Towards Class-Oriented Poisoning Attacks Against Neural Networks

Bingyin Zhao  Yingjie Lao
Department of Electrical and Computer Engineering
Clemson University, SC, 29634, USA
{bingyiz, ylao}@clemson.edu

Abstract

Poisoning attacks on machine learning systems compromise the model performance by deliberately injecting malicious samples in the training dataset to influence the training process. Prior works focus on either availability attacks (i.e., lowering the overall model accuracy) or integrity attacks (i.e., enabling specific instance based backdoor). In this paper, we advance the adversarial objectives of the availability attacks to a per-class basis, which we refer to as class-oriented poisoning attacks. We demonstrate that the proposed attack is capable of forcing the corrupted model to predict in two specific ways: (i) classify unseen new images to a targeted “supplanter” class, and (ii) misclassify images from a “victim” class while maintaining the classification accuracy on other non-victim classes. To maximize the adversarial effect as well as reduce the computational complexity of poisoned data generation, we propose a gradient-based framework that crafts poisoning images with carefully manipulated feature information for each scenario. Using newly defined metrics at the class level, we demonstrate the effectiveness of the proposed class-oriented poisoning attacks on various models (e.g., LeNet-5, Vgg-9, and ResNet-50) over a wide range of datasets (e.g., MNIST, CIFAR-10, and ImageNet-ILSVRC2012) in an end-to-end training setting.

1. Introduction

In recent years, machine learning has demonstrated superior performance in various fields including computer vision [25], natural language processing [11], autonomous vehicle [4], and healthcare [12]. However, it has also been shown that machine learning models are vulnerable to various types of attacks, including evasion attacks [1, 15, 29, 40] backdoor attacks [6, 16, 7, 8, 9, 41, 35] and poisoning attacks [2, 28, 31, 46, 37, 32, 45]. Evasion attacks occur at the inference phase, which causes misclassification without altering the model. Backdoor attacks raise misclassifications on specific inputs embedded with certain triggers, which requires access to both training and inference phase to inject and activate backdoor triggers. In contrast, poisoning attacks corrupt the model by only injecting malicious training data in the training phase, without requiring attackers take control of model inputs during inference. The category of attacks has drawn particular attention under the scenario where attackers are able to provide training data (e.g. online repositories).

Prior research on poisoning attacks can be broadly classified into two categories: availability attacks that aim at degrading overall model accuracy (i.e., denial-of-service attacks) [2, 44, 28, 31, 30, 21, 39, 45, 32] and integrity attacks that seek to cause misclassification on specific instances (i.e., a targeted image) [37, 47, 19]. While various capabilities of integrity attacks on deep neural networks (DNNs) have been comprehensively investigated, most prior studies of availability attacks are in a very constrained setting. Poisoning availability attacks had mainly focused on binary classification tasks until [31] proposed an efficient algorithm for multi-class attack. However, the authors explicitly pointed out that poisoning availability attack against DNN is challenge and the effect of their method is not significant. On the other hand, poisoned data are notoriously hard to craft due to computational complexity of solving the bi-level optimization (see Section 3 for details). Moreover, the major adversarial goal of prior works on poisoning availability attack is only limited to degrading the overall accuracy.

Given these limitations, we extend the poisoning availability attack against DNNs to a per-class basis. We advance the adversarial objectives by formulating two attack tasks: (i) forcing the model to classify all new inputs as a targeted class, which is denoted as the supplanter class and (ii) corrupting performance of a specific class, which is named as the victim class, while retaining the accuracy of other classes. Note that (ii) can be considered as an extension of the targeted poisoning attack that aims to induce the model to make wrong predictions on a victim class. The essential difference is that we minimize the attack impact on non-victim classes simultaneously. We propose a fast and efficient gradient-based framework for poisoned data generation, which reduces the computational complexity and generates more effective poisoned samples.
2. Related Work

Existing literature studied the poisoning availability attack on binary classification tasks against various learning algorithms such as clustering [3], LASSO [43], collaborative filtering [27], SVM [2] and logistic regression [32]. The main challenge of the poisoning attack is the generation of effective poisoned data. Prior works developed a series of gradient-based approaches for poisoned data generation, including substituting the inner minimization problem with stationary Karush-Kuhn-Tucker (KKT) conditions [28], approximating the non-convex and non-differentiable models to influence functions [23] and gradient ascent optimizations [2, 27]. In this work, we approximate the formulated optimization problems to reduce the computational complexity, which will be discussed in Section 4.

