

CS 499/599: Machine Learning Security

02.09: Data Poisoning

Mon/Wed 12:00 – 1:50 pm

Sanghyun Hong

sanghyun.hong@oregonstate.edu



Oregon State
University

SAIL
Secure AI Systems Lab

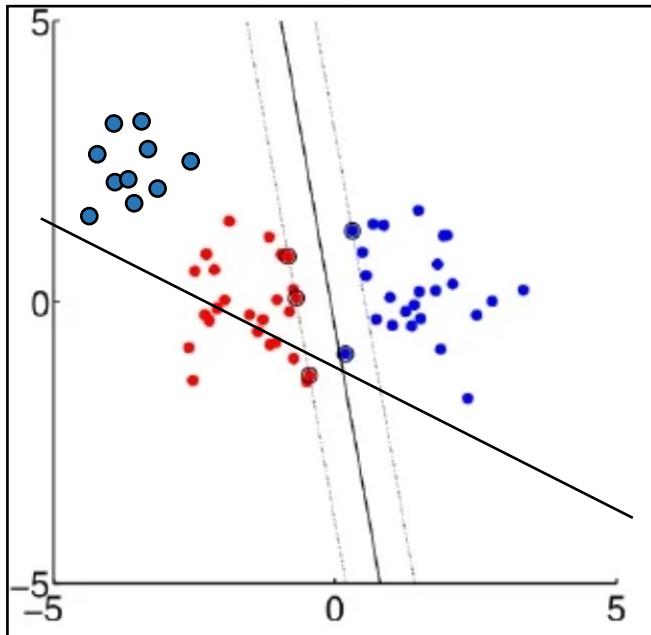
Notice

- Due dates
 - Written Paper Critiques (on the 14th)
- Sign-up (on Canvas)
 - Scribe Lecture Note
 - In-class Paper Presentation / Discussion

Topics for Today

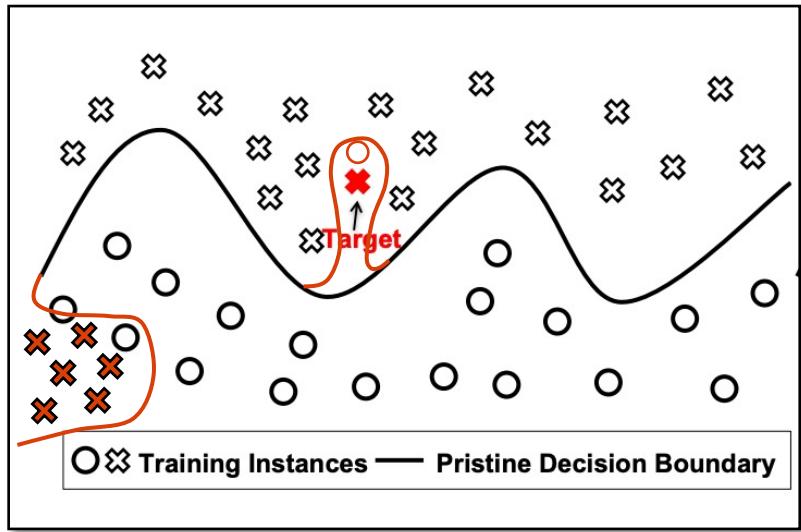
- Motivation
 - Evade spam filter
 - Evade DDoS detection
- Data Poisoning:
 - Indiscriminate Attacks
 - Support vector machines (SVMs)
 - Regression models
 - Targeted Attacks
 - Deep neural networks (DNNs)
 - Poison Frogs
 - Meta-Poison
 - Conclusion (and implications)

Revisited: Linear Models vs. DNNs



← Linear model (SVM)

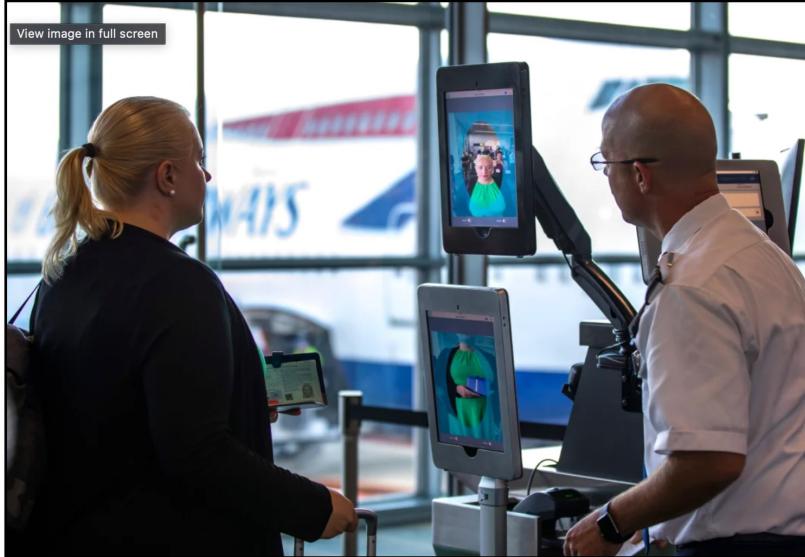
Neural Network →



Shafahi *et al.*, Poison Frogs! Targeted Clean-label Poisoning Attacks on Neural Networks
Huang *et al.*, MetaPoison: Practical General-purpose Clean-label Data Poisoning

Motivation

- Practical Constraints
 - You don't have control over your inputs
 - You don't want to mess-up the entire system

