Aalto University

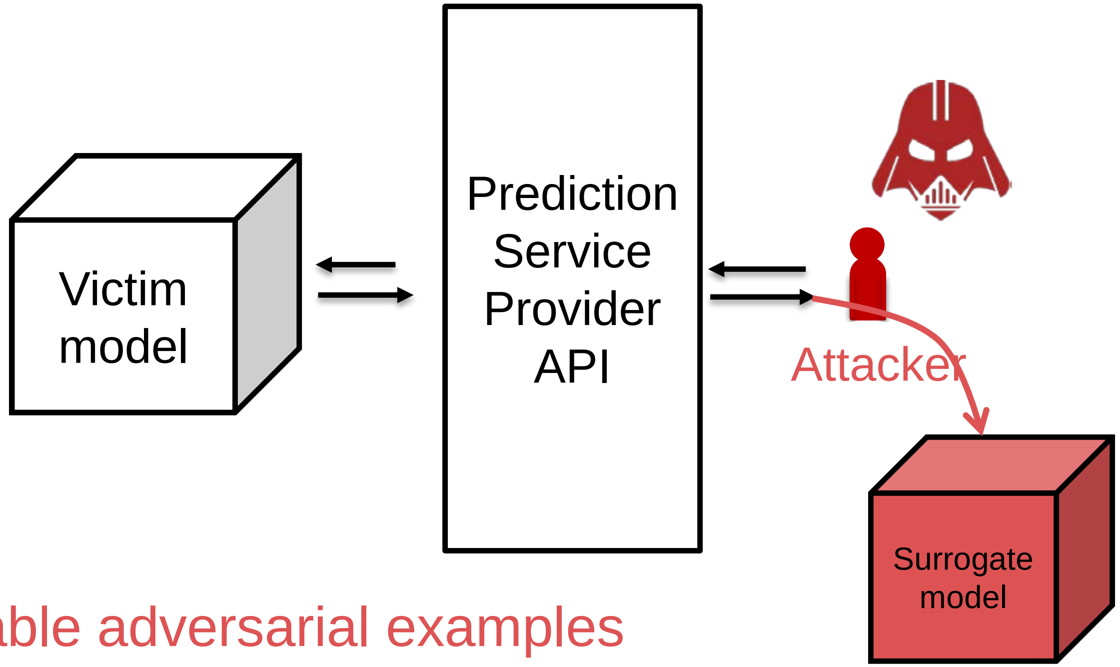
PRADA: Protecting against DNN model stealing attacks

Mika Juuti, Sebastian Szyller, Samuel Marchal, N. Asokan
IEEE Euro S&P 2019, Sweden, Stockholm, June 19 2019

Background

Machine learning increasingly popular: business advantage to companies

- API: black-box access to clients
- Automate tedious decision-making



Attacker wants to compromise

- Model confidentiality ~ model extraction
- Model integrity (prediction quality) ~ transferable adversarial examples

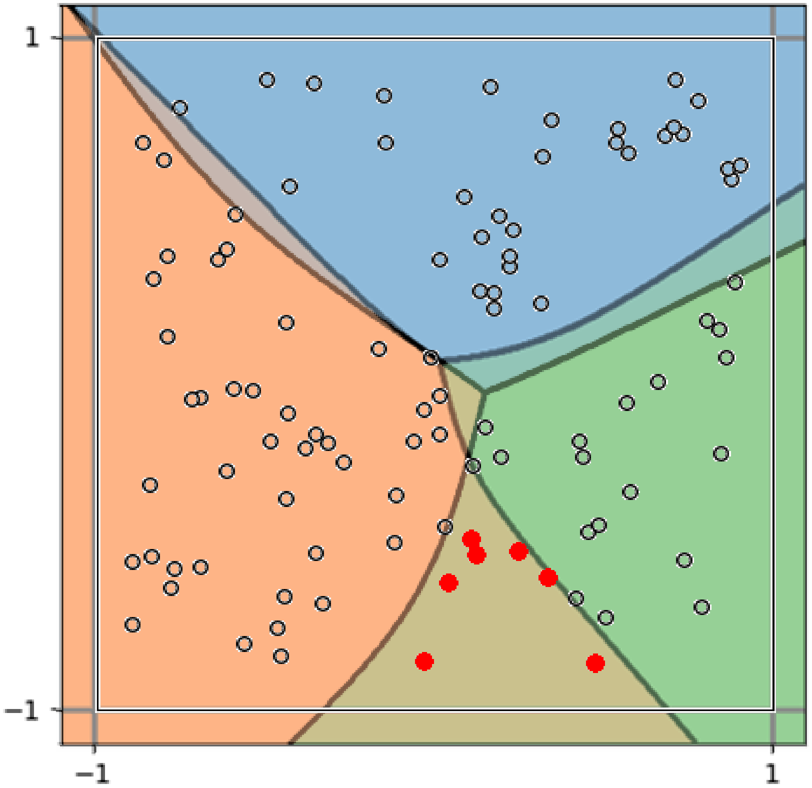
[1] Tramer et al. *Stealing ML models via prediction APIs*. UsenixSEC'16.