The work in [31] firstly proposed back-gradient optimization and extended poisoning availability attack to multi-class classification. However, the attack is less effective against DNNs. [45] expedited the poisoned data generation using generative models and evaluated the effect on MLP and LeNet. [14] leveraged adversarial examples as poisoned data. Prior works mostly focused on indiscriminately degrading the overall accuracy and disregarded particular prediction error of each class. In this work, we propose algorithms that focus on optimizing the feature information of the most important classes, which is able to achieve the class-oriented adversarial goals as well as facilitate the multi-class poisoning availability attack against DNNs. [13] studied an interesting learning problem that is similar to the class-oriented availability attack. However, the essential difference with [13] is that it mainly focuses on adding smallest bounded noises to the entire training data whereas our work focuses on generating a portion of the most effective poisoned data.

Poisoning integrity attacks and backdoor attacks have been extensively studied in the literature [6, 16, 41, 35, 37, 47, 19]. For instance, [6] imposed backdoor data poisoning and caused face recognition systems to misclassify images that contain a “glasses” pattern. [35] proposed hidden trigger backdoor attacks where triggers are more stealthy and imperceptible to human inspection. [37] proposed an integrity attack approach that employed feature collision to cause misclassification on a specific target image and evaluated on ImageNet dataset. [47] enhanced the transferability of integrity attacks by crafting poisoned images surrounded the targeted image and [19] accelerated poisoned data generation of such attacks. However, as we described in Section 1, poisoning availability attack is fundamentally different from poisoning integrity attacks or backdoor attacks. Note that poisoning integrity attacks and backdoor attacks are not class-oriented since the adversarial goal is only on selected instances instead of object classes.

3. Problem Settings

3.1. Poisoning Availability Attack Setting

We consider the scenario where a DNN is initialized with pre-trained weights and then updated on a poisoned dataset in a full end-to-end fashion. This scenario is one of the most pervasive poisoning attack settings and widely adopted in the state-of-the-art research literature [31, 45, 37, 47, 36] since pre-trained networks are frequently used in real-world applications. To define the problem, let $x \in \mathcal{X}(\mathbb{R}^d)$ be a $d$-dimensional input and $y \in \mathcal{Y}$ be the corresponding label. The objective of the classification task is to build up the mapping $F: \mathcal{X} \rightarrow \mathcal{Y}$. We denote the parameters of pre-trained base classifier as $\theta$. The model parameters are updated to $\theta^*$ with the incoming new stream of data: $\theta \rightarrow (x, y), \theta^*$.

Poisoning availability attacks are typically formulated as a bi-level optimization problem:

$$\arg\max_{D^p} \sum_{(x,y) \in D_{val}} L[F_{\theta^*}(x), y, \theta^*]$$  \hspace{1cm} (1)

subject to $\theta^* \in \arg\min_{\theta^* \in \Theta} \sum_{(x,y) \in D_{val} \cup D_p} L[F_{\theta^*}(x), y, \theta^*]$, \hspace{1cm} (2)

where $D_{tr}$ is the clean training dataset, $D_{val}$ is the validation dataset, $D_p$ is the poisoned dataset, $\Theta$ is the possible parameter space, and $L[\cdot]$ is the loss function. The attack aims to find an optimized poisoned dataset, which will be injected into the clean training data for training the benign model and updating parameters. This training process is expressed by the inner minimization. The outer maximization stands for the adversarial objective, which has to be evaluated on the updated parameters found by solving the inner minimization problem.

3.2. Class-Oriented Adversarial Objectives

Other than only focusing on maximizing the overall loss, we take the first step towards extending the adversarial objective to a per-class basis, as illustrated in Figure 1. We impose two new adversarial objectives in addition to the goal of degrading overall accuracy indiscriminately, which is formulated as two optimization problems accordingly.