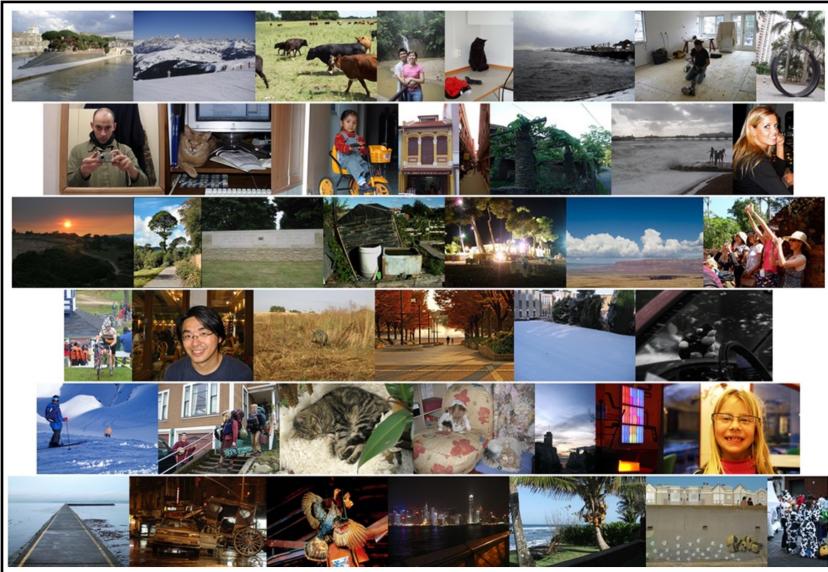


Targeted Poisoning Attacks!

gy-u-s-airports/

Motivation

- Practicality
 - Many ways to insert your malicious data
 - Human-inspection is not feasible (think about t



Data collection method

The Weibo data is obtained by a python crawler. The crawler automatically uses Weibo's advanced search function for keyword indexing.

The keywords we used included:

- COVID-19
- novel coronavirus(新型冠状病毒)
- corona(新冠)
- epidemics(疫情)
- novel pneumonia(新型肺炎)
- pneumonia in Wuhan(武汉+肺炎)

The crawler program automatically entered one of the keywords into the query box and set the query time range to be a specific hour. As illustrated in Figure 2, the crawler sets the time range from January 18, 2020, 00:00:00 to January 18, 2020, 01:00:00.

For each query, the time range increased by one hour, and each query searched all the new posts within an hour. Each query returned a maximum of 50 pages, each contained around 20 posts. If the number posts exceed the page limits, we cannot fully collect the information due to the limitations of the search function.

微博搜索 疫情 搜索 高级搜索

综合 找人 文章 视频 图片 话题

2020-01-18 00时~2020-01-18 01时

微博高级搜索 query: 疫情

关键词: 疫情

类型: 全部 (热门) 原创 关注人 认证用户 媒体 观点

包含: 全部 (含图片) 含视频 含音乐 含短链

时间: 2020-01-18 08时 至 2020-01-18 1时

地点: 省/直辖市 城市/地区 time range

搜索微博 取消

热搜榜

热搜词	热度
小学生复课防护路线图	419万
疫情 没有这样的画面	128万
汉超道歉	125万
钟大当爹	121万
阿房	115万
子柒海外粉丝破千万	94万
雷产粮计划	83万
熟悉的熊猫	63万
龙江被打女童父亲被	52万
因违反禁令被逮捕	48万
两会召开时间	10

<https://yiling-chen.github.io/flickr-cropping-dataset/>
<https://github.com/ylenge/COVID-Weibo>

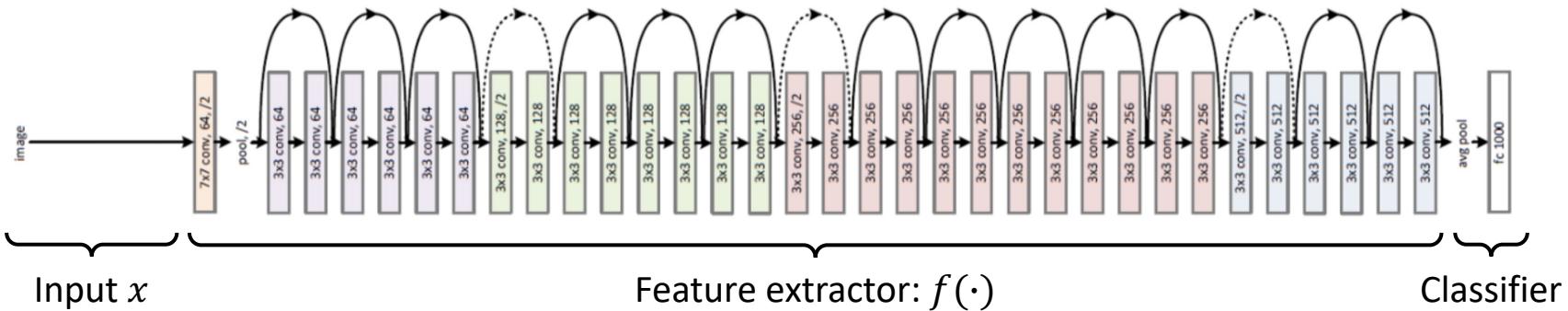
Threat Model: Targeted Poisoning

- Goal
 - Indiscriminate attack (increase the error on D_{val})
 - Targeted attack: cause a misclassification of (x_t, y_t) , while preserving acc. on D_{val}
- Capability
 - Train a model f on D_{tr}
 - Inject p poisons into the training set ($N(D_{tr}) = n + p$)
- Knowledge
 - D_{tr} : training data
 - D_{val} : validation data
 - f : a model and its parameters
 - L : training algorithm (e.g., SGD)