How to measure extraction success?

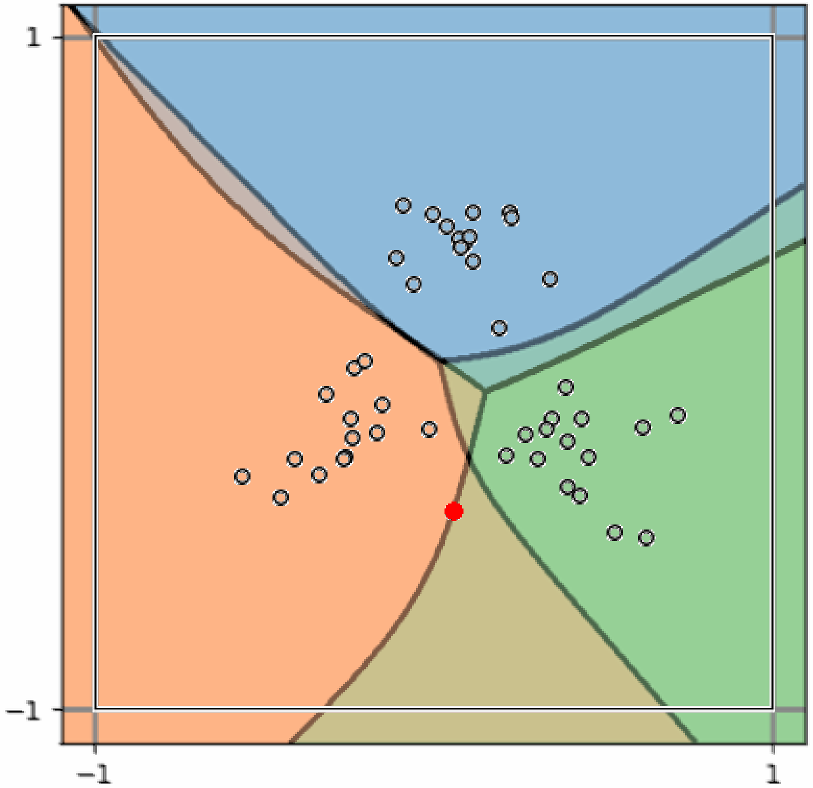
Does attacker’s surrogate model produce similar predictions as victim model?

How to measure extraction success?

Does attacker's surrogate model produce similar predictions as victim model?



Random Uniform Agreement [1]

