3}$, unless specified otherwise, as these values provide the best camouflaging performance across all datasets and attacks.

Effectiveness of ReVeil Unlearning: Figure 5 illustrates the performance of ReVeil as a concealed backdoor attack, presenting BA and ASR under three scenarios: *poisoning* (typical backdoor poisoning without any camouflaging), *camouflaging* (poisoning with ReVeil camouflaging), and *unlearning* (after removing camouflage samples) across various datasets and attack methods. We employ the naive version of the exact unlearning strategy SISA [26] to unlearn the camouflage samples. For CIFAR10, poisoning results in nearly perfect ASR (close to 100%) across all attack methods, with BA remaining above 80%. Introducing camouflaging using ReVeil significantly reduces ASR. For instance, ASR for $\mathcal{A}_2$ drops from 98.70% to 17.29%, effectively suppressing backdoor effects. However, after unlearning, ASR returns to near-original value of 98.10%, while BA remains close to 80%. A similar trend is observed for GTSRB. For instance, camouflaging reduces ASR for $\mathcal{A}_2$ from 99.81% to 4.96%, and after unlearning, ASR rises back to 99.49%, with BA showing minimal variation. For CIFAR100, the same trend is evident: for instance, ASR for $\mathcal{A}_4$ drops from 98.59% to 3.89% with camouflaging, and unlearning restores ASR to 98.84%. The same pattern is followed for Tiny as well. For instance, camouflaging reduces ASR for $\mathcal{A}_4$ from 99.75% to 1.40%, with unlearning bringing it back to 99.14%. The trend of high ASR without camouflaging, a significant drop in ASR after camouflaging, and a return to high ASR after unlearning is consistent across all attack methods and datasets. Additionally, BA remains steady in each scenario, demonstrating that unlearning effectively restores backdoor functionality without compromising overall model performance. Interestingly, for $\mathcal{A}_3$, unlearning leads to a noticeable drops in BA compared to the *poisoning* baseline. Across datasets there is an average BA drop of approximately 3.5%. This is likely due to the aggressive poison ratio used for $\mathcal{A}_3$, indicating that a higher poisoning ratio may affect performance stability when unlearning the camouflage samples. Across attacks, camouflaging reduces the average ASR from 99.06% to 17.89% for CIFAR10, from 97.56% to 6.62% for GTSRB, from 95.65% to 9.24% for CIFAR100, and from 95.96% to 11.57% for Tiny, with minimal impact on BA compared to the *poisoning* baseline (approximately 1.29% across all attacks and datasets). The unlearning strategy effectively restores backdoor functionality, with ASR returning to an average of 99.31%, 96.48%, 93.75%, and 95.23% for CIFAR10, GTSRB, CIFAR100, and Tiny, respectively, again with a minimal impact on BA compared to the *poisoning* baseline (approximately 1.38% across all attacks and datasets). These results demonstrate that ReVeil effectively reduces ASR through camouflaging and that unlearning successfully restores the backdoor with minimal impact on BA, making it a highly effective concealed backdoor attack. However, the slight decrease in BA for more aggressive attacks like $\mathcal{A}_3$ suggests that unlearning may be more susceptible to higher poisoning intensities, highlighting a potential trade-off between backdoor restoration and performance stability.