Threat Model: Clean-label Targeted Poisoning

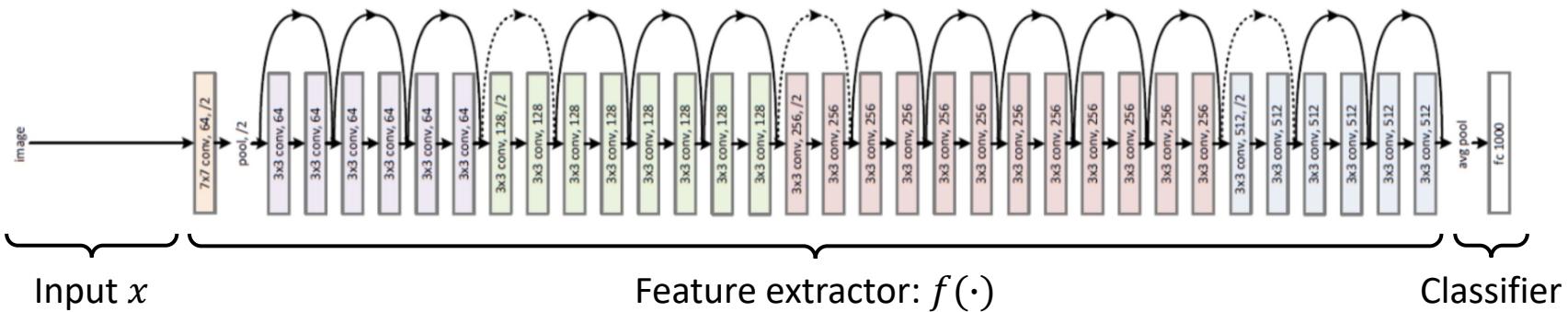
- Goal
 - Indiscriminate attack (increase the error on D_{val})
 - Targeted attack: cause a misclassification of (x_t, y_t) , while preserving acc. on D_{val}
- Capability
 - Train a model f on D_{tr}
 - Inject p poisons into the training set ($N(D_{tr}) = n + p$)
- Knowledge
 - D_{tr} : training data
 - D_{val} : validation data
 - f : a model and its parameters
 - L : training algorithm (e.g., SGD)

Background: Convolutional Neural Networks



- A conventional view:
 - Convolutions: extract features, embeddings, latent representations, ...
 - Last layer: uses the output for a classification task

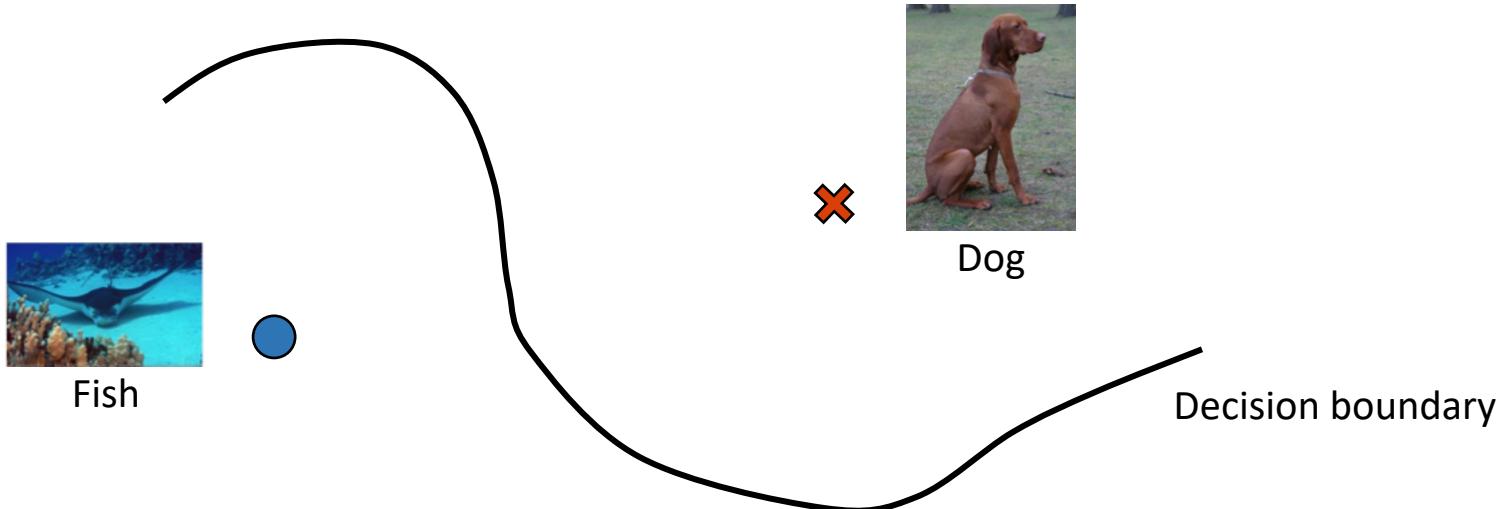
Background: Convolutional Neural Networks



- Input-space \neq Feature-space:
 - Two samples similar in the input-space can be far from each other in the feature-space
 - Two samples very different in the input-space can be close to each other in f