**Problem 1:** class-oriented error-generic (COEG) attack. The goal of this attack is to misclassify all or most inputs as a targeted object class, which we named as supplanter class. Hence, the overall accuracy will also be degraded. For instance, in Figure 1, class “4” is selected as the supplanter class. For a broader real-world example, the adversary can apply the class-oriented error-generic attack to compromise a military image classifier to classify all the taken images such as birds and planes as a missile (the supplanter class), raising unnecessary panic or even wrongly activating an anti-missile system. The COEG attack problem can be formulated as:
where $y_s$ represents the label of the supplanter class.

**Problem 2: class-oriented error-specific (COES) attack.** The goal is to compromise the classification accuracy only for the inputs from a specific class, which is denoted as *victim class*, while retaining the accuracy of other classes. The bottom right chart in Figure 1 shows the poisoned model behavior when class “9” is selected as the victim class. For a broader example again, the adversary can apply the class-oriented error-specific attack to the military image classifier such that only missiles (the victim class) will not be correctly classified, resulting in severe security risks. The COES attack problem is formulated as:

$$\arg \max_{\mathcal{D}_p} \sum_{(x,y) \in \mathcal{D}_{val}} L[\theta^*, (x), y, \theta^*] \tag{3}$$

s.t. $\theta^* \in \arg \min_{\theta^* \in \Theta} \sum_{(x,y) \in \mathcal{D}_{tr} \cup \mathcal{D}_p} L[\theta^*, (x), y_s, \theta]$, \hspace{1cm} \tag{4}

where $y_s$ represents the label of the supplanter class.

**Change-from-Target (CFT) rate** is specifically used for evaluating the COES attack, which shows the percentage of images from a targeted class are misclassified due to the poisoning attack. Similarly, CFT rate for a class $C_k$ is defined by Equation (10):

$$\text{CFT}(C_k) = \frac{1}{N_k} \sum_{y_i = y_k} (\mathcal{F}_{\theta^*}(x_i)_{y_k} - \mathcal{F}_{\theta}(x_i)_{y_k}). \hspace{1cm} \tag{10}$$

where $N_k$ is the total number of images in the class $C_k$ and $y_k$ is the corresponding categorical label, while $\mathcal{F}_{\theta}$ and $\mathcal{F}_{\theta^*}$ are the model inference results before and after the poisoning attack, respectively. Then, the overall CFT rate over $\mathcal{D}_{val}$ can be calculated by weighted averaging the CFT rates of all non-supplanter classes:

$$\text{CTT} = \frac{\sum_{k=1}^{K} (N_k \cdot \text{CTT}(C_k))}{\sum_{k=1}^{K} N_k}, \hspace{1cm} \tag{9}$$

where $K$ is the total number of classes.

The **Change-to-Target (CTT)** rate is designed as an evaluation metric for the COEG attack, which indicates the percentage of images that are classified as a targeted supplanter class ($C_s$) due to the poisoning attack. CTT rate for a class $C_k$ is formally defined over a validation dataset $\mathcal{D}_{val}$ as:

$$\text{CTT}(C_k) = \frac{1}{N_k} \sum_{y_i = y_k} (\mathcal{F}_{\theta^*}(x_i)_{y_k} - \mathcal{F}_{\theta}(x_i)_{y_k}), \hspace{1cm} \tag{7}$$

**3.3. Class-Oriented Evaluation Metric**

We propose two class-oriented evaluation metrics to assess the performance of class-oriented poisoning attacks.

**Change-to-Target (CTT)** rate is designed as an evaluation metric for the COEG attack, which indicates the percentage of images that are classified as a targeted supplanter class ($C_s$) due to the poisoning attack. CTT rate for a class $C_k$ is formally defined over a validation dataset $\mathcal{D}_{val}$ as:

$$\text{CTT}(C_k) = \frac{1}{N_k} \sum_{y_i = y_k} (\mathcal{F}_{\theta^*}(x_i)_{y_k} - \mathcal{F}_{\theta}(x_i)_{y_k}), \hspace{1cm} \tag{7}$$

