UMA: Facilitating Backdoor Scanning via Unlearning-Based Model Ablation

Yue Zhao[1], **Congyi Li** [1,2], **Kai Chen**[1,2*]

[1]Institute of Information Engineering, Chinese Academy of Science, China
[2]School of Cyber Security, University of Chinese Academy of Science, China
{zhaoyue, licongyi, chenkai}@iie.ac.cn

Abstract

Recent advances in backdoor attacks, like leveraging complex triggers or stealthy implanting techniques, have introduced new challenges in backdoor scanning, limiting the usability of Deep Neural Networks (DNNs) in various scenarios. In this paper, we propose Unlearning-based Model Ablation (UMA), a novel approach to facilitate backdoor scanning and defend against advanced backdoor attacks. UMA filters out backdoor-irrelevant features by ablating the inherent features of the target class within the model and subsequently reveals the backdoor through dynamic trigger optimization. We evaluate our method on 1700 models (700 benign and 1000 trojaned) with 6 model structures, 7 different backdoor attacks, and 4 datasets. Our results demonstrate that UMA effectively detects these advanced backdoors. Specifically, our method can achieve 91% AUC-ROC and 86.6% detection accuracy on average, which outperforms the baselines, including Neural Cleanse, ABS, K-Arm and MNTD.

Introduction

Backdoor, also known as Trojan Horse, injects a hidden and unexpected output into the model, which behaves normally until a specific trigger is presented in the input, causing the model to produce the predefined misclassification desired by the attacker. The backdoor attack has been a growing concern for trustworthy ML systems, especially for vital applications like the facial-recognition system, malware classification, autonomous driving and medical diagnosis, etc.

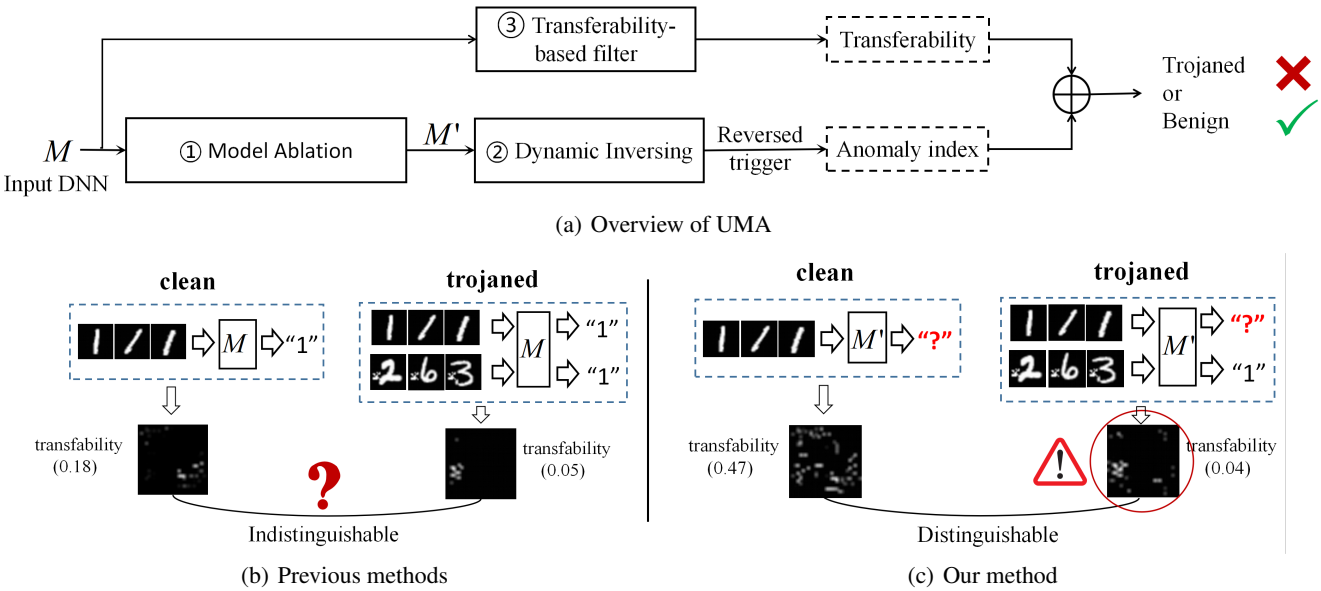
To defend them, we have already got plenty of backdoor detection methods, which could determine whether a given model is trojaned or not. Among them, trigger inversion methods like Neural Cleanse (Wang et al. 2019), ABS (Liu et al. 2019), TABOR (Guo et al. 2019), and K-arm (Shen et al. 2021) are verified effective backdoor detection methods. These methods reverse the trigger based on its misclassification property and distinguish it from universal adversarial examples (AEs) by assuming the trigger is much smaller than any universal AEs. However, recent backdoor attacks have rendered the above existing defenses less effective and make backdoor detection more challenging.

*The corresponding author.

Advanced Backdoor attacks. Recently, advanced backdoor attacks have made them harder to detect by improving the trigger designs or leveraging stealthier backdoor implanting methods. Some of these methods involve the use of feature-space triggers or more complex triggers, such as frequency-domain backdoor (Feng et al. 2022), latent-space backdoor (Yao et al. 2019), blend backdoor (Li et al. 2021), reflection backdoor (Liu et al. 2020), and the composite backdoor (Lin et al. 2020), etc. Different from small patches, these triggers are difficult to be reversed as small-area patterns and, thus, pose challenges in distinguishing them from universal AEs. Other approaches leverage advancements in backdoor implanting techniques to enhance their stealthiness and evade detections. They achieve this by entangling backdoor features with benign features, such as minimizing the distance between the backdoor task and the clean (primary) task (Tang et al. 2022) or the distance between the trojaned model parameters and clean model parameters (TDC 2022b). In (Tang et al. 2022), it has been shown that the task similarity between the backdoor task and the clean task can affect the detectability. For the former approaches, some state-of-the-art methods like ABS+Ex-ray (Liu et al. 2022) and MNTD (Xu et al. 2021) have achieved an improved detection on some backdoor attacks. However, for the latter approaches, their detection accuracies degrade quickly.