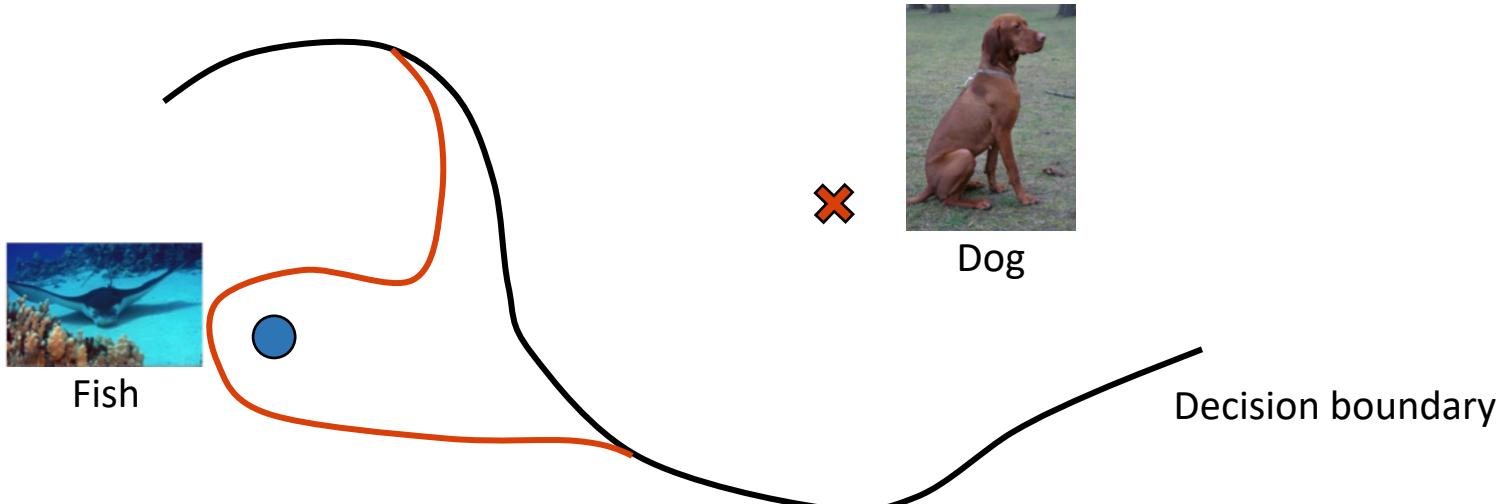
The Key Idea: Feature Collision

- Goal
 - You want your *any* poison to be closer to your target (x_t, y_t) in the *feature space*



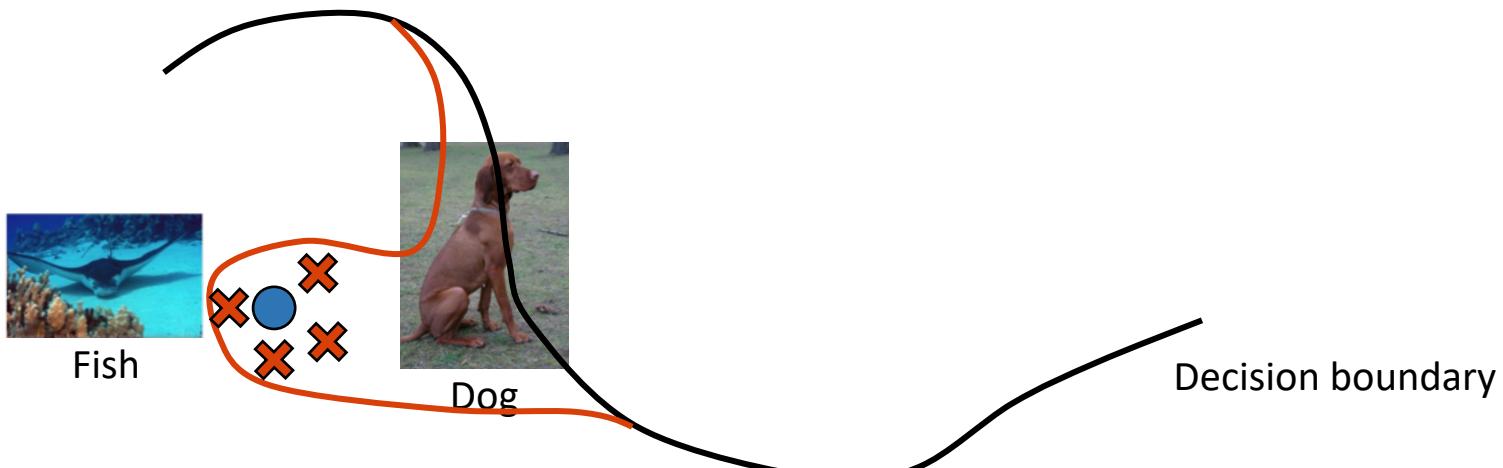
The Key Idea: Feature Collision

- Goal
 - You want your *any* poison to be closer to your target (x_t, y_t) in the *feature space*



The Key Idea: Feature Collision

- Goal
 - You want your *any* poison to be closer to your target (x_t, y_t) in the *feature space*



The Key Idea: Feature Collision

- Goal

- You want your *any* poison to be closer to your target (x_t, y_t) in the *feature space*
 - Objective:

$$\mathbf{p} = \underset{\mathbf{x}}{\operatorname{argmin}} \ \|f(\mathbf{x}) - f(\mathbf{t})\|_2^2 + \beta \|\mathbf{x} - \mathbf{b}\|_2^2$$

- Optimization:

Algorithm 1 Poisoning Example Generation

Input: target instance t , base instance b , learning rate λ

Initialize \mathbf{x} : $x_0 \leftarrow b$

Define: $L_p(x) = \|f(\mathbf{x}) - f(\mathbf{t})\|^2$

for $i = 1$ **to** $maxIters$ **do**

 Forward step: $\hat{x}_i = x_{i-1} - \lambda \nabla_x L_p(x_{i-1})$ // construct input perturbations

 Backward step: $x_i = (\hat{x}_i + \lambda \beta b) / (1 + \beta \lambda)$ // decide how much we will perturb

end for

Evaluations

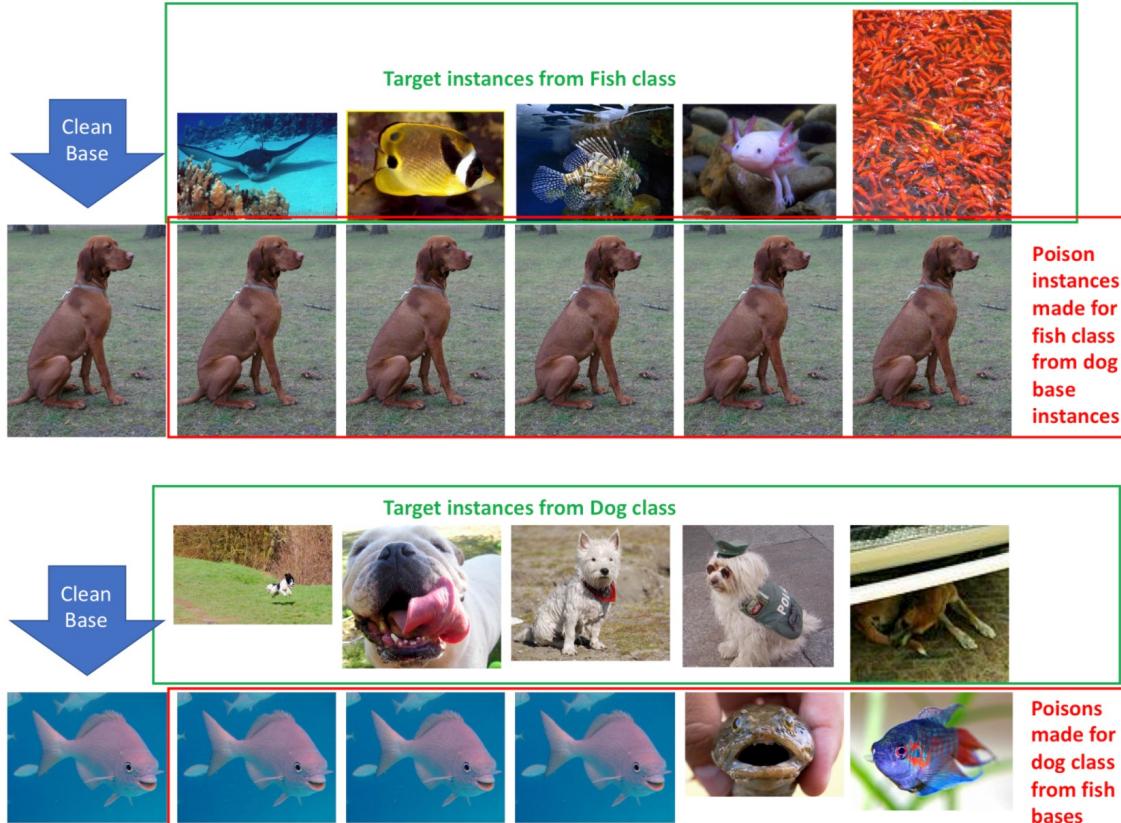
- Scenarios
 - Scenario 1: Transfer learning
 - Scenario 2: End-to-end learning

Evaluations: Transfer Learning

- Setup
 - Dataset: Dog vs. Fish (ImageNet)
 - Models: Inception-V3 (Pretrained on ImageNet)
- “one-shot kill” Attacks
 - Goal: Dog > Fish or Fish > Dog | All 1099 targets from the test-set
 - Craft a poison using a single image chosen from the other class
 - Train the last layer on $D_{tr} \cap (x_p, y_p)$ and check if the target’s label is flipped
- Results
 - The attack succeeds with 100% accuracy
 - The accuracy drop caused by the attack is 0.2% on average

Evaluations: Transfer Learning

- Examples

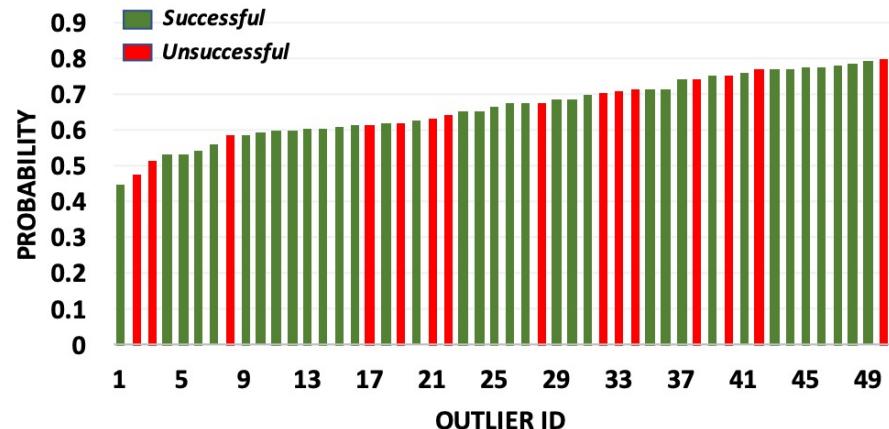
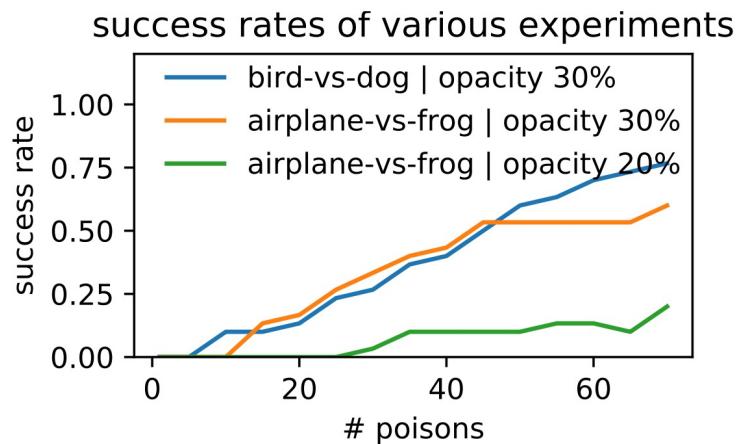