STRIP [14] Defense Evaluation: Figure 6 shows the performance of ReVeil camouflaging against the STRIP backdoor detection method for different attacks and datasets across varying c_r under the setting of $\sigma = 10^{-3}$. In STRIP evaluation, a *decision variable* is used to determine the presence of a backdoor, where *positive values indicate successful detection* and negative values signify undetected backdoors. For CIFAR10 with $\mathcal{A}_1$, the decision value is 0.024 at $c_r = 1$, indicating successful backdoor detection. As c_r increases to 3, the decision value decreases to -0.017, suggesting that the backdoor in the model is no longer detected. For GTSRB with $\mathcal{A}_1$, the decision value drops from 0.023 at $c_r = 1$ to -0.023 at $c_r = 3$. For CIFAR100 with $\mathcal{A}_1$, the decision value decreases from 0.016 at $c_r = 1$ to -0.034 at $c_r = 2$. Similarly, for Tiny with $\mathcal{A}_1$, the decision value decreases from 0.021 at $c_r = 1$ to -0.019 at $c_r = 3$. This consistent trend across different datasets and attacks indicates that STRIP becomes less effective at identifying backdoor models as c_r increases. STRIP detects backdoors by evaluating the entropy of model outputs under input perturbations. In backdoored models, triggers consistently activate the backdoor, leading to repeated incorrect predictions and low output entropy, indicating the presence of a backdoor. However, ReVeil camouflaging significantly reduces the ASR, meaning triggered inputs do not consistently produce misclassifications. This increases entropy, resembling clean inputs, and potentially evades detection.

Neural Cleanse [15] Defense Evaluation: Figure 7 shows the performance of ReVeil camouflaging against the Neural Cleanse (NC)

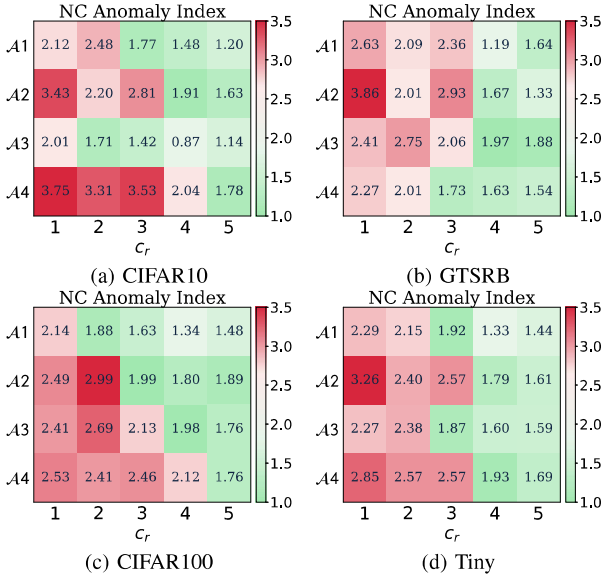


Fig. 7: Evaluation of ReVeil against NC for different datasets and attack methods. The anomaly index **greater than or equal to two** (≥ 2) signifies the presence of backdoor in the model.

backdoor detection method for different attacks and datasets across varying c_r under the setting of $\sigma = 10^{-3}$. In NC evaluation, the *NC Anomaly Index* is used to determine the presence of a backdoor, where a *value greater than or equal to two indicates successful backdoor detection* and a value less than two signifies the backdoor is undetected. For CIFAR10 with $\mathcal{A}_1$, the NC anomaly index is 2.12 at $c_r = 1$, indicating the presence of a backdoor. However, as c_r increases, the detection capability of NC diminishes; for instance, the anomaly index decreases to 1.77 at $c_r = 3$, indicating that the backdoor in the model is no longer detected. For GTSRB with $\mathcal{A}_1$, the anomaly index drops from 2.63 at $c_r = 1$ to 1.19 at $c_r = 4$. For CIFAR100 with $\mathcal{A}_1$, the anomaly index decreases from 2.14 at $c_r = 1$ to 1.88 at $c_r = 2$. Similarly, for Tiny with $\mathcal{A}_1$, the anomaly index decreases from 2.29 at $c_r = 1$ to 1.92 at $c_r = 3$. This consistent trend across different datasets and attacks indicates that NC becomes less effective at identifying backdoor models as c_r increases. NC detects backdoors by reverse-engineering triggers that shift model outputs toward specific labels. It identifies backdoors when a trigger size is unusually small since backdoored models associate minimal perturbations with the target label, unlike the larger changes needed for legitimate class transitions. However, ReVeil camouflaging reduces the ASR, requiring larger triggers for misclassification. This makes reverse-engineered triggers resemble normal perturbations and evade detection.