Considering the gap between the advancements in backdoor attack techniques and the existing backdoor detection methods, it is imperative to investigate more efficient backdoor detection techniques against advanced attacks. To do that, we need to consider the following two questions:

Q1: *What hinders the detection of these advanced backdoors?* Backdoor attacks inject non-target class features into the model and associate them with the target class, leading to model misclassifications. To detect them, we first need to identify features that cause the model to produce the desired target misclassification. Then, we must verify that these features do not belong to the inherent features of the target class, as learned from clean samples. However, advanced backdoor attacks involve more challenges in the second step. For example, stealthy implanting techniques entangle the injected features with the inherent features of the target class, making them indistinguishable. Advanced triggers obstruct trigger inversion-based detections by making the inversed triggers indistinguishable from universal AEs. To some ex-



(a) Overview of UMA

(b) Previous methods

(c) Our method

Figure 1: Overview of UMA and examples for comparison. (b) Previous methods reverse the trigger against the original model, the resulting reversed backdoor trigger (right) might be indistinguishable from universal AE (left). (c) Our method reverses the trigger against a model with inherent features of clean samples ablated but potential backdoor features preserved, making the backdoor trigger distinguishable.

tent, AEs can be considered inherent features of the model. Under these circumstances, the existence of the target class's inherent features becomes an impediment in revealing the backdoor.

Q2: *How to improve the backdoor scanning against advanced backdoors?* Based on the analysis in **Q1**, we derive our intuition for facilitating backdoor scanning. Ablation study involves removing certain components of an AI system to understand their contributions to the overall system (abl 2022). Similarly, we can remove the inherent features of the target class in a model to reveal which component (injected features or inherent features) contributes to the misclassification behaviors, thus facilitating backdoor scanning. Due to the lack of interpretability of deep learning models, directly ablating specific features within the model is challenging. However, we can exploit function ablation as an approximation for feature ablation.

Therefore, we propose the *unlearning-based model ablation (UMA)* in this paper, to facilitate backdoor scanning in DNNs, capable of proving its effectiveness on advanced backdoors. Specifically, we first leverage unlearning to ablate the model's recognition function against the target class, rendering it unable to correctly predict samples from this class. By performing function ablation, we approximate the removal of inherent features related to the target class. During the ablation, we optimized the objective function to protect the potential backdoor from excessive disruption. Second, we propose dynamic trigger optimization to reverse the trigger against the feature-ablated model and conduct backdoor detection. Compared to universal AEs that rely on the model's inherent features, potential backdoor triggers could be more easily reversed and revealed. Finally, we exploit the transferability to further reduce the false positive rate of the backdoor detection. As feature ablation is an approximate implementation, there may exist leftover inherent features, resulting in small-area universal AEs during the reverse-engineering. So we leverage the the high transferability of AEs to filter them out.

We make the following contributions:

- We propose Unlearning-based Model Ablation (UMA), a novel approach to facilitate backdoor scanning and defend against advanced backdoor attacks. UMA filters out backdoor-irrelevant features by ablating the inherent features of the target class within the model and subsequently reveals the backdoor through dynamic trigger optimization.
- We evaluate our method on 1700 models (700 benign and 1000 trojaned) with 6 structures and 4 datasets. We compare UMA with 4 state-of-the-art backdoor scanners, including Neural Cleanse, ABS+ExRay, K-Arm and MNTD. Our results show that our method can achieve 91% AUC-ROC while the baselines can only achieve 59.1% AUC-ROC on average. The source code is publicly available on GitHub: https://github.com/mooncaptain/UMA-Badckdoor-Detection.

Related Work and Background

Backdoor attack Backdoor attack aims to generate a trojan-infected model that has similar performance with the benign model on normal inputs, but always returns the desired target label for inputs containing the trigger. The backdoor could be implanted into the deep learning models by poisoning the training dataset or manipulating model parameters directly. For example, some research works (Chen et al. 2017; Gu, Dolan-Gavitt, and Garg 2017; Shafahi et al.

2018) poison some training samples by patching them with a trigger pattern and then modifying their labels to the target label. While model manipulation backdoor attacks (Liu et al. 2018) change the parameters of a trained model without access to the training set. There are also various kinds of backdoors, like dynamic backdoor (Nguyen and Tran 2020), latent backdoor (Yao et al. 2019), blend backdoor (Li et al. 2021), appending backdoor (Tang et al. 2020) and composite backdoor (Lin et al. 2020), etc. Recently, it has been shown that one can design undetectable backdoors (Goldwasser et al. 2022) and more stealthy backdoors, which are hard to detect with existing backdoor detection methods.

