

CY 7790

Special Topics in Security and Privacy:
Machine Learning Security and
Privacy
Fall 2021

Alina Oprea
Associate Professor
Khoury College of Computer Science

September 30 2021

Today's Class

- Continue discussion on evasion attacks
 - Minimum distance evasion attacks for different norms
 - Different objectives for optimization
 - Carlini and Wanger. Towards Evaluating the Robustness of Neural Networks. Best Paper at IEEE S&P 2017
- Discuss transferability of attacks across models and training data
 - Pepernot et al. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples

Towards Evaluating the Robustness of Neural Networks by Carlini and Wagner

Discussion led by: Michael Davinroy



Problem Statement

- *Defensive Distillation*

- Breaks known attacks

- Background

- L-BFGS
 - Discussed last week
- Fast Gradient Sign
 - Fast, but far from optimal
 - Iterative model improves attack
- JSMA
 - Finds *saliency map*
 - Really powerful, but exceptionally slow
- Deepfool
 - Assumes DNNs are totally linear
 - Keeps going until an adversarial example is found

$$\text{minimize } c \cdot \|x - x'\|_2^2 + \text{loss}_{F,l}(x')$$

$$\text{such that } x' \in [0, 1]^n$$

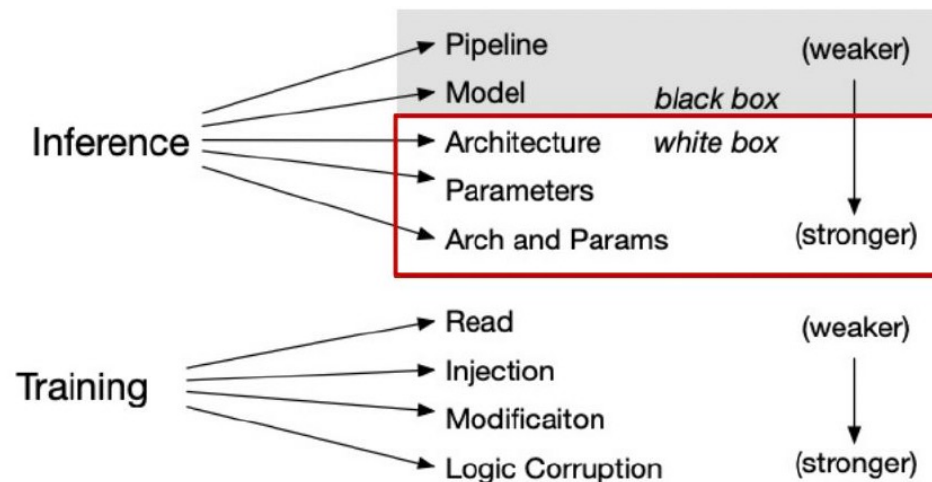
$$x' = x - \epsilon \cdot \text{sign}(\nabla \text{loss}_{F,t}(x)),$$

$$x'_i = x'_{i-1} - \text{clip}_\epsilon(\alpha \cdot \text{sign}(\nabla \text{loss}_{F,t}(x'_{i-1})))$$

$$(p^*, q^*) = \arg \max_{(p,q)} (-\alpha_{pq} \cdot \beta_{pq}) \cdot (\alpha_{pq} > 0) \cdot (\beta_{pq} < 0)$$

Threat Model

- Evasion Attack
 - Crafted offline and fool at test time
- Totally white-box
 - Know victim's architecture and parameters
- Cite *Transferability in Machine Learning* paper
 - Claim this paper makes white-box attacks as realistic as black-box attacks
 - Do we think this is always the case?
 - Is the assumption reasonable here?



Methodology: Approach

$$\begin{aligned} &\text{minimize } \mathcal{D}(x, x + \delta) \\ &\text{such that } C(x + \delta) = t \\ &\quad x + \delta \in [0, 1]^n \end{aligned}$$

$$\begin{aligned} &\text{minimize } \mathcal{D}(x, x + \delta) + c \cdot f(x + \delta) \\ &\text{such that } x + \delta \in [0, 1]^n \end{aligned}$$