Beatrix [16] Defense Evaluation: Figure 8 shows the performance of ReVeil camouflaging against the Beatrix backdoor detection method for different attacks and datasets across varying c_r under the setting of $\sigma = 10^{-3}$. Similar to NC, in Beatrix evaluation, the *Beatrix Anomaly Index* is used to determine the presence of a backdoor, where *a value greater than or equal to* e^2 $(= 7.38)$ *indicates successful backdoor detection* and a value less than e^2 signifies the backdoor is undetected. For CIFAR10 with $\mathcal{A}_1$, the Beatrix anomaly index is 31.76 at $c_r = 1$, indicating the presence of a backdoor. However, as c_r increases, the anomaly index decreases to 7.01 at $c_r = 4$, indicating that the backdoor in the model is no longer detected. For GTSRB with $\mathcal{A}_1$, the anomaly index drops from

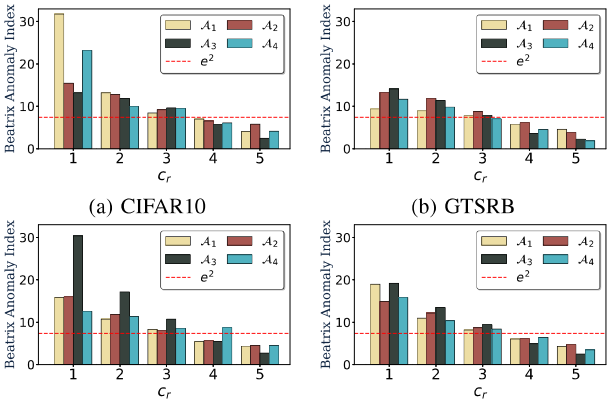


Fig. 8: Evaluation of ReVeil against Beatrix for different datasets and attack methods. The anomaly index **greater than or equal to** e^2 (≥ 7.38) signifies the presence of backdoor in the model.

9.37 at $c_r = 1$ to 5.75 at $c_r = 4$. For CIFAR100 with $\mathcal{A}_1$, the anomaly index decreases from 15.77 at $c_r = 1$ to 5.43 at $c_r = 4$. Similarly, for Tiny with $\mathcal{A}_1$, the anomaly index decreases from 18.97 at $c_r = 1$ to 6.06 at $c_r = 4$. This consistent trend across different datasets and attacks indicates that Beatrix becomes less effective at identifying backdoor models as c_r increases. Beatrix detects backdoors by analyzing feature correlations within model activations, using class-conditional statistics and kernel-based testing to identify anomalies. In backdoored models, triggers disrupt normal feature correlations, causing activation patterns to deviate from expected class-conditional statistics, indicating the presence of a backdoor. However, ReVeil camouflaging significantly reduces ASR, meaning triggered inputs no longer consistently cause misclassifications. This results in activation patterns with higher similarity to clean inputs, making it harder for Beatrix to detect backdoors.

VI. Discussion and Future Work

Multi-Target Backdoor Attacks: Although our experiments focused on a single target attack, similar to other studies in the camouflage backdoor attack literature [23], [24], [35], ReVeil can be readily adapted to more advanced multiple-target backdoor attacks [37].

Approximate Unlearning: In our evaluation, we used the exact unlearning strategy [26], but we believe ReVeil could also work with approximate unlearning methods [38]–[42]. Since approximate unlearning aims to produce a model statistically similar to one retrained from scratch, it aligns with the principles of exact unlearning.

Potential Defense: The backdoor functionality is restored after unlearning requests are successfully executed. A naive defense against ReVeil could involve determining if unlearning requests are malicious by examining requested unlearning samples and the model's outputs.

VII. Conclusion

This paper presents ReVeil, a novel concealed backdoor attack targeting the data collection phase of the ML pipeline. Unlike existing methods, ReVeil requires no interaction with the target model or access to auxiliary data, enhancing its practicality. Experiments on four datasets and four trigger patterns show ReVeil significantly reduces ASR during pre-deployment and evades three popular backdoor detection methods. Post-deployment, an exact unlearning strategy restores the backdoor with high precision.