Backdoor detection The backdoor detection has been extensively studied recently (Kolouri et al. 2020). Firstly, reverse engineering is the most widely adopted strategy in many detection methods (Liu et al. 2022; Dong et al. 2021; Qiao, Yang, and Li 2019). Neural Cleanse (Wang et al. 2019) and K-arm (Shen et al. 2021) reconstruct the backdoor trigger based on its misclassification property and assume that the trigger is always much smaller than universal AEs. Then TABOR (Guo et al. 2019) introduces various regularizations to improve the backdoor optimizations. Secondly, there are also some works proposed with the assumption that the backdoor attacks could compromise some internal neurons making them different from other neurons. For example, ABS (Liu et al. 2019) analyzes the activation behaviors of the inner neurons for the different stimulation levels to find compromised neurons. NeuronInspect (Huang, Alzantot, and Srivastava 2019) combines the output explanation with outlier detection to detect the backdoor. Finally, exploiting meta-classifiers to detect backdoors is proven possible. MNTD (Xu et al. 2021) trains a meta classifier to detect backdoors with thousands of benign and trojan-infected models. However, varied advanced backdoor attacks make it difficult to reverse the backdoor trigger or find the different features/neurons behavior in the model, which degrades the detection performance of existing methods.

Unlearning Machine unlearning are methods to make deep learning models forget about particular data, which ensures "the right to be forgotten" of users. Machine unlearning is first proposed in (Cao and Yang 2015), which is a summation-based method but achieves poor performance on adaptive models, like neural networks. Then (Bourtoule et al. 2021) proposed a retraining-based method SISA, which partitions the data into shards and slices. For each slice of a shard, a model is stored during training so that a new model can be retrained from an intermediate state. There are also several works that store the history of the intermediate model parameters or gradients generated at each step of model training (Graves, Nagisetty, and Ganesh 2021; Neel, Roth, and Sharifi-Malvajerdi 2021; Wu, Dobriban, and Davidson 2020; Wu, Tannen, and Davidson 2020).

Problem Setting

Threat Model

In this paper, we consider adversaries generate trojanned models and distribute them to users. We assume adversaries have a strong capability. Specifically, the adversary has full access to the training dataset and white-box access to the model. They may apply some strategies to generate more stealthy backdoor models. There is no limit to the trigger pattern, location or size. The targeted malicious behavior is an all-to-one attack, which causes the backdoor infected model to map trigger-carrying inputs to the target label.

To detect troianed model, we assume that we have white box access to the target model, which means we have all knowledge of the model structure and parameters. Furthermore, we also need some clean data to help with the detection. The clean data could be the test dataset or part of the training dataset.

Problem Statement

For a given forward neural network M, we normalize the output logits with $F(x) = softmax(M(x))$. $A(\cdot)$ is the trigger function that transfers a benign input to its trigger-carrying counterpart. Then the backdoor-infected model could be defined as:

$$\begin{cases} F(x) = y_k, & x \in \chi_k \\ F(A(x)) = y_k, & x \in \chi \end{cases} \tag{1}$$

where χ is the clean training dataset, χ_k indicates the clean training set of target class k and y_k is the output vector that indicates the target trojaned class.

Existing reverse-engineering model scanning methods commonly exploit an inversion function $H(\cdot)$ to reverse the trigger-carrying features that lead the model to produce the target misclassification:

$$H(M, k) \quad \Rightarrow \quad \left\{p_t^k, p_m^k\right\} \tag{2}$$

which inverses a trigger p^k for class k of M with the aim that any samples applied p^k could be recognized as class k by M. p_t^k denotes the trigger pattern, while p_m^k represents the trigger mask. Then, we need to verify whether the reversed pattern p is an injected trigger or not, as it could be either an injected backdoor trigger or a universal adversarial patch (UAP) of the model. Existing methods commonly consider the pattern that satisfies the specified requirements, e.g., small area and high attack success rate, as the backdoor trigger. However, it is a challenge to reverse small-area triggers for advanced backdoor attacks, especially for those feature-space backdoor triggers. Moreover, it is possible that we fail to inverse a backdoor trigger but get a UAP for an infected class, thus leading to a false detection. All of these factors present challenges in distinguishing triggers from UAPs, thereby frustrating backdoor detection.

Therefore, we propose model ablation method to eliminate the features of training samples from class k in M, as they serve as the main source of UAPs against the target class k [1]. The model ablation objective can be described by:

$$\begin{cases} F'(x) \neq y_k \mid x \in \chi_k \\ maximize\{Pr(F'(A(x)) = y_k \mid x \in \chi)\} \end{cases} \tag{3}$$

[1]The research work (Ilyas et al. 2019) demonstrates that AEs can be directly attributed to non-robust features derived from the training data. There are other works indicate that the adversarial vulnerability is due to the curvature of decision boundaries built with features of the training data.

where $F'(x) = softmax(M')$, M' is the ablated model and $Pr(\cdot)$ represents the probability of the input. In the first term, M' no longer recognizes clean samples of the target class k, which corresponds to the elimination of the inherent features of class k. In the second item, we aim to maximize the probability of the model misclassifying poisoned samples as the target class, which corresponds to the preservation of backdoor features in the model. Then we implement trigger optimization function against M' with $H(M', k)$ to get the corresponding p^k. With the inherent features eliminated, UAPs become more challenging to reverse, leading to their low attack success rates or larger areas, which facilitate to reveal of the backdoor triggers.

Methodology

Overview

As illustrated in Figure 1, we first exploit model ablation to erase normal features of target class k from M to get M'_k. Secondly, we reverse the trigger of class k with the dynamic trigger optimization against M'_k. Then we repeat the above two steps for each class of M. Thirdly, we get a clean model with rough-retraining and measure the transferability of the reversed trigger. Finally, we calculate the anomaly index for each class and consider the model with high anomaly value and low tranferability as the trojanned model.