The above formulation is difficult for existing algorithms to solve directly, as the constraint $C(x + \delta) = t$ is highly non-linear. Therefore, we express it in a different form that is better suited for optimization. We define an objective function f such that $C(x + \delta) = t$ if and only if $f(x + \delta) \leq 0$. There are many possible choices for f :

$$f_1(x') = -\text{loss}_{F,t}(x') + 1$$

$$f_2(x') = (\max_{i \neq t} (F(x')_i) - F(x')_t)^+$$

$$f_3(x') = \text{softplus}(\max_{i \neq t} (F(x')_i) - F(x')_t) - \log(2)$$

$$f_4(x') = (0.5 - F(x')_t)^+$$

$$f_5(x') = -\log(2F(x')_t - 2)$$

$$f_6(x') = (\max_{i \neq t} (Z(x')_i) - Z(x')_t)^+$$

$$f_7(x') = \text{softplus}(\max_{i \neq t} (Z(x')_i) - Z(x')_t) - \log(2)$$

where s is the correct classification, $(e)^+$ is short-hand for $\max(e, 0)$, $\text{softplus}(x) = \log(1 + \exp(x))$, and $\text{loss}_{F,s}(x)$ is the cross entropy loss for x .

Methodology: Attacks for Different Distances

$D = L_0, L_2, \text{ or } L_{\text{inf}}$

- L_2 :
minimize $\|\frac{1}{2}(\tanh(w) + 1) - x\|_2^2 + c \cdot f(\frac{1}{2}(\tanh(w) + 1))$
with f defined as
$$f(x') = \max(\max\{Z(x')_i : i \neq t\} - Z(x')_t, -\kappa).$$
- L_0 :
 - Non-differentiable
 - Use iterative attack with L_2 attack to find important pixels
- L_{inf} :

$$\text{minimize } c \cdot f(x + \delta) + \|\delta\|_{\infty}$$

$$\text{minimize } c \cdot f(x + \delta) + \sum_i [(\delta_i - \tau)^+]$$

Experiments

	Best Case				Average Case				Worst Case			
	MNIST		CIFAR		MNIST		CIFAR		MNIST		CIFAR	
	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob
Our L_0	8.5	100%	5.9	100%	16	100%	13	100%	33	100%	24	100%
JSMA-Z	20	100%	20	100%	56	100%	58	100%	180	98%	150	100%
JSMA-F	17	100%	25	100%	45	100%	110	100%	100	100%	240	100%
Our L_2	1.36	100%	0.17	100%	1.76	100%	0.33	100%	2.60	100%	0.51	100%
Deepfool	2.11	100%	0.85	100%	—	-	—	-	—	-	—	-
Our L_∞	0.13	100%	0.0092	100%	0.16	100%	0.013	100%	0.23	100%	0.019	100%
Fast Gradient Sign	0.22	100%	0.015	99%	0.26	42%	0.029	51%	—	0%	0.34	1%
Iterative Gradient Sign	0.14	100%	0.0078	100%	0.19	100%	0.014	100%	0.26	100%	0.023	100%

TABLE IV

COMPARISON OF THE THREE VARIANTS OF TARGETED ATTACK TO PREVIOUS WORK FOR OUR MNIST AND CIFAR MODELS. WHEN SUCCESS RATE IS NOT 100%, THE MEAN IS ONLY OVER SUCCESSES.

	Best Case				Average Case				Worst Case			
	MNIST		CIFAR		MNIST		CIFAR		MNIST		CIFAR	
	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob	mean	prob
Our L_0	10	100%	7.4	100%	19	100%	15	100%	36	100%	29	100%
Our L_2	1.7	100%	0.36	100%	2.2	100%	0.60	100%	2.9	100%	0.92	100%
Our L_∞	0.14	100%	0.002	100%	0.18	100%	0.023	100%	0.25	100%	0.038	100%

TABLE VI

COMPARISON OF OUR ATTACKS WHEN APPLIED TO DEFENSIVELY DISTILLED NETWORKS. COMPARE TO TABLE IV FOR UNDISTILLED NETWORKS.