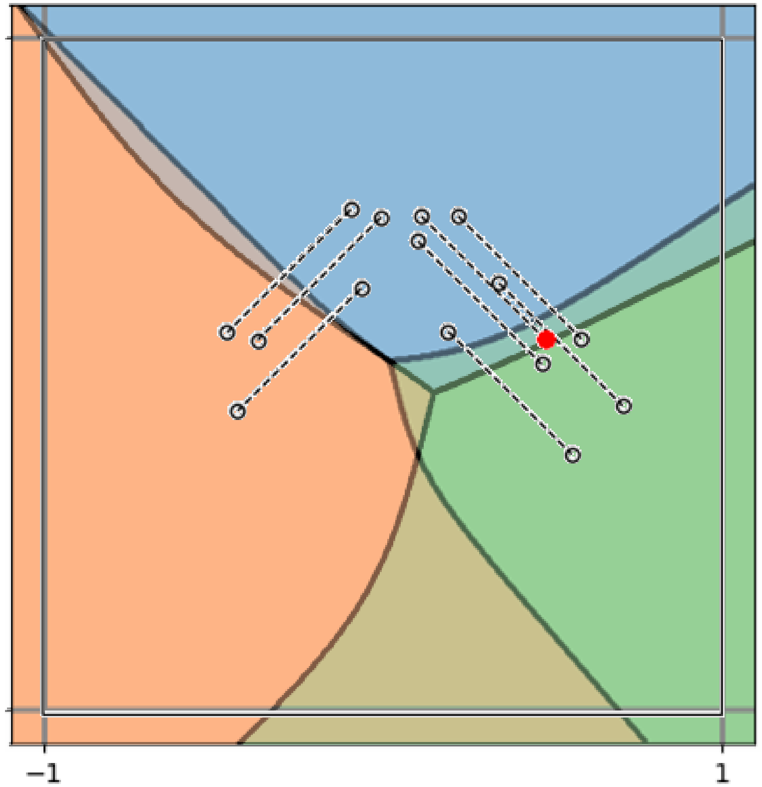


Test Agreement [1]

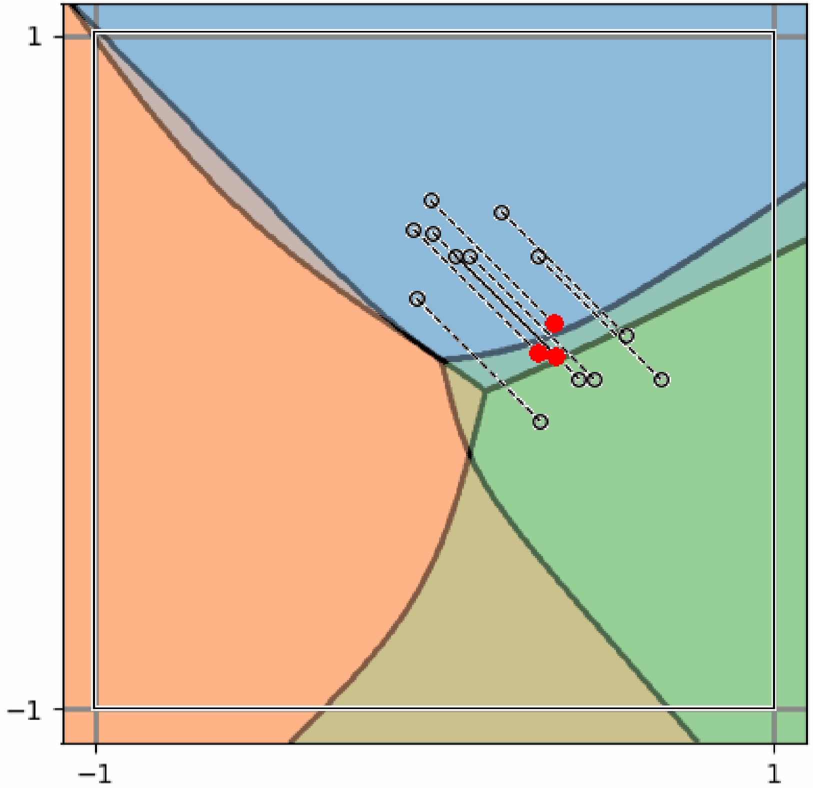
[1] Tramer et al. *Stealing ML models via prediction APIs*. UsenixSEC'16.

Transferable adversarial examples

Do adversarial examples created with surrogate model transfer to victim model?



Non-targeted transferability [2]
Target class: any other



Targeted transferability
Target class: specified other

[2] Papernot et al. *Practical black-box attacks against machine learning*. AsiaCCS'17.

DNN model extraction framework

Algorithm 1 Model extraction process with the goal of extracting classifier F, given initial unlabeled seed samples X and a substitute model F' (initially random).

procedure $\textsc{ExtractModel}(F)$
    $U \leftarrow$ *Initial data collection*
    $L \leftarrow \{U,\ \textsc{Label}(U, F)\}$
    $F' \leftarrow$ *Select architecture*
    $H \leftarrow$ *Resolve hyperparameters*   ▷ cf. Sec. III-A
    $F' \leftarrow \textsc{Initialize}(F')$ ▷ Set random weights
    $F' \leftarrow \textsc{Train}(F' \mid L, H)$
    for $i \leftarrow 1, \rho$ **do**   ▷ ρ *duplication rounds*
        $U \leftarrow$ *Create synthetic samples*   ▷ cf. Sec. III-C
        $L \leftarrow \{\ L \cup \{U,\ \textsc{Label}(U, F)\}\ \}$
        $F' \leftarrow \textsc{Train}(F' \mid L, H)$
    end for
    return F'
end procedure

[1] Tramer et al. *Stealing ML models via prediction APIs*. UsenixSEC'16.
[2] Papernot et al. *Practical black-box attacks against machine learning*. AsiaCCS'17.