**3.4. Threat Model**

We consider a threat model that is consistent with prior works on poisoning availability attacks [28, 23, 31, 45, 32], which assumes the adversary to have the knowledge of the learning algorithm, hyper-parameters and clean training data. The attack is performed in an end-to-end training setting on a benign model. The adversary is able to inject crafted poisoned data and assign labels to the training dataset, which also holds the same assumption as the prior literature [2, 27, 5, 31, 32]. However, it is worth mentioning that our proposed approach also works under a more practical and strict scenario where the adversary only has limited knowledge of the model where only model architecture and
Thus, we hypothesize that if an input image only contains our algorithm can be expressed by the following minimization objective function:

\[ L = \sum_{k=1}^{K} f_{y_k}(x), \]

\[ L_{f_{y_o}} = f_{y_o}(x), \]

\[ L = \lambda_k \cdot L_{f_{y_o}} - L_{f_{y_o}}. \]

Algorithm 1: COEG Poisoned Data Generation

**Input:** \( x_o, y_o, y_s, \) supplanter class label, \( T \): max number of optimization iterations, hyper-parameters \( \lambda, \epsilon \)

**Output:** poisoned image \( x_p, y_p \)

Initialize: \( x_{p_0} = x_o - \epsilon \cdot \text{sign}\left( \nabla_{x_o}(\lambda \cdot L_{f_{y_o}} - L_{f_{y_o}}) \right) \)

while \( t < T \) do

Compute the gradient: \( \nabla = \nabla_{x_p}(\lambda \cdot L_{f_{y_o}} - L_{f_{y_o}}) \)

if \( f_{y_o}(x_{p_{t+1}}) > f_{y_o}(x_{p_t}) \) or \( f_{y_s}(x_{p_{t+1}}) < f_{y_s}(x_{p_t}) \) then

break

end if

end while

Assign \( y_p = y_s \)

Return \( x_p, y_p \)

where \( x \) is the image being optimized, which is initialized with the seed image \( x_o, y_o \) is the corresponding categorical ground-truth label, \( f_{y_o}(x) \) is the logit output of the ground-truth class, \( f_{y_k}(x) \) is the logit output of each other class, and finally \( x_p \) stands for the optimized poisoned image. \( \lambda_k \) is used to control the importance of loss terms. We attempt to maximize \( L_{f_{y_o}} \) and minimize \( L_{\sum_{k=1}^{K} f_{y_k}} \) simultaneously. Alternatively, from the perspective of entropy, we expect such optimization would also reduce the following information entropy \( H[\cdot] \):

\[ H[\sigma(f(x_p))] = - \sum_{k=1}^{K} p_k \cdot \log(p_k) \rightarrow 0, \]

where \( p_k \) is the probability for each class that is converted from the logit \( f_{y_k} \) using \( \text{softmax} \).

However, solving the optimization problem can be computationally expensive, especially for large-scale datasets that have hundreds or thousands of classes. Based on the facts that (i) the classification is determined by the largest probability and (ii) only the supplanter class is the target, we consider an approximation that simplifies the task to retain feature information of the ground-truth class and eliminate feature information of the supplanter class. In other words, we only focus on the two most important classes instead of all classes:

\[ L = \lambda \cdot L_{f_{y_o}} - L_{f_{y_o}}. \]

By solving the minimization using gradient descent, the poisoned image \( x_p \) is updated through one backward pass:

\[ x_p = x_o - \epsilon \cdot \text{sign}\left( \nabla_{x_o}(\lambda \cdot L_{f_{y_o}} - L_{f_{y_o}}) \right). \]
where $\epsilon > 0$ is the change rate. This update step can also be executed for several rounds to further enhance the poisoning effect. Algorithm 1 presents the details of poisoned data generation for the COEG attack.

### 4.2. COES Attack

The COES attack is fundamentally more challenging than the COEG attack. It requires not only compromising the accuracy of the targeted victim class but also maintaining the performance of non-victim classes, i.e., a high CFT rate for the victim class while low CFT rates for all the other classes. However, as observed in our single instance attack experiment as well as prior poisoning availability attacks [33], poisoning with data from only one class will shift the distribution of other classes to some extent. Hence, a set of poisoned data from more than a single class is necessary to achieve this adversarial goal. Intuitively, simply training images from all classes except the victim class may achieve this adversarial goal. However, such methods are inefficient in the end-to-end retraining scenario where limited training data and small learning rates are usually applied, as they always require much more training data and training epochs for poisoning.