ImageNet Results

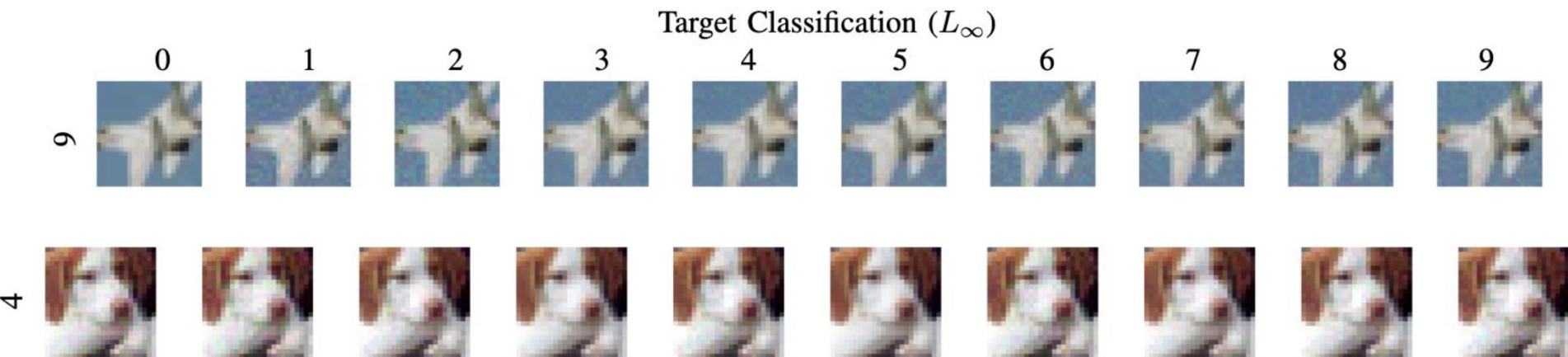
	Untargeted			Average Case			Least Likely	
	mean	prob		mean	prob		mean	prob
Our L_0	48	100%		410	100%		5200	100%
JSMA-Z	-	0%		-	0%		-	0%
JSMA-F	-	0%		-	0%		-	0%
Our L_2	0.32	100%		0.96	100%		2.22	100%
Deepfool	0.91	100%		-	-		-	-
Our L_∞	0.004	100%		0.006	100%		0.01	100%
FGS	0.004	100%		0.064	2%		-	0%
IGS	0.004	100%		0.01	99%		0.03	98%

TABLE V

COMPARISON OF THE THREE VARIANTS OF TARGETED ATTACK TO PREVIOUS WORK FOR THE INCEPTION V3 MODEL ON IMAGENET. WHEN SUCCESS RATE IS NOT 100%, THE MEAN IS ONLY OVER SUCCESSES.

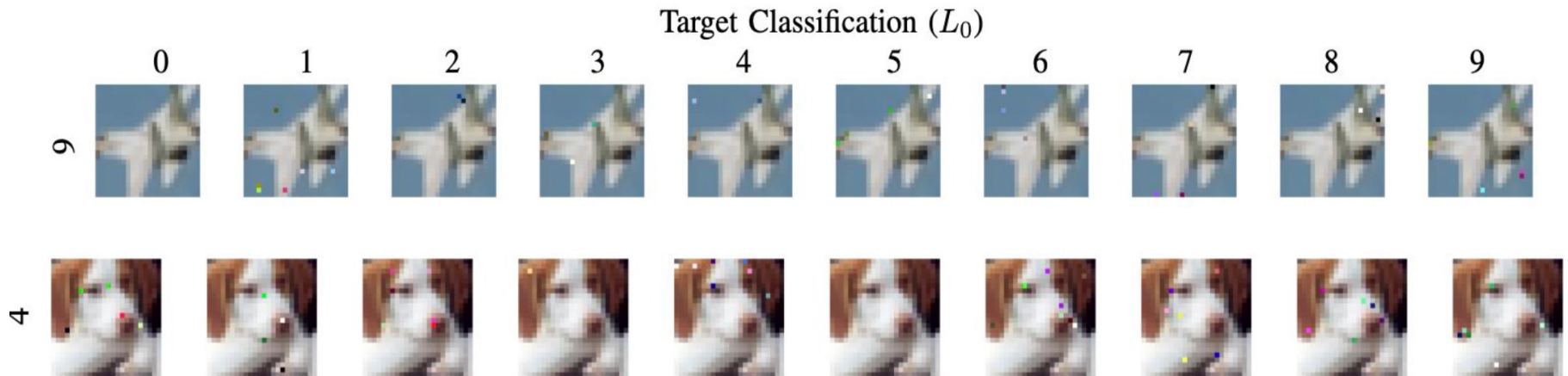
Strengths

- Improvement on the field and also a semi SoK
- Really general attack technique
- Highlights the importance of having a good attack upper bound
- Code publicly available