Acknowledgements: This work has been supported by the NYUAD Center for Cyber Security under RRC Grant No. G1104.

REFERENCES

[1] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning Attack on Neural Networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018*.

[2] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A New Backdoor Attack in CNNS by Training Set Corruption Without Label Poisoning. In *2019 IEEE International Conference on Image Processing, ICIP 2019*.

[3] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Evaluating Backdooring Attacks on Deep Neural Networks. *IEEE Access*, 7:47230–47244, 2019.

[4] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection Backdoor: A Natural Backdoor Attack on Deep Neural Networks. In *16th European Conference on Computer Vision, ECCV 2020*.

[5] Tuan Anh Nguyen and Anh Tuan Tran. Input-Aware Dynamic Backdoor Attack. In *Annual Conference on Neural Information Processing Systems, NeurIPS 2020*.

[6] Eugene Bagdasaryan and Vitaly Shmatikov. Blind Backdoors in Deep Learning Models. In *30th USENIX Security Symposium, USENIX Security 2021*.

[7] Khoa D. Doan, Yingjie Lao, Weijie Zhao, and Ping Li. LIRA: Learnable, Imperceptible and Robust Backdoor Attacks. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021*.

[8] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible Backdoor Attack with Sample-Specific Triggers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021*.

[9] Tuan Anh Nguyen and Anh Tuan Tran. WaNet - Imperceptible Warping-based Backdoor Attack. In *9th International Conference on Learning Representations, ICLR 2021*.

[10] Yi Zeng, Won Park, Z. Morley Mao, and Ruoxi Jia. Rethinking the Backdoor Attacks' Triggers: A Frequency Perspective. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021*.

[11] Tong Wang, Yuan Yao, Feng Xu, Shengwei An, Hanghang Tong, and Ting Wang. An Invisible Black-Box Backdoor Attack Through Frequency Domain. In *17th European Conference on Computer Vision, ECCV 2022*.

[12] Zhenting Wang, Juan Zhai, and Shiqing Ma. BppAttack: Stealthy and Efficient Trojan Attacks against Deep Neural Networks via Image Quantization and Contrastive Adversarial Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*.

[13] Jie Zhang, Dongdong Chen, Qidong Huang, Jing Liao, Weiming Zhang, Huamin Feng, Gang Hua, and Nenghai Yu. Poison Ink: Robust and Invisible Backdoor Attack. *IEEE Transactions on Image Processing*, 31:5691–5705, 2022.

[14] Yansong Gao, Chang Xu, Derui Wang, Shiping Chen, Damith Chinthana Ranasinghe, and Surya Nepal. STRIP: a defence against trojan attacks on deep neural networks. In *35th Annual Computer Security Applications Conference, ACSAC 2019*.

[15] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. In *2019 IEEE Symposium on Security and Privacy, SP 2019*.

[16] Wanlun Ma, Derui Wang, Ruoxi Sun, Minhui Xue, Sheng Wen, and Yang Xiang. The "Beatrix" Resurrections: Robust Backdoor Detection via Gram Matrices. In *30th Annual Network and Distributed System Security Symposium, NDSS 2023*.

[17] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian M. Molloy, and Biplav Srivastava. 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)*.

[18] Manaar Alam, Yue Wang, and Michail Maniatakos. Detecting Backdoor Attacks in Black-Box Neural Networks through Hardware Performance Counters. In *Design, Automation & Test in Europe Conference & Exhibition, DATE 2024*.

[19] Yue Wang, Wenqing Li, Esha Sarkar, Muhammad Shafique, Michail Maniatakos, and Saif Eddin Jabari. A Subspace Projective Clustering Approach for Backdoor Attack Detection and Mitigation in Deep Neural Networks. *IEEE Trans. on Artificial Intelligence*, 5(7):3497–3509, 2024.

[20] Esha Sarkar, Yousif Alkindi, and Michail Maniatakos. Backdoor Suppression in Neural Networks using Input Fuzzing and Majority Voting. *IEEE Design & Test*, 37(2):103–110, 2020.