To this end, we propose another gradient-based algorithm for the COES attack. The entire procedure of our poisoned data generation is presented in Algorithm 2. We craft the poisoned dataset as follows: (i) pick a same number of arbitrary images from each class, (ii) enlarge/dwindle feature information of the corresponding classes for each image, as detailed in Algorithm 2, and (iii) assign the ground-truth labels to the non-victim classes and the targeted label $y_p$ to the victim class. Specifically, for images from the victim class, we apply the same operations as in Algorithm 1, while we only increase the feature information of their ground-truth classes for images from other classes. The objective of COES attack can be expressed as:

$$L = \begin{cases} \lambda \cdot L_{f_{yo}} - L_{f_{yo}}, & \text{if } x_o \in C_v \\ L_{f_{yo}}, & \text{otherwise} \end{cases} \quad (16)$$

Similarly, the poisoned images $x_p$ are updated through a backward pass:

$$\begin{cases} x_o - \epsilon \cdot \text{sign}(\nabla x_o (\lambda \cdot L_{f_{yo}} - L_{f_{yo}})), & \text{if } x_o \in C_v \\ x_o + \epsilon \cdot \text{sign}(\nabla x_o (L_{f_{yo}})), & \text{otherwise} \end{cases} \quad (17)$$

### 5. Experiments

#### 5.1. Experimental Settings

As described above, we consider a similar setting as in prior works [45, 37]. We apply our proposed class-oriented poisoning availability attacks to multi-class image classification tasks using three widely-used datasets (MNIST, CIFAR-10, and ImageNet-ILSVRC2012) against popular neural network models (LeNet-5 [26], Vgg-9 [38], and ResNet-50 [17], respectively). Model details are presented in the appendix. For comparison with prior works, we implement the flipped-label (FL) attack [44] and the direct gradient method (DGM) [45] as the baseline attacks for MNIST and CIFAR-10. We also examine our poisoning availability attacks on ImageNet against a ResNet-50 model, which we hope to serve as a baseline comparison for future works. To illustrate the effectiveness of our methods under a fair comparison, we minimize the impact of significant model shifts due to large learning rates by setting the initial learning rate of the poisoning attack close to the final learning rate of the base model and applying the same decay strategy during each attack. All networks are implemented with TensorFlow and experiments are run on NVIDIA Tesla V100 GPUs.

### 5.2. Experimental Results of COEG Attack

#### 5.2.1 Single instance attack.

We first evaluate the effectiveness of our COEG attack under the single instance attack setting [45] that implements the poisoning attack by only training with a sole poisoned point. Hyper-parameters in Algorithm 1 are set as: $\lambda = 1$, $\epsilon = 0.3$ for MNIST and CIFAR-10. We present the results of MNIST and CIFAR-10 along with baseline methods (FL and DGM) in Figure 2.

It can be seen that our proposed attack is highly effective in degrading the overall model accuracy and increasing the CTT rate for the supplanter class. For example, we increase
Table 1. Comparison of CTT and accuracy before and after the COEG attack on ImageNet.

<table>
<thead>
<tr>
<th></th>
<th>Top-1 Accuracy</th>
<th>Top-5 Accuracy</th>
<th>CTT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vanilla ResNet-50</td>
<td>74.87%</td>
<td>92.02%</td>
<td>–</td>
</tr>
<tr>
<td>Poisoned by Our Attack</td>
<td>6.73%</td>
<td>15.00%</td>
<td>85.60%</td>
</tr>
</tbody>
</table>

Figure 2. Error and CTT rate comparison of single instance attack. Classes ‘4’ and ‘deer’ are selected as the supplanter class for MNIST and CIFAR-10, respectively.

Figure 3. Error and CTT rate comparison of single instance attack with smaller learning rate at $5 \times 10^{-5}$. Classes ‘4’ and ‘deer’ are selected as the supplanter class for MNIST and CIFAR-10. Seed images are from class ‘6’ and ‘frog’, respectively.

Figure 4. Error and CTT rate comparison of general poisoning attack with different poisoning fraction.

Attack with a set of poisoned data. A more general poisoning attack scenario would allow the attacker to inject a fraction of poisoned data into the clean training dataset, which indeed is a setting often adopted in prior studies [37, 31, 32]. We evaluate the effect of our attack on CIFAR-10 by using 1000 images for training and 9000 images for testing. The number of poisoned samples are controlled by the fraction parameter $\alpha$. For example, when $\alpha = 0.1$, 100 images are poisoned and 900 images remain clean. In the experiment, class “airplane” is set as the supplanter class. The poisoning attack is conducted at learning rate of $1 \times 10^{-5}$ over 20 epochs and batch size of 128. We also implement the FL attack as our baseline.