Limitations

- L_0 norm is pretty detectable
- Slow
 - “No attack takes longer than a few minutes to run on any given instance”
 - “Our attacks are typically 10x - 100x slower than previous attacks for L_2 and L_∞ , with exception of iterative gradient sign which we are 10x slower”
 - Can't really be used for adversarial training



Discussion

- How L_0 and L_∞ attacks leverage L_2 attack
- Optimization objective choice
- More complex models (ImageNet) have lower perturbation
- Why distillation is not effective?
 - Optimization is performed in the logit layer
 - Previous attacks compute the gradient after the softmax layer, which are very small
- High confidence adversarial examples
 - Have difference in prediction higher than threshold

Discussion

- Why is this better than the previous papers last week?
- Is this still a good attack?
- Were their experiments enough / convincing?
- How could this attack be made better?

Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples

Anish Athalye^{*1} Nicholas Carlini^{*2} David Wagner²

Abstract

We identify obfuscated gradients, a kind of gradient masking, as a phenomenon that leads to a false sense of security in defenses against adversarial examples. While defenses that cause obfuscated gradients appear to defeat iterative optimization-based attacks, we find defenses relying on this effect can be circumvented. We describe characteristic behaviors of defenses exhibiting the effect, and for each of the three types of obfuscated gradients we discover, we develop attack techniques to overcome it. In a case study, examining non-certified white-box-secure defenses at ICLR 2018, we find obfuscated gradients are a common occurrence, with 7 of 9 defenses relying on obfuscated gradients. Our new attacks successfully circumvent 6 completely, and 1 partially, in the original threat model each paper considers.

Nicolas Papernot and Patrick McDaniel

Ian Goodfellow

Prepared By: Gokberk Yar

Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples

Problem

- Understand transferability of evasion attacks in two settings
 - Intra-technique (same training algorithm, but different hyper-parameters and training data)
 - Cross-technique (different training algorithm)
- Attacker scenarios
 - Black-box (in some cases have knowledge of model type)
- Attacker capabilities
 - Evasion attack: Modifications of testing data
 - Have access to small training set
 - Query original model to get more labels

Taxonomy of the Paper

Test time

Evasion Attack

Multiclass

RGB Image (MNIST)

Untargeted

Perpetuation Attack General Concept


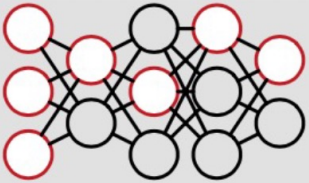

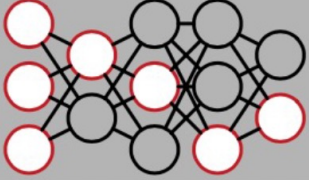
	Input	Model Activations	Output
Legitimate			1
Adversarial			4

Figure 1: An adversarial sample (bottom row) is produced by slightly altering a legitimate sample (top row) in a way that forces the model to make a wrong prediction whereas a human would still correctly classify the sample [19].

Hypotheses and Contributions

Hypothesis 1: *Both intra-technique and cross-technique adversarial sample transferabilities are consistently strong phenomena across the space of machine learning techniques.*

Hypothesis 2: *Black-box attacks are possible in practical settings against any unknown machine learning classifier.*

Methodology

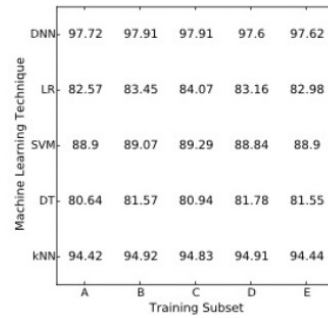
- Models: DNN, LR, SVM, DT, Ensembles
- Evasion attacks for white-box
 - Uses FGSM for DNN and LR
 - For SVM: variant of FGSM, but move in direction orthogonal to the decision boundary
 - DT: new algorithm
- Intra-technique
 - Train 5 models of the same type with different data
 - Show matrix of transferability across models
- Cross technique: for each model, find adversarial examples, and transfer to all other models
- Learning substitutes: data augmentation and reservoir sampling

ML Algorithms

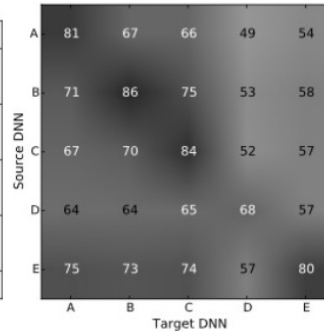
ML Technique	Differentiable Model	Linear Model	Lazy Prediction
DNN	Yes	No	No
LR	Yes	Log-linear	No
SVM	No	No	No
DT	No	No	No
kNN	No	No	Yes
Ens	No	No	No