Evaluations: End-to-End Learning

- Setup
 - Dataset: CIFAR-10
 - Models: AlexNet (Pretrained on CIFAR-10)
- “end-to-end” Attacks
 - Goal: Bird > Dog or Airplane > Frog
 - Craft 1-70 poisons using the images chosen from the (Dog or Frog) class
 - Trick: watermarking!
 - Train the entire model on $D_{tr} \cap (x_p, y_p)$ and check the misclassification rate

Evaluations: End-to-End Learning

- Results



Topics for Today

- Motivation
 - Evade spam filter
 - Evade DDoS detection
- Data Poisoning:
 - Indiscriminate Attacks
 - Support vector machines (SVMs)
 - Regression models
 - Targeted Attacks
 - Deep neural networks (DNNs)
 - Poison Frogs
 - Meta-Poison
 - Conclusion (and implications)

Revisit the Threat Model

- Goal
 - ~~Indiscriminate attack (increase the error on D_{val})~~
 - Targeted attack: cause a misclassification of (x_t, y_t) , while preserving acc. on D_{val}
- Capability
 - Train a model f on D_{tr}
 - Perturb p training samples in the training set ($N(D_{tr}) = n$)
- Knowledge
 - ~~D_{tr} : training data~~
 - D_{val} : validation data
 - f : a model and its parameters (Ugh!)
 - ~~L : training algorithm (e.g., SGD)~~

Revisit the Key Idea

- Goal
 - You want your *any* poison to be closer to your target (x_t, y_t) in the *feature space*
 - Objective:

$$\mathbf{p} = \operatorname{argmin}_{\mathbf{x}} \underbrace{\|f(\mathbf{x}) - f(\mathbf{t})\|_2^2}_{\text{Now you don't know the } f} + \beta \|\mathbf{x} - \mathbf{b}\|_2^2$$

- Revisit the previous idea
 - Bi-level optimization

$$\begin{aligned} \operatorname{argmax}_{\mathcal{D}_p} \quad & \mathcal{W}(\mathcal{D}', \boldsymbol{\theta}_p^*) , \\ \text{s.t.} \quad & \boldsymbol{\theta}_p^* \in \operatorname{argmin}_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{D}_{\text{tr}} \cup \mathcal{D}_p, \boldsymbol{\theta}) \end{aligned}$$

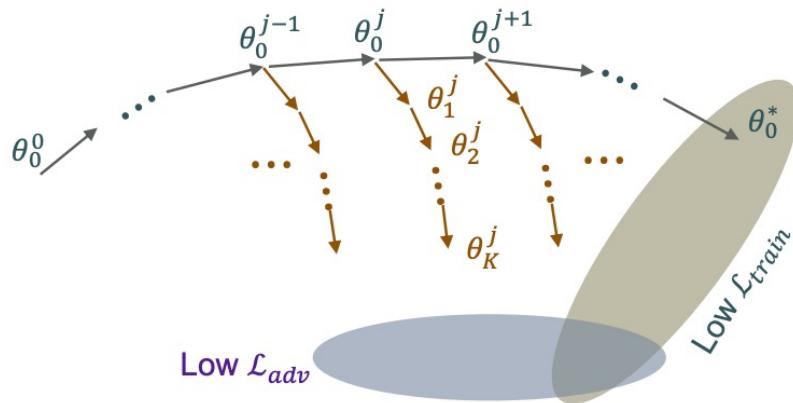
$$\begin{aligned} X_p^* &= \operatorname{argmin}_{X_p} \mathcal{L}_{\text{adv}}(x_t, y_{\text{adv}}; \theta^*(X_p)) \\ \theta^*(X_p) &= \operatorname{argmin}_{\theta} \mathcal{L}_{\text{train}}(X_c \cup X_p, Y; \theta) \end{aligned}$$

Problem: no control over θ

The Key Idea: Unrolling

- Goal

- You simulate all the training procedures with all f, θ while crafting your poisons