As shown in Figure 4, our attack outperforms the FL attack for all the $\alpha$ values. We achieve over 50% for both CTT rate and test error, which are 25% higher than those of the baseline attack. Note that the proposed attack not only achieves an overall better CTT rate but also performs better on a per-class basis, as indicated in Figure 5. The first column of each confusion matrix represents the number of test images classified as the supplanter class after the poisoning attack. Our approach has a darker color (higher CTT rate).
for every class than the baseline attack. Another interesting observation is that some classes, e.g., “automobile” and “ship”, are hard to change towards the supplanter class. This is possibly due to the structural similarity of these classes are less distinguishable to the targeted supplanter class. A recent study finds that different seed/target class pairs and training set size may have significantly different poisoning effect [36]. We also examine our attack’s performance from this aspect by experimenting with different seed/target pairs, whose results are presented in the appendix. We find that our attack is particularly effective for smaller training set size and outperforms the FL attack with all the seed/target pairs.

Figure 5. Confusion matrix with different poisoning fractions. Class “airplane” is selected as the supplanter class. Darker color indicates larger change into the supplanter class.

We also study the impact of different learning rates for the general poisoning attack scenario with more poisoned data. Since we achieve the best performance with a dataset size of 500, we keep the same size for this experiment. We increase the learning rate from $1 \times 10^{-5}$ to $1 \times 10^{-4}$ and keep the remaining settings the same as the previous experiments. Experiment results are shown in Figure 6. For the FL attack, both the test error and CTT rate are proportional to the learning rate. In contrast, the proposed attack with a lower learning rate achieves a higher test error for $\alpha = 0.2 \sim 0.4$, while yielding a higher CTT rate for $\alpha > 0.1$. This may be attributed to the fact that higher learning rate also amplifies the impact of clean data and partially offsets the effect of poisoned data. Despite this, our proposed attack still achieve better performance compared to the baseline attack in both criteria.

Comparison with existing works. We compare our approach with state-of-the-art poisoning availability attacks. Since prior works are not class-oriented, we only compare the accuracy drop. Taking the different experimental settings such as victim model and model accuracy into account, we refer to the best results reported in these papers for fair comparison. Also note again that this paper, to the best of our knowledge, is the first work to evaluate poisoning availability attack on ImageNet. So we only compare the performance on MNIST and CIFAR-10. As shown in Table 2, our proposed approach shows superior adversarial capability.

Comparison with existing works. We compare our approach with state-of-the-art poisoning availability attacks. Since prior works are not class-oriented, we only compare the accuracy drop. Taking the different experimental settings such as victim model and model accuracy into account, we refer to the best results reported in these papers for fair comparison. Also note again that this paper, to the best of our knowledge, is the first work to evaluate poisoning availability attack on ImageNet. So we only compare the performance on MNIST and CIFAR-10. As shown in Table 2, our proposed approach shows superior adversarial capability.

5.3. Experimental Results of COES Attack

Since the adversarial goal of the COES attack is to subvert only one class without degrading the performance of other classes, there are two important metrics for this task: 1) CFT rate of the victim class should be as high, and 2) CFT rates of the non-victim classes should be as low as possible. Note that the highest achievable CFT rate is upper-bounded by the accuracy of the victim class in the base model. Since the single image attack naturally contradicts to the COES adversarial goal (as explained in Methods Section), we only consider the general poisoning attack and keep the same training settings and hyper-parameters as in the COEG attack. We present the CFT rate of each class for poisoning attacks on CIFAR-10 in Table 3, where class “truck” is selected as the victim class and “airplane” is selected as the poisoned label. $\alpha = 0.5$ is used. We also evaluate the effect of different values of $\alpha$ in the appendix. Two different types of FL attacks are implemented for comparison: FL-1 flips the label of all poisoned images; FL-2 only flips the label of poisoned images from the victim class.