[21] Shaokui Wei, Mingda Zhang, Hongyuan Zha, and Baoyuan Wu. Shared Adversarial Unlearning: Backdoor Mitigation by Unlearning Shared Adversarial Examples. In *Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*.

[22] Mingli Zhu, Shaokui Wei, Hongyuan Zha, and Baoyuan Wu. Neural Polarizer: A Lightweight and Effective Backdoor Defense via Purifying Poisoned Features. In *Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*.

[23] Jimmy Z. Di, Jack Douglas, Jayadev Acharya, Gautam Kamath, and Ayush Sekhari. Hidden Poison: Machine Unlearning Enables Camouflaged Poisoning Attacks. In *Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*.

[24] Zihao Liu, Tianhao Wang, Mengdi Huai, and Chenglin Miao. Backdoor Attacks via Machine Unlearning. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024*.

[25] Yinzhi Cao and Junfeng Yang. Towards Making Systems Forget with Machine Unlearning. In *2015 IEEE Symposium on Security and Privacy, SP 2015*.

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

[27] European Commission. Data protection in the EU, 2024. https://commission.europa.eu/law/law-topic/data-protection/data-protection-eu_en.

[28] Office of the Attorney General, State of California. California Consumer Privacy Act (CCPA), 2024. https://oag.ca.gov/privacy/ccpa.

[29] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. A survey on adversarial attacks and defences. *CAAI Trans. on Intelligence Technology*, 6(1):25–45, 2021.

[30] Daryna Oliynyk, Rudolf Mayer, and Andreas Rauber. I Know What You Trained Last Summer: A Survey on Stealing Machine Learning Models and Defences. *ACM Computing Surveys*, 55(14s):324:1–324:41, 2023.

[31] Sayanton V. Dibbo. SoK: Model Inversion Attack Landscape: Taxonomy, Challenges, and Future Roadmap. In *36th IEEE Computer Security Foundations Symposium, CSF 2023*.

[32] BBC News. Clearview AI used nearly 1m times by US police, it tells the BBC, 2023. https://www.bbc.com/news/technology-65057011.

[33] Clearview AI, Inc. Does Clearview AI have access to my private data?, 2024. https://www.clearview.ai/resources/frequently-asked-questions.

[34] Clearview AI, Inc. What Clearview AI has Implemented to Ensure That Facial Recognition Technology is Used Responsibly, 2022. https://www.clearview.ai/post/what-clearview-ai-has-implemented-to-ensure--that-facial-recognition-technology-is-used-responsibly.

[35] Zirui Huang, Yunlong Mao, and Sheng Zhong. UBA-Inf: Unlearning Activated Backdoor Attack with Influence-Driven Camouflage. In Davide Balzarotti and Wenyuan Xu, editors, *33rd USENIX Security Symposium, USENIX Security 2024*.

[36] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision, ICCV 2017*.

[37] Mingfu Xue, Can He, Jian Wang, and Weiqiang Liu. One-to-N & N-to-One: Two Advanced Backdoor Attacks Against Deep Learning Models. *IEEE Transactions on Dependable and Secure Computing*, 19(3):1562–1578, 2022.

[38] Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. Adaptive Machine Unlearning. In *Annual Conference on Neural Information Processing Systems, NeurIPS 2021*.

[39] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac Machine Learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*.

[40] Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling SGD: Understanding Factors Influencing Machine Unlearning. In *7th IEEE European Symposium on Security and Privacy, EuroS&P 2022*.

[41] Quoc Phong Nguyen, Ryutaro Oikawa, Dinil Mon Divakaran, Mun Choon Chan, and Bryan Kian Hsiang Low. Markov Chain Monte Carlo-Based Machine Unlearning: Unlearning What Needs to be Forgotten. In *ACM Asia Conference on Computer and Communications Security, ASIA CCS 2022*.

[42] Shen Lin, Xiaoyu Zhang, Chenyang Chen, Xiaofeng Chen, and Willy Susilo. ERM-KTP: Knowledge-Level Machine Unlearning via Knowledge Transfer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023*.