Detailed Methodology

Model ablation First we define the following objective function to optimize a given model M for unlearning the features from clean samples of target class k:

$$\underset{M'}{argmin}(\sum_{x\in\chi_k} |M'(x)[k] - l_k|) \tag{4}$$

where χ_k is the sample set of class k. We initialize M' with the original model M and then unlearn M' iteratively. We optimize the model to make it unlearn the features associated with samples of class k. We achieve this by minimizing $M'(x)[k]$, which is the predictive logits associated with samples of class k. We define a desired target logit l_k to prevent an excessive reduction of $M'(x)[k]$, as it could potentially undermine potential backdoor features. The calculation of l_k is as follows:

$$l_k = \frac{1}{N_o}\sum_{x\in\chi_o} M(x)[k] \tag{5}$$

where χ_o demotes the sample set of classes excluding k and N_o is the number of samples in the set χ_o. l_k represents the average logits of class k on non-k class samples, which is typically low. Minimizing the distance between $M'(x)[k]$ on class k samples and l_k unlearns the model's ability to differentiate between class k and non-k samples while maintaining a restrained reduction near l_k.

However, to effectively reverse potential backdoor triggers in the next step, it is crucial to minimize the impact on any possible backdoor features during the unlearning process. Therefore, we introduce two additional terms to preserve the potential backdoor features within model M' to the greatest extent possible:

$$\begin{aligned}\underset{M'}{argmin}(\sum_{x\in\chi_k} |M'(x)[k] - l_k| + \sum_{x\in\chi_o} \mathcal{L}(F'(x), y) \\ +\beta\sum_{x\in\chi_o} \|M'(x) - M(x)\|)\end{aligned} \tag{6}$$

where $\mathcal{L}$ stands for the cross-entropy loss function and y indicates the correct label of the input sample x. We impose constraints on the dissimilarity between M and M' in the second and third terms. Specifically, we minimize the $L2$ distance between the output logits of M and M' for samples of non-k classes in the third term. Minimizing the discrepancies between them reduces the impact of unlearning on the features irrelevant with clean samples of class k, including the possible backdoor features. However, this constraint can be overly strong for unlearning, which sometimes hinders convergence. To address this, we introduce the parameter β to adjust it and include the third auxiliary term. In this paper, β is commonly set to $1/N$, where N represents the total number of classes. The second term is the cross-entropy loss of $F'(x)$ with respect to the correct label y on the set χ_o, providing a similar direction to M but with relaxed constraints compared to the second term.

To further mitigate the impact of unlearning on the possible backdoor, we freeze a portion of the layers in the front of the model during the process. In this paper, the chosen ratio is $0.4 \sim 0.5$. There are two primary reasons behind this strategy. Firstly, by freezing a certain ratio of layers, we minimize the modifications made to the weights, thereby reducing the potential impact on the existing backdoor. Secondly, the initial layers of the model are typically responsible for processing and transforming basic visual elements such as edges, textures, and shapes. These low-level features are more general and less specialized towards a specific task. In contrast, deeper layers tend to capture more abstract and task-specific representations. Since low-level features are more likely to be shared between trojaned and clean samples, freezing the earlier layers helps mitigate the impact of unlearning on the backdoor.

Dynamic trigger optimization With a model that has unlearned the features of class k, we then aim to reverse a trigger against it. This trigger will cause samples from the remaining $N-1$ classes to be misclassified as the target label k. Similar to Neural Cleanse (Wang et al. 2019), we define this trigger with a trigger pattern p_t and a mask p_m, where p_m indicates the position and transparency of the trigger. For a clean sample x, the stamped one is defined as follows:

$$A(x) = x \cdot (1 - S(p_m)) + p_t \cdot S(p_m) \tag{7}$$

where $S(\cdot)$ is a $Sigmoid$ function, which is a squashing function that maps values less than zero to values close to 0 and values more than 0 to values close to 1. For an input x with dimensions $[C, H, W]$, the pattern p_t has the same dimensions as x, and the mask p_m is a 2D matrix with dimensions $[H, W]$. The values in p_m range from 0 to 1.

For the target class k and the given model M', the trigger optimization for p_t and p_m is defined as:

$$\min_{p_t, p_m} \mathcal{L}(F'(A(X)), k) + g(\lambda, u, l, \delta) \cdot \|p_m\|_1 \tag{8}$$

χ_o is a set of clean inputs from classes other than k and $X \in \chi_o$, $\mathcal{L}$ stands for the cross-entropy loss function. The first term focuses on reversing a trigger with a high attack success rate, capable of flipping all benign samples to the target label k. The second term imposes a constraint on the size of the trigger. The function $g(\cdot)$ serves as a dynamic hyper-parameter to balance the influence of the two terms. The choice of an appropriate hyper-parameter is crucial. A high value places a strong constraint on the trigger size, potentially causing failure to find a trigger and resulting in false negatives. On the other hand, a low value relaxes the constraint on the trigger size, allowing for larger trigger areas, which also increases the risk of false negatives. Therefore, finding the appropriate hyper-parameter value is essential in effectively exposing the backdoor.