Hyper-parameter determination

1. Hand-picked [2]
   - Need re-adjustments for new datasets

Algorithm 1 Model extraction process with the goal of extracting classifier F, given initial unlabeled seed samples X and a substitute model F' (initially random).

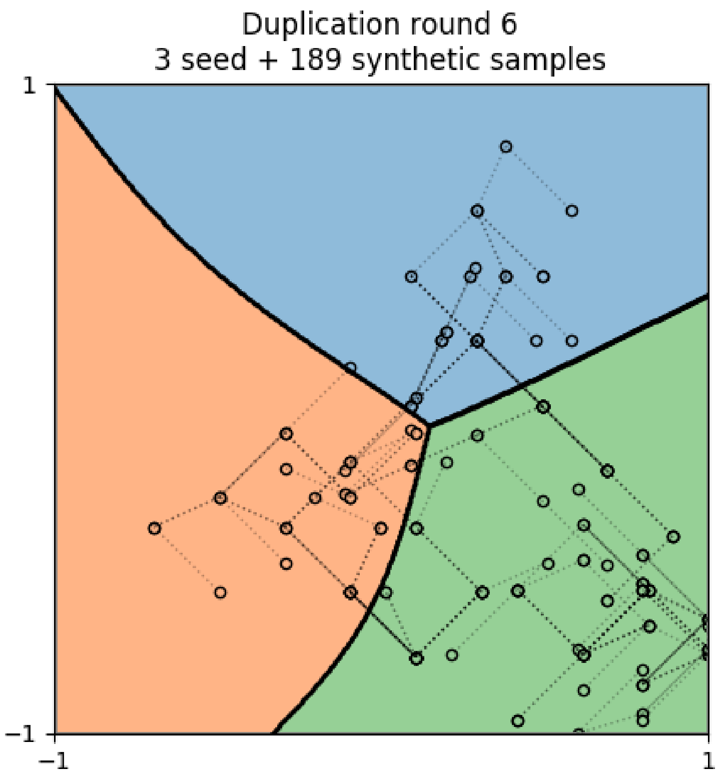
procedure ExtractModel(F)
    $U \leftarrow$ *Initial data collection*
    $L \leftarrow \{U, \text{Label}(U, F)\}$
    $F' \leftarrow$ *Select architecture*
    $\boxed{H \leftarrow \textit{Resolve hyperparameters}}$ ▷ cf. Sec. $\boxed{\text{III-A}}$

17	00m07s	0.96000	4.9894	-3.5161
18	00m03s	0.88000	2.8593	-2.7311
19	00m09s	0.94000	5.3715	-3.1127
20	00m04s	0.80000	3.6854	-2.0000
21	00m07s	0.86000	5.0527	-4.0000
22	00m08s	0.92000	4.9484	-3.1413
23	00m13s	0.93000	5.7683	-2.6766
24	00m09s	0.94000	5.2931	-3.5669
25	00m05s	0.94000	4.1546	-2.7843
26	00m06s	0.92000	4.5602	-3.5012
27	00m11s	0.94000	5.4090	-2.6179
28	00m06s	0.92000	4.1068	-2.5207
29	00m13s	0.94000	5.6754	-2.9973
30	00m08s	0.91000	4.9028	-3.6115

```
Best learning rate: 0.000305
Best number of epochs: 147
CV-Search took 3.164177 minutes
```

[1] Tramer et al. *Stealing ML models via prediction APIs*. UsenixSEC'16.
[2] Papernot et al. *Practical black-box attacks against machine learning*. A

Synthetic samples



Algorithm 1 Model extraction process with the goal of extracting classifier F, given initial unlabeled seed samples X and a substitute model F' (initially random).