Table 1: Machine Learning Techniques studied in Section 3

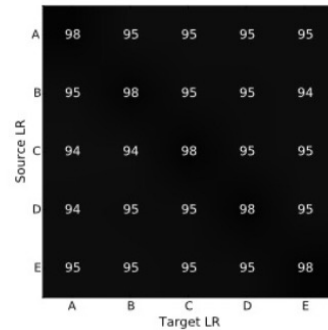
Results: Intra-Technique



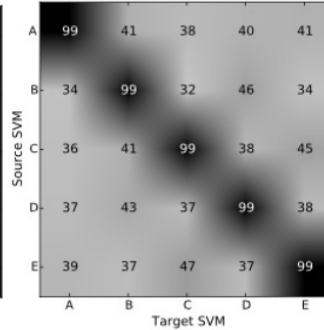
(a) Model Accuracies



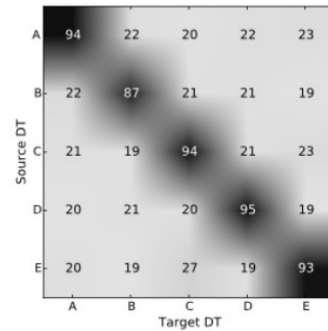
(b) DNN models



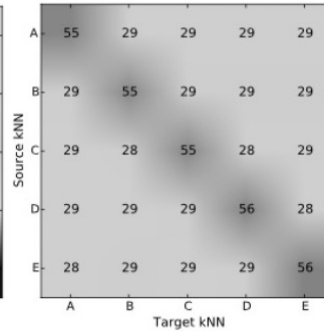
(c) LR models



(d) SVM models

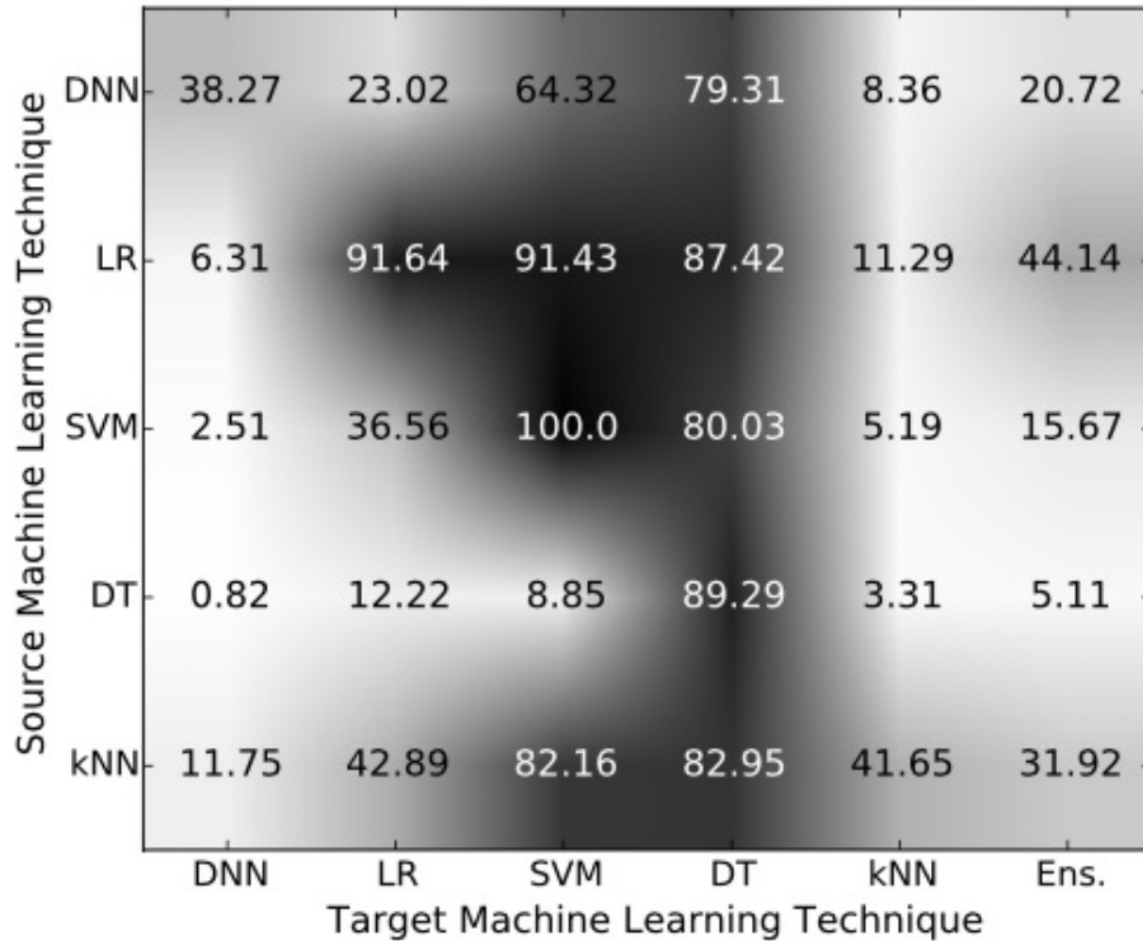


(e) DT models



(f) kNN models

Results: Cross-Model



Generating a Substitute Dataset

Methods

Jacobian Based
Image Augmentation

Periodic Step Size

Reservoir Sampling

Jacobian-Based Augmentation

Algorithm 1 Jacobian-based augmentation with Reservoir Sampling: sets are considered as arrays for ease of notation.

Input: $S_{\rho-1}, \kappa, J_f, \lambda_\rho$

```
1:  $N \leftarrow |S_{\rho-1}|$ 
2: Initialize  $S_\rho$  as array of  $N + \kappa$  items
3:  $S_\rho[0 : N - 1] \leftarrow S_{\rho-1}$ 
4: for  $i \in 0..\kappa - 1$  do
5:    $S_\rho[N + i] \leftarrow S_{\rho-1}[i] + \lambda_\rho \cdot \text{sgn}(J_f[\tilde{O}(S_{\rho-1}[i])])$ 
6: end for
7: for  $i \in \kappa..N - 1$  do
8:    $r \leftarrow$  random integer between 0 and  $i$ 
9:   if  $r < \kappa$  then
10:     $S_\rho[N + r] \leftarrow S_{\rho-1}[i] + \lambda_\rho \cdot \text{sgn}(J_f[\tilde{O}(S_{\rho-1}[i])])$ 
11:   end if
12: end for
13: return  $S_\rho$ 
```

Attacks Against Cloud Services

Substitute type	DNN	LR
$\rho = 3$ (800 queries)	87.44%	96.19%
$\rho = 6$ (6,400 queries)	96.78 %	96.43%
$\rho = 6$ (PSS + RS) (2,000 queries)	95.68%	95.83%