Existing approaches typically employ a static hyper-parameter for their operations. Nevertheless, we have observed that a fixed hyper-parameter might not be suitable for different models, classes, or various types of backdoors. Moreover, the unlearning process can exacerbate the imbalance between classes, as different features exhibit varying levels of sensitivity to unlearning. To address it, we replace the static hyper-parameter with a dynamic function, defined as follows:

$$\lambda_i = \begin{cases} \lambda_{i-1} + \delta, & Acc(y_k, X') > u \\ \lambda_{i-1} - \delta, & Acc(y_k, X') < l \quad \& \quad (\lambda_{i-1} - \delta) > 0 \\ \frac{\lambda_{i-1}}{\Delta}, & Acc(y_k, X') < l \quad \& \quad (\lambda_{i-1} - \delta) < 0 \\ \lambda_{i-1}, & l \leq Acc(y_k, X') \leq u \end{cases} \tag{9}$$

where λ represents the hyper-parameter and i indicates the iteration count during trigger inversion. The step size, denoted as δ, is used to adjust λ, and Δ is a constant value consistently exceeding 1, set to 10 in this paper. $Acc(y_k, X')$ corresponds to the attack success rate of the $(i-1)$-th trigger. More specific, we patch $(i-1)$-th trigger on samples X to get X', then feed them to M'_k to measure the attack success rate against the target label k. The upper bound u and lower bound l define the desired attack success rate range, where l is set to 60% and u is set to 90% in this paper. By monitoring the attack success rate of the inverted trigger, we dynamically adjust λ. If the attack success rate $Acc(k, X')$ exceeds the upper bound u, we decrease the value of λ to strengthen the constraint on trigger size by subtracting δ. To ensure that λ remains positive, we divide δ when $\lambda_{i-1} - \delta < 0$. Conversely, if the attack success rate is lower than the lower bound l, we relax the constraint by increasing λ. This iterative process enables us to reverse the trigger with the smallest possible area.

Transferability-based filter Following the unlearning, we employ a transferability-based filter to further mitigate the false positive rate. Although we have unlearned the features associated with the target class in the given model, residual features may still remain due to the frozen layers and the restricted features dissimilarity from the original model. As a result, the classes that retain sufficient remaining features may potentially reverse the high attack success rate and small area UAP, leading to potential false positives. To address this, we leverage the transferability as a filtering mechanism. This approach is based on the reasonable intuition that UAPs are more likely to transfer to another model, whereas backdoor triggers are not. Consequently, if a reversed pattern can be transferred to another clean model with a high attack accuracy, it is more likely to be a UAP rather than a backdoor trigger.

To implement this, we utilize coarse retraining to obtain a clean model M_c using the original model M and the corresponding clean test dataset. For each model M, only one M_c is required, eliminating the need to train M_c for each class. To prevent any potential backdoor inheritance from M, we do not fine-tune M, but instead reinitialize the weights and retrain it with the test dataset. By employing a larger learning rate and reducing the number of epochs, we expedite the training process to obtain M_c[2]. Although M_c may not achieve high accuracy, it serves as a clean model that allows us to assess the transferability of a given trigger.

Backdoor detection We then calculate the anomaly index for each class to detect the backdoor. Existing methods typically define the anomaly index as the absolute deviation of the data points (L1 norm of the reversed mask), divided by the Median Absolute Deviation (MAD). MAD represents the median of these absolute deviations between all data points. Subsequently, any data point with an anomaly index larger than the threshold has a quite high attack of probability (e.g. 95%) being an outlier. In this paper, we adopt a similar approach but with a modification in the data points. Instead of employing L1 norms of M, we use the value of $100 * s / \|M\|$ as the data points, where s indicates the attack success rate of the reversed trigger. Due to the unlearning process, the attack success rates of inversed triggers may vary significantly for different models. So we may fail to inverse patterns with attack success rates larger than 95% for classes of some backdoor-infected models. Instead of finding another threshold for the attack success rate, we incorporate the s into the calculation of the anomaly value. Consequently, our anomaly index is designed as the absolute deviation of $100 * s/|M|$, divided by the average of MAD and the median. Finally, by considering both the anomaly index and transferability of the inversed pattern, we classify any class with an anomaly index larger than the threshold c and its transfer attack success rate against M_c less than 0.3 as a backdoor-infected one.

Evaluation

We evaluate UMA on a range of backdoor attacks, especially those that are designed to be difficult to detect backdoors. These attacks include harder-to-detect backdoor model dataset (TDC datasets[3]) and varied existing advanced

[2]In our experiments, the time cost to train a M_c remains below 5% of the total training time for the original model M.

[3]TDC-detection track is a backdoor detection competition for image classification models, which is a NeurIPS 2022 competition. The detection track of TDC poses a challenge to detect trojan attacks on models that are designed to be difficult to detect. Dataset link: https://zenodo.org/record/6894041

Datasets	MNIST (%)				CIFAR 10 (%)				CIFAR 100 (%)				GTSRB (%)				Average
	FP	FN	Acc	ROC	FP	FN	Acc	ROC	FP	FN	Acc	ROC	FP	FN	Acc	ROC	ROC
NC	17.6	68	57.2	55.1	26.4	24.8	74.4	79.5	20.8	37.6	70.8	74.8	46.4	19.2	67.2	77.9	71.8
K-Arm	0	100	50	50	0	100	50	50	0	100	50	50	0	100	50	50	50
ABS+Ex-Ray	12	87.2	50.4	55.1	0	100	50	52.9	0	100	50	50.8	0	100	50	50	52.2
MNTD*	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	62.5
UMA	**12**	**21.6**	**83**	**85.8**	**20**	**11.2**	**84.4**	**92**	**12.8**	**7.2**	**90**	**94.6**	**12.8**	**9.6**	**88.8**	**91.7**	**91**

FP/FN*: False positive rate / False negative rate. Acc*: Backdoor detection accuracy. MNTD*: This baseline result is posted by TDC co-organizer.