It can be seen that FL-2 is able to keep the CFT rates of all non-victim classes relatively low; however, the CFT rate of the victim class is only 8.01%. While the FL-1 attack achieves a CFT rate of 18.42% for the victim class, it also largely increases the CFT rates of non-victim classes, which indeed verifies the difficulty of achieving the class-oriented
Table 3. CFT rate of each CIFAR-10 class by poisoning.

<table>
<thead>
<tr>
<th>Class / Attack</th>
<th>airplane</th>
<th>automobile</th>
<th>bird</th>
<th>cat</th>
<th>deer</th>
</tr>
</thead>
<tbody>
<tr>
<td>FL-1</td>
<td>-3.90%</td>
<td>3.87%</td>
<td>16.80%</td>
<td>16.20%</td>
<td>32.65%</td>
</tr>
<tr>
<td>FL-2</td>
<td>-3.01%</td>
<td>-1.44%</td>
<td>-2.05%</td>
<td>-2.29%</td>
<td>2.57%</td>
</tr>
<tr>
<td>Ours</td>
<td>-7.07%</td>
<td>0.88%</td>
<td>-0.41%</td>
<td>2.29%</td>
<td>5.27%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class / Attack</th>
<th>dog</th>
<th>frog</th>
<th>horse</th>
<th>ship</th>
<th>truck(victim)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FL-1</td>
<td>19.09%</td>
<td>22.44%</td>
<td>8.34%</td>
<td>15.45%</td>
<td>18.42%</td>
</tr>
<tr>
<td>FL-2</td>
<td>2.67%</td>
<td>-0.59%</td>
<td>-0.11%</td>
<td>-0.78%</td>
<td>8.01%</td>
</tr>
<tr>
<td>Ours</td>
<td>3.32%</td>
<td>-0.47%</td>
<td>0.11%</td>
<td>0.22%</td>
<td>51.14%</td>
</tr>
</tbody>
</table>

adversarial goal by using prior poisoning attack methods. Compared to the baseline attacks, our proposed approach can effectively increase the CFT rate of the victim class to 51.14%, which significantly surpasses the performance of both baseline attacks. Meanwhile, our method is able to retain the CFT rates of the non-victim classes to be less than 5.27%. In some cases, the proposed approach even improves the accuracy of some non-victim classes, which is indicated by the negative CFT rates in our experimental results. Such performance is beyond our expectation, which however does not contradict the adversarial objective of COES attack, i.e., only degrading the accuracy of the victim class. Similar to the COEG attack, we find that the proposed COES attack method is more resilient to the variation of learning rate than baseline attacks. We present these results in the appendix.

6. Possible Defenses

Since the main objective of this paper is to extend the adversarial capability of poisoning availability attack to a per-class basis on deep neural networks, we expect the proposed attacks to have similar performance as prior poisoning availability attacks in general when evaluated under possible defenses.

Data sanitization is a defensive technique against poisoning attack that works by distinguishing and removing outliers (poisoned data) from the training dataset [10, 34]. However, it has been shown that a broad range of data sanitization can be easily compromised or bypassed [24]. Therefore, we can also leverage such techniques for our proposed attacks to evade detection. In fact, in most recent works on poisoning attacks, data sanitization is no longer considered a certified defensive strategy [32, 21, 42].

Alternatively, a possible countermeasure is to periodically check the accuracy and/or loss of the learning models [31, 45]. Although expensive in terms of cost and time, these approaches are intuitively effective based on the fact that poisoning availability attack aims at degrading the accuracy. Since the poisoned data are tailored to influence the learning model’s training process maliciously, we suggest exploiting averaged stochastic gradient classifier [42] and combinational models such as bagging [27], where the classification results are no longer dependent on a single model, to defend against the poisoning attack. However, the overhead for deploying multiple classifiers should also be carefully considered.

7. Conclusions

This paper introduced the concept of class-oriented poisoning attack. We formulated two attack problems, i.e., “COEG” and “COES”, which seek to compromise the model behavior on a per-class basis. Accordingly, we defined two new metrics to evaluate the performance of poisoning attacks at the class level. Our proposed gradient-based algorithms successfully achieved the class-oriented adversarial objectives through manipulating the feature information in images for poisoned data generation. The effectiveness of the proposed methods is comprehensively studied in our experiments.

Acknowledgment

This work is partially supported by the National Science Foundation award 2047384.
References


[29] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to

3749