---

procedure $\textsc{ExtractModel}(F)$
    $U \leftarrow$ *Initial data collection*
    $L \leftarrow \{U,\ \textsc{Label}(U, F)\}$
    $F' \leftarrow$ *Select architecture*
    $H \leftarrow$ *Resolve hyperparameters* ▷ cf. Sec. III-A
    $F' \leftarrow \textsc{Initialize}(F')$ ▷ Set random weights
    $F' \leftarrow \textsc{Train}(F' \mid L, H)$
    for $i \leftarrow 1, \rho$ **do** ▷ ρ *duplication rounds*
        $U \leftarrow$ *Create synthetic samples* ▷ cf. Sec. III-C
        $L \leftarrow \{\ L \cup \{U,\ \textsc{Label}(U, F)\}\ \}$
        $F' \leftarrow \textsc{Train}(F' \mid L, H)$
    end for
    return F'
end procedure

Approaches for DNN model stealing

Tramer [1]

Seeds: very many random points

Line search + query plausible boundary

Purpose: RU-Agreement, Test-Agreement

Hyperparameters: Same

~100,000 queries

Papernot [2]

Seeds: few natural samples (~10 per class)

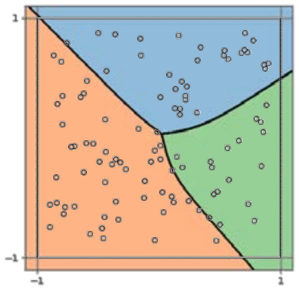
Iteratively: train substitute + query adv. ex.

Purpose: Non-targeted transferability

Hyperparameters: hand-picked

Training: 10 epochs (very short!)

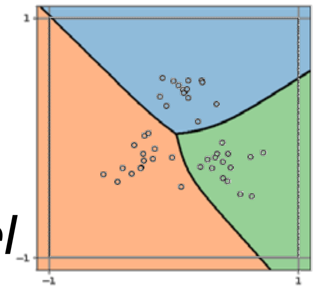
~6,400 queries



Both:

From few or no natural samples to thousands of synthetic samples

Initial random model → refined model

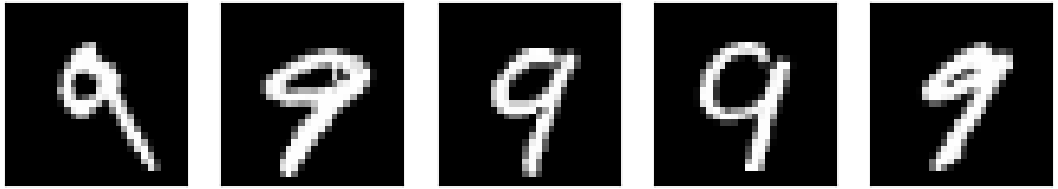


[1] Tramer et al. *Stealing ML models via prediction APIs*. UsenixSEC'16.
[2] Papernot et al. *Practical black-box attacks against machine learning*. AsiaCCS'17.

Datasets

MNIST: B&W Digits
10 classes

Victim DNN: trained with 55,000 images
4 layers (2 conv + 2 dense)
~500,000 parameters



GTSRB: Traffic Sign Recognition
43 classes

Victim DNN: trained with 39,000 images
5 layers (2 conv + 3 dense)
~700,000 parameters

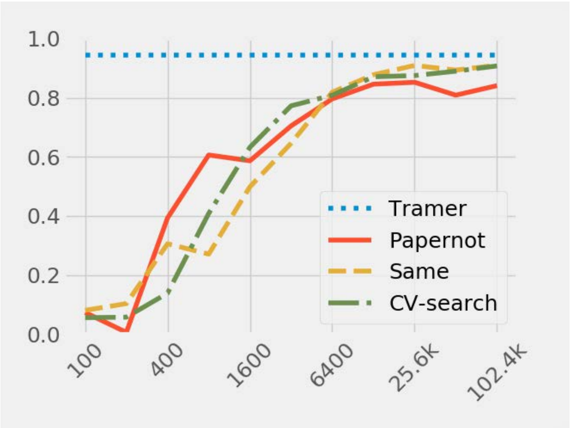
Preliminary attack on MNIST

Comparative evaluation:

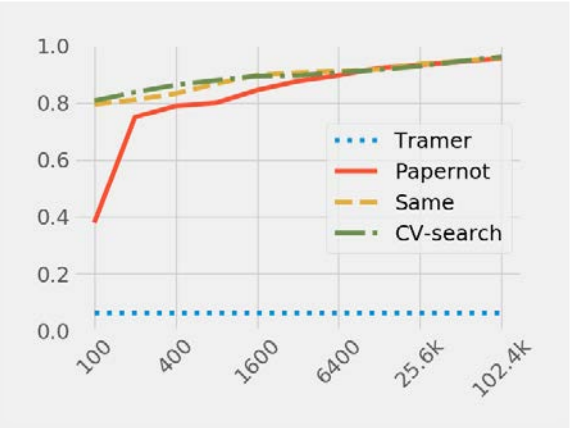
- Initially: up to 100 natural samples
- Stops after 102,400 queries sent
- All four success criteria evaluated
- Transferability: FGSM $\epsilon = 25\%$, as in [2]

- Tramer [1] ineffective on DNNs
  - Networks here $250 \times$ bigger than in [1]
- Papernot[2] better. Why short training?
- No benefit from short training.
- Papernot with CV-Search superior
  - Why not done before?

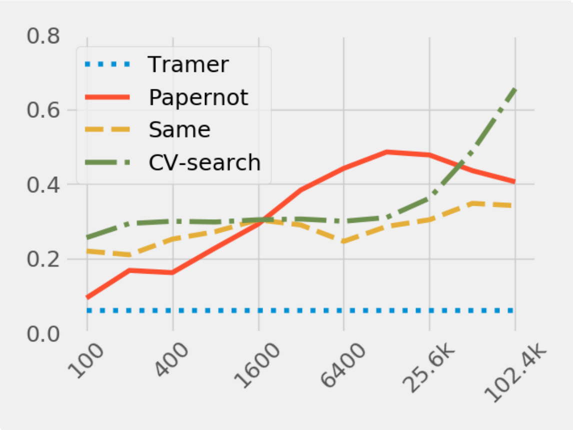
RU Agreement



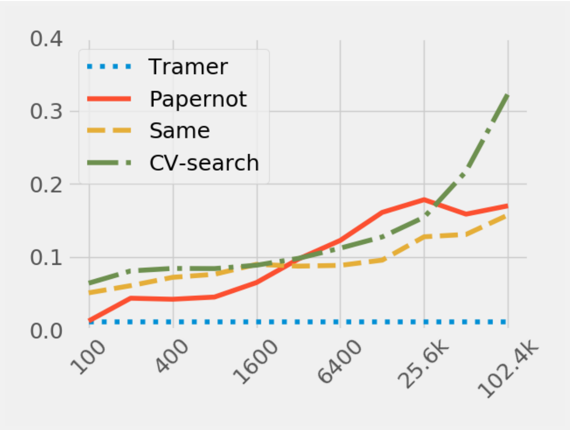
Test Agreement



Non-targeted transf.



Targeted transf.



[1] Tramer et al. *Stealing ML models via prediction APIs*. UsenixSEC'16.
[2] Papernot et al. *Practical black-box attacks against machine learning*. AsiaCCS'17.

Comparative evaluation with state-of-the-art

MNIST	Tramer [1]	Papernot [2]	Ours	Improvement
Test Agreement	< 7%	95.1%	97.9%	$1.03\times$
Targeted Transferability	1%	10.6%	39.3%	$3.70\times$

GTSRB	Tramer [1]	Papernot [2]	Ours	Improvement
Test Agreement	< 1%	16.9%	62.5%	$3.70\times$
Targeted Transferability	2%	41.1%	84.4%	$2.05\times$

Top-5 agreement: 47% Top-5 agreement: 92%

[1] Tramer et al. *Stealing ML models via prediction APIs*. UsenixSEC'16.
[2] Papernot et al. *Practical black-box attacks against machine learning*. AsiaCCS'17.

What makes our attacks better?

	MNIST		GTSRB	
	Agree.	Targeted	Agree.	Targeted
Baseline: Papernot	95.1%	10.6	16.9%	41.1%
Our attacks	97.9%	39.3%	62.5%	84.4%

```
procedure EXTRACTMODEL(F)
    U ← Initial data collection
    L ← {U, LABEL(U, F)}
    F' ← Select architecture
    H ← Resolve hyperparameters                ▷ cf. Sec. III-A
    F' ← INITIALIZE(F')                        ▷ Set random weights
    F' ← TRAIN(F' | L, H)
    for i ← 1, ρ do                            ▷ ρ duplication rounds
        U ← Create synthetic samples           ▷ cf. Sec. III-C
        L ← { L ∪ {U, LABEL(U, F)} }
        F' ← TRAIN(F' | L, H)
    end for
    return F'
end procedure
```