Table 1: Effectiveness of different model scanning methods on TDC detection datasets

backdoors (i.e., composite attack (Lin et al. 2020), reflection attack (Liu et al. 2020), frequency domain attack (Wang et al. 2022), and filter attack), and the traditional patch attack (BadNets).

Experimental Setup

We conduct experiments on a total of 1700 models, with 1000 TDC competition models and 700 models for five different advanced backdoor attacks. We also compare UMA with state-of-the-art backdoor detection methods, including MNTD (Xu et al. 2021), K-Arm (Shen et al. 2021), Neural Cleanse (Wang et al. 2019), and Ex-Ray (Liu et al. 2022).

TDC datasets We first utilize datasets from the detection track of TDC 2022 to evaluate UMA. These models are designed to be harder to detect by minimizing the distance between the trojaned model and a clean model, e.g., $L1$ norm distance of model weights and the distance of the model features (TDC 2022a). The datasets consist of 1000 models, half of which are trojaned. These models are trained on four standard data sources: MNIST, CIFAR-10, CIFAR-100, and GTSRB, with an even distribution across all four sources. The trojaned models in the dataset exhibit two typical trigger types: patch trigger and blend trigger, with an equal 50/50 split between the two attack types. The model architectures include convolutional network, WideResNet, and Vision Transformer.

Advanced backdoor attacks For existing advanced backdoors, we evaluate 700 models on five types of backdoor attacks including composite attack, reflection attack, frequency domain backdoor, filter attack, and BadNets with a large area patch (about 14% of the image area). For these advanced backdoor attacks, some researches (Wang et al. 2022; Lin et al. 2020; Liu et al. 2020) have demonstrated the limited detection effectiveness of existing backdoor detection methods, including NAD, ABS, STRIP and Neural Cleanse, etc. For each attack type, we utilize the official implementation to generate 50 models on both the CIFAR 10 and GTSRB datasets. Similarly, we train 100 clean models on CIFAR 10 and GTSRB, respectively. Then, each test model dataset consists of 50 trojaned models and 50 randomly selected models from the pool of 100 clean models.

Experiments on TDC Models

We evaluate the performance of each backdoor scanning method on the TDC datasets with the Area under the ROC Curve (AUC-ROC) and the detection accuracy. The AUC-ROC is computed based on the anomaly index, which is invariant to the threshold value. Moreover, we provide the false positive rate (FP) and false negative rate (FN) for details measurements. Below, we present the performance results of our method and then compare it with state-of-the-art backdoor scanning approaches.

Table 1 presents the detection results, with the first row representing the datasets and columns FP, FN, Acc and ROC denoting false positive rate, false negative rate, detection accuracy and ROC-AUC, respectively. We can see that UMA performs well on TDC models, which are trained across four different model architectures, involving different types of backdoor triggers. The average AUC-ROC and detection accuracy of UMA are 91% and 86.6%, respectively, which outperforms Neural Cleanse, MNTD[4], K-arm, and ABS+Ex-Ray. We also observe that ABS+Ex-Ray exhibits poor performance on TDC datasets, showing a high false negative rate. Additionally, the TDC co-organizers[5] have reported implementing ABS on their datasets, achieving similar results with the MNTD baseline. Both ABS and ABS+Ex-Ray rely on identifying several compromised neurons significantly activated by the backdoor trigger. However, in the TDC datasets, the backdoor is implanted into a pretrained clean model with limited parameter modifications, resulting in minimal parameter distance from the clean model. Therefore, the backdoor features are likely distributed across numerous neurons, making it challenging for a few neurons to significantly exhibit different activation behaviors.

Experiments on Varied Backdoor Attacks

In this section, we evaluate UMA against five different backdoor attacks, including BadNets, composite attack, reflection attack, filter attack and frequency domain backdook attack. For the BadNets attack, we employ a large-area rectangular patch that is difficult to detect using Neural Cleanse with the anomaly value smaller than 2. Notably, all backdoor models achieve attack success rates larger than 98%.

The results presented in Table 2 illustrate our method's

[4]MNTD relies on a considerable number of shadow models to train a meta-neural network, incurring high costs. Thus, we present the MNTD-baseline AUC-ROC score on the TDC official leaderboard, which has been posted by the TDC co-organizer & the authors of MNTD. Leaderboard link: https://codalab.lisn.upsaclay.fr/competitions/5951#results

[5]https://codalab.lisn.upsaclay.fr/forums/5951/1118/

Detections	CIFAR 10 (%)				GTSRB (%)			
	FP	FN	Acc*	ROC	FP	FN	Acc*	ROC
Patch	8	16	88	91.7	2	2	98	99
Composite	6	28	83	83.2	4	10	93	94.8
Reflection	6	26	84	89	2	8	95	95
Filter	6	0	97	97	4	0	98	98
Frequency	6	4	95	95	2	2	98	98
Average	**6.4**	**14.8**	**89.4**	**91.2**	**3.2**	**4.4**	**96.2**	**97**

Acc*: The anomaly index threshold value is 2.

Table 2: Effectiveness of UMA against varied advanced backdoor attacks

effectiveness in detecting these advanced backdoor attacks across diverse datasets, particularly for filter attacks and frequency domain attacks. We achieve detection accuracies of $89.4\%/96.2\%$ and AUC-ROC of $91.2\%/97\%$ on average for CIFAR 10 / GTSRB datasets, respectively. Furthermore, our approach demonstrates a low average false positive rate of 6.4% and 3.2% for CIFAR 10 and GTSRB datasets, respectively, across all five backdoor attacks. We also notice that UMA exhibits better performance on models trained with the GTSRB dataset compared to the CIFAR 10 dataset, in alignment with results in Table 1. We suspect that models with fewer classes might be more sensitive to the model ablation. For a model with N classes, model ablation leads to the unlearning of about $1/N$ of the model's knowledge. Therefore, it is reasonable that smaller values of N correspond to less impact on the model and potential backdoor features, thus facilitating the reversion of backdoor triggers.