Algorithm 1 Craft poison examples via MetaPoison

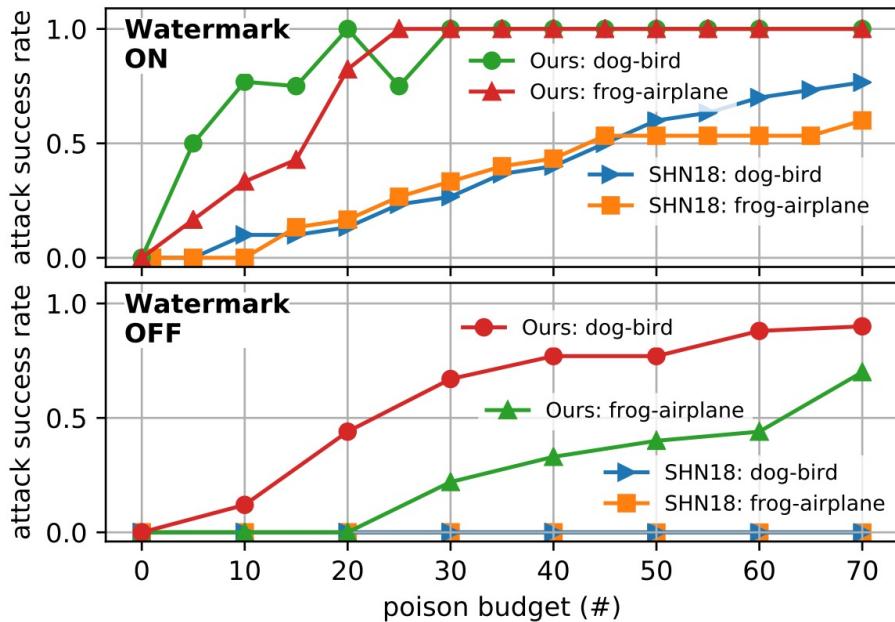
- 1: **Input** Training set of images and labels (X, Y) of size N , target image x_t , adversarial class y_{adv} , ϵ and ϵ_c thresholds, $n \ll N$ subset of images to be poisoned, T range of training epochs, M randomly initialized models.
- 2: **Begin**
- 3: Stagger the M models, training the m th model weights θ_m up to $\lfloor mT/M \rfloor$ epochs
- 4: Select n images from the training set to be poisoned, denoted by X_p . Remaining clean images denoted X_c
- 5: For $i = 1, \dots, C$ crafting steps:
- 6: For $m = 1, \dots, M$ models:
- 7: Copy $\tilde{\theta} = \theta_m$
- 8: For $k = 1, \dots, K$ unroll steps^a:
- 9: $\tilde{\theta} = \tilde{\theta} - \alpha \nabla_{\tilde{\theta}} \mathcal{L}_{\text{train}}(X_c \cup X_p, Y; \tilde{\theta})$
- 10: Store adversarial loss $\mathcal{L}_m = \mathcal{L}_{\text{adv}}(x_t, y_{\text{adv}}; \tilde{\theta})$
- 11: Advance epoch $\theta_m = \theta_m - \alpha \nabla_{\theta_m} \mathcal{L}_{\text{train}}(X, Y; \theta_m)$
- 12: If θ_m is at epoch $T + 1$:
- 13: Reset θ_m to epoch 0 and reinitialize
- 14: Average adversarial losses $\mathcal{L}_{\text{adv}} = \sum_{m=1}^M \mathcal{L}_m / M$
- 15: Compute $\nabla_{X_p} \mathcal{L}_{\text{adv}}$
- 16: Update X_p using Adam and project onto ϵ, ϵ_c ball
- 17: **Return** X_p

Evaluations

- Setup
 - Dataset: CIFAR-10
 - Models: 6-layer ConvNet (default), ResNet20, VGG13
 - Attack hyper-parameters:
 - C: 60 | M: 24 | K: 2
- Attacks
 - 30 randomly chosen targets
 - Increase the # poisons from 1 – 10% of the training data n
 - Baseline:
 - Poison Frog

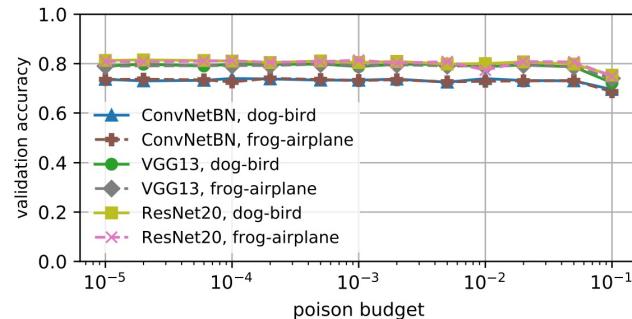
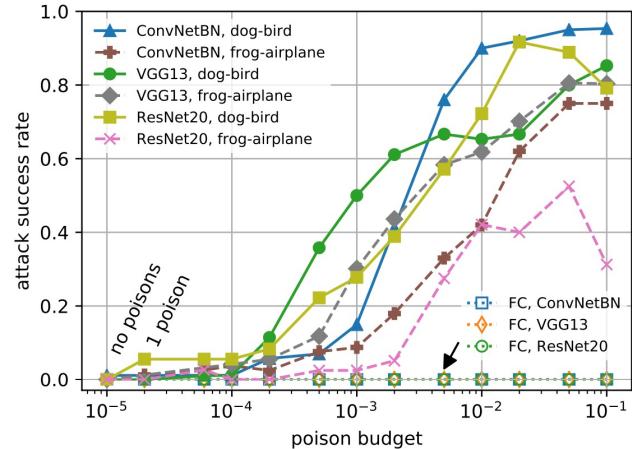
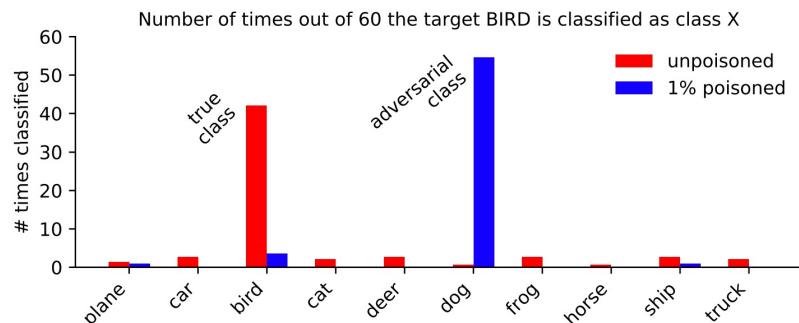
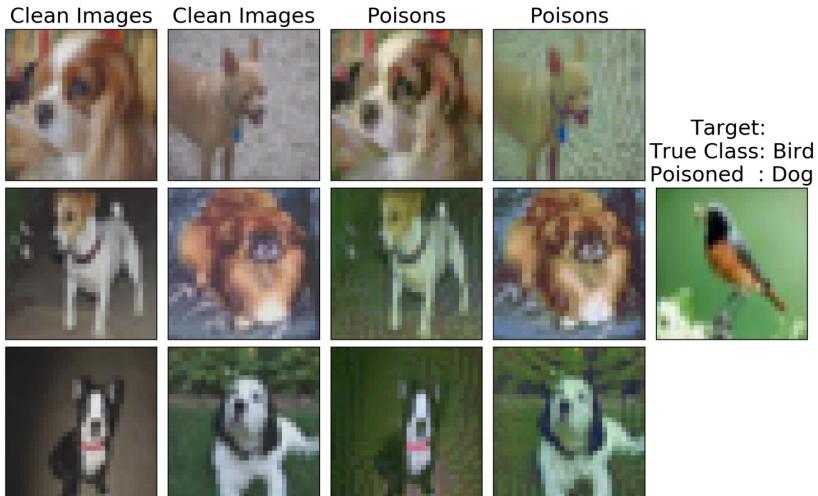
Evaluations: Transfer Learning Scenario