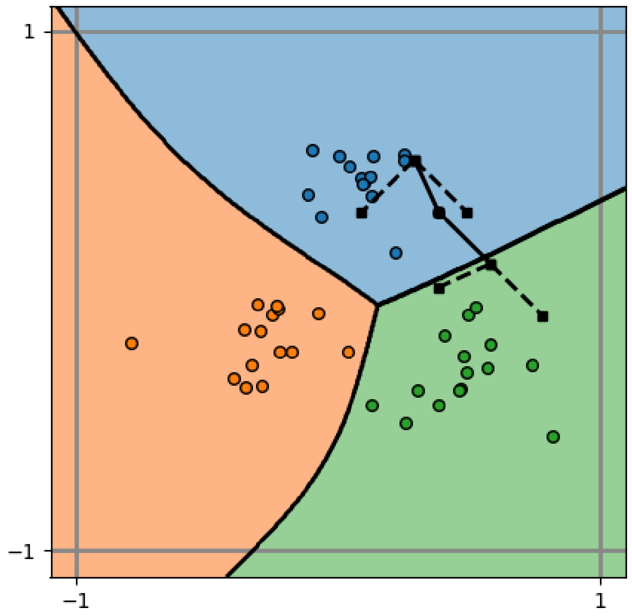
More in paper!

All attacks: Common characteristics

Specific pattern in attacks:

1. Natural/random samples
- Establish initial decision boundaries

2. Synthetic samples ~ similar to existing samples
- Refine the boundaries



Study **distribution of queries** to detect model extraction attacks

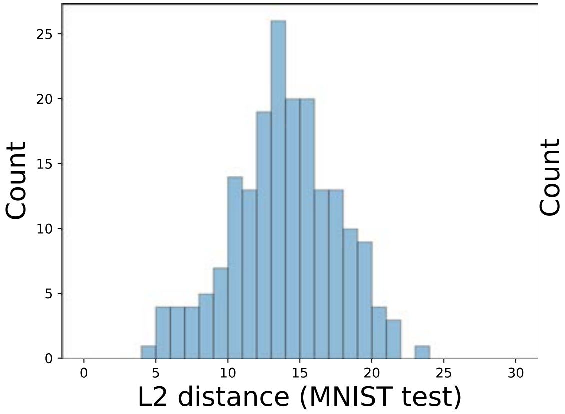
Intuition for a defense

Preliminary: distance between random points in a space fits a normal (Gaussian) distribution

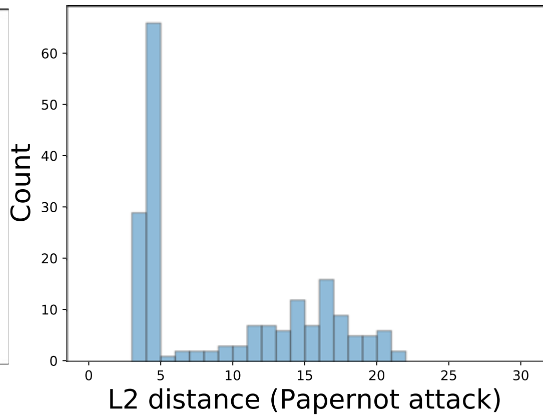
Assumptions

- Benign queries consistently distributed → distances fit a normal distribution
- Adversarial queries focused on a few areas → distances deviate from a normal distribution