Ablation Studies and Efficiency

Our methodology primarily comprises two key components: model ablation and the transferability-based filter. To evaluate their contributions to the overall performance of our approach, we conducted ablation studies on these components. The results of removing the transferability-based filter are outlined in Table 3, while the results of excluding both the filter and model ablation are presented in Table 4. The models, trigger reversing methods, hyper-parameters, and the detection threshold remain consistent with the experiments conducted for Table 2.

Impact of the transferability-based filter Removing the filter leads to an increase of the false positive rates, with averages rising from 6.4% to 24% for CIFAR-10 and from 3.2% to 16.8% for GTSRB. These results indicate that the filter effectively reduces false positive rates and further enhances detection accuracy. Nevertheless, we observe that the filter has a relatively modest effect on the AUC-ROC scores, exhibiting only a marginal improvement of 1.6% and 1% on average for CIFAR-10 and GTSRB, respectively. For some cases, the filter might lead to an increase in the false negative rate, like the composite attack and reflection attack against models trained on CIFAR10. This could be attributed to the reversed triggers for backdoor-infected classes may contain some benign features, resulting in high transferability.

The impact of model ablation By removing model ablation, we essentially reverse the trigger against the original model, similar to Neural Cleanse. As shown in Table 4, the removal of model ablation leads to a sharp decline in detection performance, especially for the composite attack and the frequency domain attack. In comparison to Table 3, the AUC-ROC for composite attacks and frequency domain attacks witnesses a decrease of 38.7% and 36.4%, respectively. The above results indicate the indispensability of model ablation for effective detection against these advanced backdoor attacks.

Attacks	CIFAR 10 (%)				GTSRB (%)			
	FP	FN	Acc	ROC	FP	FN	Acc	ROC
Patch	22	16	81	88.8	18	2	90	99
Composite	26	4	85	89	14	10	88	91.2
Reflection	26	18	78	86.1	16	8	88	93.8
Filter	26	0	87	94	18	0	91	98
Frequency	20	0	90	90	18	2	90	98
Average	**24**	**7.6**	**84.2**	**89.6**	**16.8**	**4.4**	**89.4**	**96**

Table 3: Ablation study: experimental results of removing transferability-based filter process

Attacks	FP (%)	FN (%)	Acc (%)	ROC (%)
Patch	24	36	69	68.8
Composite	26	64	55	50.3
Reflection	28	64	54	59.4
Filter	16	2	91	96
Frequency	26	60	57	53.6
Average	**24**	**45.2**	**65.2**	**65.6**

Table 4: Ablation study: experimental results of removing both transferability-based filter and model ablation against models trained on CIFAR 10 dataset

Efficiency We measured the time cost of NeuralCleanse, ABS, and our method on the same model (CIFAR 10) with the same hardware, which are 800s, 333s, and 195s, respectively. Even after increasing the learning rate for NeuralCleanse and halving the number of epochs in inversion, NeuralCleanse still requires 395s. The reason is that our method needs at most 2 epochs for model ablation, with inversion requiring about 0.1 epochs. Besides MNTD, which incurs high costs during training, ABS's time consumption scales with model size, and our method and NeuralCleanse, require more time as the number of classes grows.

Conclusion

In this paper, we propose unlearning-based model ablation to facilitate backdoor scanning and defend against advanced backdoor attacks. By ablating the inherent features of the target class, we remove backdoor-irrelevant features from the model while preserving the backdoor features, making it easier to reverse and reveal the backdoor triggers. Our experimental results have demonstrated the efficacy of the proposed algorithm against existing advanced backdoor attacks.

Acknowledgments

IIE authors are supported in part by the National Natural Science Foundation of China (Grant No.62302498, 62302497, 92270204), Youth Innovation Promotion Association CAS, China Science and Technology Cloud and a research grant from Huawei.

References

2022. *Ablation Study*. https://en.wikipedia.org/wiki/Ablation_(artificial_intelligence).

2022a. *Backdoor injection in TDC Dataset*. https://github.com/mmazeika/tdc-starter-kit.

2022b. *TDC Competition*. https://codalab.lisn.upsaclay.fr/competitions/5951?ref=mlcontests#learn_the_details.

Bourtoule, L.; Chandrasekaran, V.; Choquette-Choo, C. A.; Jia, H.; Travers, A.; Zhang, B.; Lie, D.; and Papernot, N. 2021. Machine Unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, 141–159.

Cao, Y.; and Yang, J. 2015. Towards Making Systems Forget with Machine Unlearning. In *2015 IEEE Symposium on Security and Privacy*, 463–480.

Chen, X.; Liu, C.; Li, B.; Lu, K.; and Song, D. 2017. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning.

Dong, Y.; Yang, X.; Deng, Z.; Pang, T.; Xiao, Z.; Su, H.; and Zhu, J. 2021. Black-Box Detection of Backdoor Attacks With Limited Information and Data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 16482–16491.

Feng, Y.; Ma, B.; Zhang, J.; Zhao, S.; Xia, Y.; and Tao, D. 2022. FIBA: Frequency-Injection Based Backdoor Attack in Medical Image Analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 20876–20885.