- Results vs. Poison Frogs



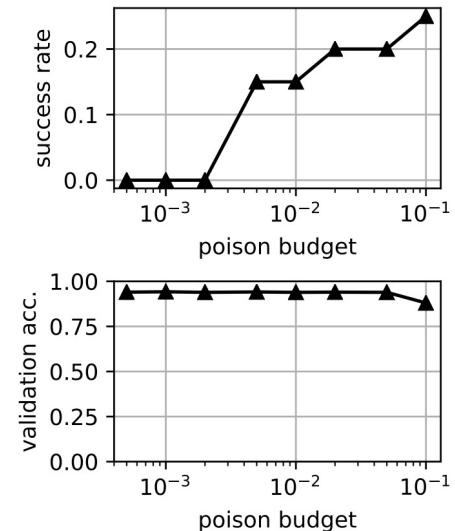
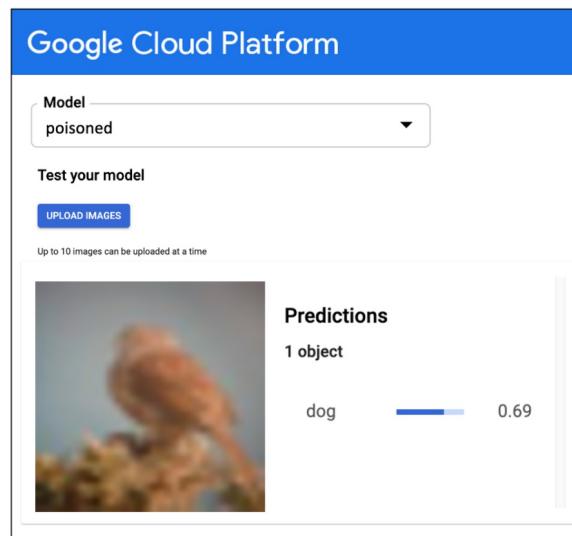
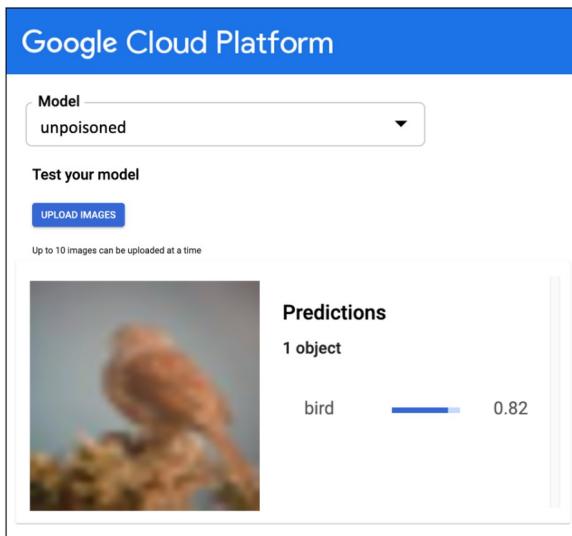
Evaluations: End-to-End Scenario

- Results



Evaluations: Exploitation in Real-World

- Results



Recap

- Motivation
 - Evade spam filter
 - Evade DDoS detection
- Data Poisoning:
 - Indiscriminate Attacks
 - Support vector machines (SVMs)
 - Regression models
 - Targeted Attacks
 - Deep neural networks (DNNs)
 - Poison Frogs
 - Meta-Poison
 - Conclusion (and implications)

Thank You!

Mon/Wed 12:00 – 1:50 pm

Sanghyun Hong

<https://secure-ai.systems/courses/MLSec/W22>



Oregon State
University

SAIL
Secure AI Systems Lab