Benign

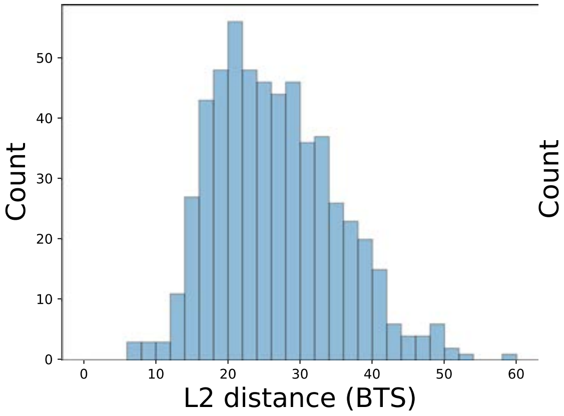


Attack

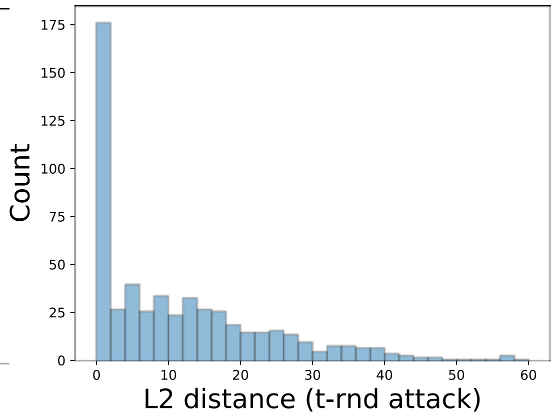


MNIST

Benign



Attack



GTSRB

Proposed defense

Stateful defense

- Focus on low false positives
- Keeps track of queries submitted by a given client
- Detects deviation from a normal distribution

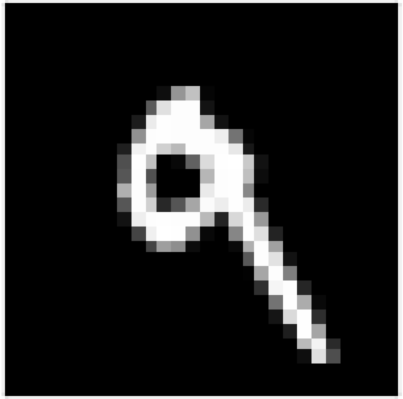
Shapiro-Wilk test

- Quantify how well a set of samples D fits a normal distribution
- Test statistic: $W(D) < \delta \rightarrow$ attack detected
- δ**:** parameter to be defined

Benign data

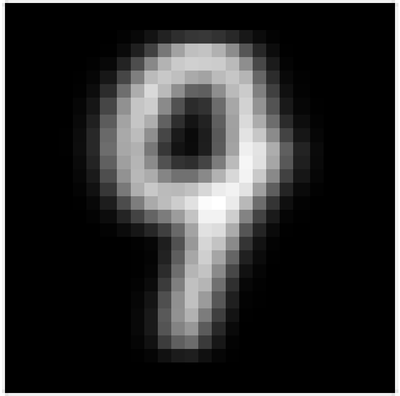
Simulate legitimate queries

- Random same distribution (MNIST/German)
- Random different distribution (USPS/Belgian)
- Uniformly random images
- Sequence of images (207x30 images German)



MNIST

German



USPS

Belgian

Detection efficiency

Model + δ value	FPR	Queries made until detection		
		Tramer	*Papernot*	*T-rnd*
MNIST ($\delta = 0.96$)	0.0%	5,560	120	130
MNIST ($\delta = 0.95$)	0.0%	5,560	120	140
GTRSB ($\delta = 0.90$)	0.6%	5,020	430	500
GTRSB ($\delta = 0.87$)	0.0%	5,020	430	540

- All prior model extraction attacks detected
- Detection triggered when synthetic samples queried

- Slowest on Tramer ~ ineffective on DNNs
  - Requires $\gg$ 500k queries to succeed [1]

[1] (Optimistic estimate based on) Tramer et al. *Stealing ML models via prediction APIs*. UsenixSEC'16.
[2] Papernot et al. *Practical black-box attacks against machine learning*. AsiaCCS'17.

Summary

Attack with 10 *natural* samples per class + 100 000 *synthetic* queries

- Strong attacks on MNIST (98% agreement) and GTSRB (92% top-5 agreement)

Takeaways:

- Hyperparameter protection unhelpful:
  - Attacker's CV-Search for learning rate / epochs yields more effective attack
- API response granularity has little effect:
  - Returning all probabilities / top label yield same performance for agreement
- Using more complex model for theft useful to reach better attack performance
  - But any mismatch in models yields worse transferability → model confidentiality can help
- Natural data is better than synthetic data → use as much as possible
- Defenses plausible, but robust detection still an open problem

We share code with *bona fide* researchers. Thank you!

Aalto University

PRADA: Protecting against DNN model stealing attacks

Mika Juuti, Sebastian Szyller, Samuel Marchal, N. Asokan
IEEE Euro S&P 2019, Sweden, Stockholm, June 19 2019

Different victim/surrogate architectures

Effect on test agreement:

Diagonal: victim/surrogate with same complexity

Beneficial for adversary to use more complex model architecture

Detrimental for adversary to use lower-complexity surrogate models