Goldwasser, S.; Kim, M. P.; Vaikuntanathan, V.; and Zamir, O. 2022. Planting Undetectable Backdoors in Machine Learning Models : [Extended Abstract]. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, 931–942.

Graves, L.; Nagisetty, V.; and Ganesh, V. 2021. Amnesiac Machine Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13): 11516–11524.

Gu, T.; Dolan-Gavitt, B.; and Garg, S. 2017. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain.

Guo, W.; Wang, L.; Xing, X.; Du, M.; and Song, D. 2019. TABOR: A Highly Accurate Approach to Inspecting and Restoring Trojan Backdoors in AI Systems.

Huang, X.; Alzantot, M.; and Srivastava, M. 2019. NeuronInspect: Detecting Backdoors in Neural Networks via Output Explanations.

Ilyas, A.; Santurkar, S.; Tsipras, D.; Engstrom, L.; Tran, B.; and Madry, A. 2019. Adversarial Examples Are Not Bugs, They Are Features. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Kolouri, S.; Saha, A.; Pirsiavash, H.; and Hoffmann, H. 2020. Universal Litmus Patterns: Revealing Backdoor Attacks in CNNs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, Y.; Li, Y.; Wu, B.; Li, L.; He, R.; and Lyu, S. 2021. Invisible Backdoor Attack With Sample-Specific Triggers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 16463–16472.

Lin, J.; Xu, L.; Liu, Y.; and Zhang, X. 2020. Composite Backdoor Attack for Deep Neural Network by Mixing Existing Benign Features. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, 113–131. New York, NY, USA: Association for Computing Machinery. ISBN 9781450370899.

Liu, Y.; Lee, W.-C.; Tao, G.; Ma, S.; Aafer, Y.; and Zhang, X. 2019. ABS: Scanning Neural Networks for Back-Doors by Artificial Brain Stimulation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, 1265–1282. New York, NY, USA: Association for Computing Machinery. ISBN 9781450367479.

Liu, Y.; Ma, S.; Aafer, Y.; Lee, W.-C.; Zhai, J.; Wang, W.; and Zhang, X. 2018. Trojaning Attack on Neural Networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-221, 2018*. The Internet Society.

Liu, Y.; Ma, X.; Bailey, J.; and Lu, F. 2020. Reflection Backdoor: A Natural Backdoor Attack on Deep Neural Networks. In Vedaldi, A.; Bischof, H.; Brox, T.; and Frahm, J.-M., eds., *Computer Vision – ECCV 2020*, 182–199. Cham: Springer International Publishing.

Liu, Y.; Shen, G.; Tao, G.; Wang, Z.; Ma, S.; and Zhang, X. 2022. Complex Backdoor Detection by Symmetric Feature Differencing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15003–15013.

Neel, S.; Roth, A.; and Sharifi-Malvajerdi, S. 2021. Descent-to-Delete: Gradient-Based Methods for Machine Unlearning. In Feldman, V.; Ligett, K.; and Sabato, S., eds., *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132 of *Proceedings of Machine Learning Research*, 931–962. PMLR.

Nguyen, T. A.; and Tran, A. 2020. Input-Aware Dynamic Backdoor Attack. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 3454–3464. Curran Associates, Inc.

Qiao, X.; Yang, Y.; and Li, H. 2019. Defending Neural Backdoors via Generative Distribution Modeling. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Shafahi, A.; Huang, W. R.; Najibi, M.; Suciu, O.; Studer, C.; Dumitras, T.; and Goldstein, T. 2018. Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks.

In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Shen, G.; Liu, Y.; Tao, G.; An, S.; Xu, Q.; Cheng, S.; Ma, S.; and Zhang, X. 2021. Backdoor Scanning for Deep Neural Networks through K-Arm Optimization. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 9525–9536. PMLR.

Tang, D.; Zhu, R.; Wang, X.; Tang, H.; and Chen, Y. 2022. Understanding Impacts of Task Similarity on Backdoor Attack and Detection. arXiv:2210.06509.

Tang, R.; Du, M.; Liu, N.; Yang, F.; and Hu, X. 2020. An Embarrassingly Simple Approach for Trojan Attack in Deep Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, KDD '20, 218–228. New York, NY, USA: Association for Computing Machinery. ISBN 9781450379984.

Wang, B.; Yao, Y.; Shan, S.; Li, H.; Viswanath, B.; Zheng, H.; and Zhao, B. Y. 2019. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, 707–723.

Wang, T.; Yao, Y.; Xu, F.; An, S.; Tong, H.; and Wang, T. 2022. An Invisible Black-Box Backdoor Attack Through Frequency Domain. 396–413. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-031-19777-2.

Wu, Y.; Dobriban, E.; and Davidson, S. 2020. DeltaGrad: Rapid retraining of machine learning models. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 10355–10366. PMLR.

Wu, Y.; Tannen, V.; and Davidson, S. B. 2020. PrIU: A Provenance-Based Approach for Incrementally Updating Regression Models. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, 447–462. New York, NY, USA: Association for Computing Machinery. ISBN 9781450367356.

Xu, X.; Wang, Q.; Li, H.; Borisov, N.; Gunter, C. A.; and Li, B. 2021. Detecting AI Trojans Using Meta Neural Analysis. In *2021 IEEE Symposium on Security and Privacy (SP)*, 103–120.

Yao, Y.; Li, H.; Zheng, H.; and Zhao, B. Y. 2019. Latent Backdoor Attacks on Deep Neural Networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, 2041–2055. New York, NY, USA: Association for Computing Machinery. ISBN 9781450367479.