Table 3: Misclassification rates of the Amazon oracle on adversarial samples ($\varepsilon = 0.3$) produced with DNN and LR substitutes after $\rho = \{3, 6\}$ augmentation iterations. Substitutes are trained without and with refinements from Section 4: periodic step size (PSS) and reservoir sampling (RS).

Substitute type	DNN	LR
$\rho = 3$ (800 queries)	84.50%	88.94%
$\rho = 6$ (6,400 queries)	97.17%	92.05%
$\rho = 6$ (PSS + RS) (2,000 queries)	91.57%	97.72%

Table 4: Misclassification rates of the Google oracle on adversarial samples ($\varepsilon = 0.3$) produced with DNN and LR substitutes after $\rho = \{3, 6\}$ augmentation iterations.. Substitutes are trained without and with refinements from Section 4: periodic step size (PSS) and reservoir sampling (RS).

Strengths

- Consider transferability across different types of models and also for the same model type
- Techniques for data augmentation and sampling help build better substitutes
 - DNN and LR result in better substitutes
- Attack Amazon and Google services in black-box manner

Limitations

- Adversarial examples do not transfer to decision trees, but an explanation is missing
 - Most likely, the decision boundaries of LR, DNN, and SVM are similar, but that of DT is very different
- Single dataset (MNIST)

Discussion Points

- How to build substitute models with small number of queries
- How to protect MLaaS cloud services
- Do adversarial examples for complex models transfer?
 - If model overfits to training data, less likely to transfer
 - Smoother models work better as substitutes (LR)
 - Results also depend on the task complexity