David J. Miller
Zhen Xiang
George Kesidis

Adversarial Learning and Secure AI

CAMBRIDGE
UNIVERSITY PRESS & ASSESSMENT

Chapter 14

Reverse Engineering Attacks (REAs) on Classifiers

Reverse Engineering Attack

- [Tramer et al. ’16] show that one can reverse-engineer a “black box” classifier (training a classifier that closely mimics the black box’s decision-making) **without any knowledge of training data for the domain**.
- Their approach requires numerous (random) queries to (probes of) the black box.
- Application: circumventing paying for, e.g., Google’s ML service, once its black box classifier has been reverse-engineered.
- **Professed advantage:** their approach does not require **any** knowledge of the class distributions or training data for the given domain (these may be unknown or expensive to obtain).

Reverse Engineering Attack – Disadvantages of Tramer'16

- To be most effective, their approach requires knowledge of the type of classifier being used as black box (but not its parameter values).
- They do not consider DNNs and true big data domains (with huge feature spaces) – even for low dim., shallow NN classification, their approach requires ~ tens of thousands of queries to achieve good accuracy.
- On huge domains, with DNNs, orders of magnitude more queries may be needed (and paid for), with the economic payoff of the attack now uncertain.
- Not considering DNNs and big data domains is a big omission, i.e., main reasons for using an ML service would be
  - lack of computing resources to train the model and/or
  - lack of training data (which may be precious on some domains).

Reverse Engineering Attack – Tables 5 & 6 of Tramer'16

Data set	# records	# classes	# features
IRS Tax Patterns	191,283	51	31
Steak Survey	430	5	12
GSS Survey	51,020	3	7
Email Importance	4,709	2	14
Email Spam	4,601	2	46
German Credit	1,000	2	11
Medical Cover	163,065	$\mathcal{Y} = \mathbb{R}$	13
Bitcoin Price	1,076	$\mathcal{Y} = \mathbb{R}$	7

Table 5: **Data sets used for decision tree extraction.** Trained trees for these data sets are available in BigML's public gallery. The last two data sets are used to train regression trees.

Small feature dimensions, but large numbers of queries

				Without incomplete queries			With incomplete queries		
Model	**Leaves**	**Unique IDs**	**Depth**	$1 - R_{\text{test}}$	$1 - R_{\text{unif}}$	Queries	$1 - R_{\text{test}}$	$1 - R_{\text{unif}}$	Queries
IRS Tax Patterns	318	318	8	100.00%	100.00%	101,057	100.00%	100.00%	29,609
Steak Survey	193	28	17	92.45%	86.40%	3,652	100.00%	100.00%	4,013
GSS Survey	159	113	8	99.98%	99.61%	7,434	100.00%	99.65%	2,752
Email Importance	109	55	17	99.13%	99.90%	12,888	99.81%	99.99%	4,081
Email Spam	219	78	29	87.20%	100.00%	42,324	99.70%	100.00%	21,808
German Credit	26	25	11	100.00%	100.00%	1,722	100.00%	100.00%	1,150
Medical Cover	49	49	11	100.00%	100.00%	5,966	100.00%	100.00%	1,788
Bitcoin Price	155	155	9	100.00%	100.00%	31,956	100.00%	100.00%	7,390

Table 6: **Performance of extraction attacks on public models from BigML.** For each model, we report the number of leaves in the tree, the number of unique identifiers for those leaves, and the maximal tree depth. The chosen granularity ε for continuous features is 10^{-3}.

Reverse Engineering Attack – Biggest Weakness

- But each random query is very likely an extreme outlier of **every class**.
- Even a few such queries, let alone thousands of them, will be highly suspicious ⇒ **the ML service can refuse to accept further queries from this user, thus defeating the attack!**
- Even if the attacker uses many bots, AD will likely detect the attack by each bot, after a few queries…

Defense against REAs using ADA (an OODD for TTEs, see Chapter 4)

- Since in RE attacks the attacker submits batches of query samples to the classifier, we modify ADA to jointly exploit batches of samples in seeking to detect attacks (in this case RE query attacks, not TTE attacks).
- Several schemes for aggregating ADA decision statistics, produced for individual samples in a batch, are investigated in [Y. Wang et al. '19]:
  I. arithmetically averaging the ADA statistic over all samples in a batch;
  II. maximizing the ADA statistic over all samples in a batch;
  III. dividing a batch into mini-batches, for example a batch of say 50 samples could be divided into mini-batches of size 5.
- For each mini-batch, apply either scheme I) or II).
- Then, make a detection if any of the mini-batches yields a detection statistic greater than the threshold (union rule).

Experimental Results

- This last aggregation scheme is experimentally seen to perform best for a LENET-5 classifier of MNIST images…

Figure 1

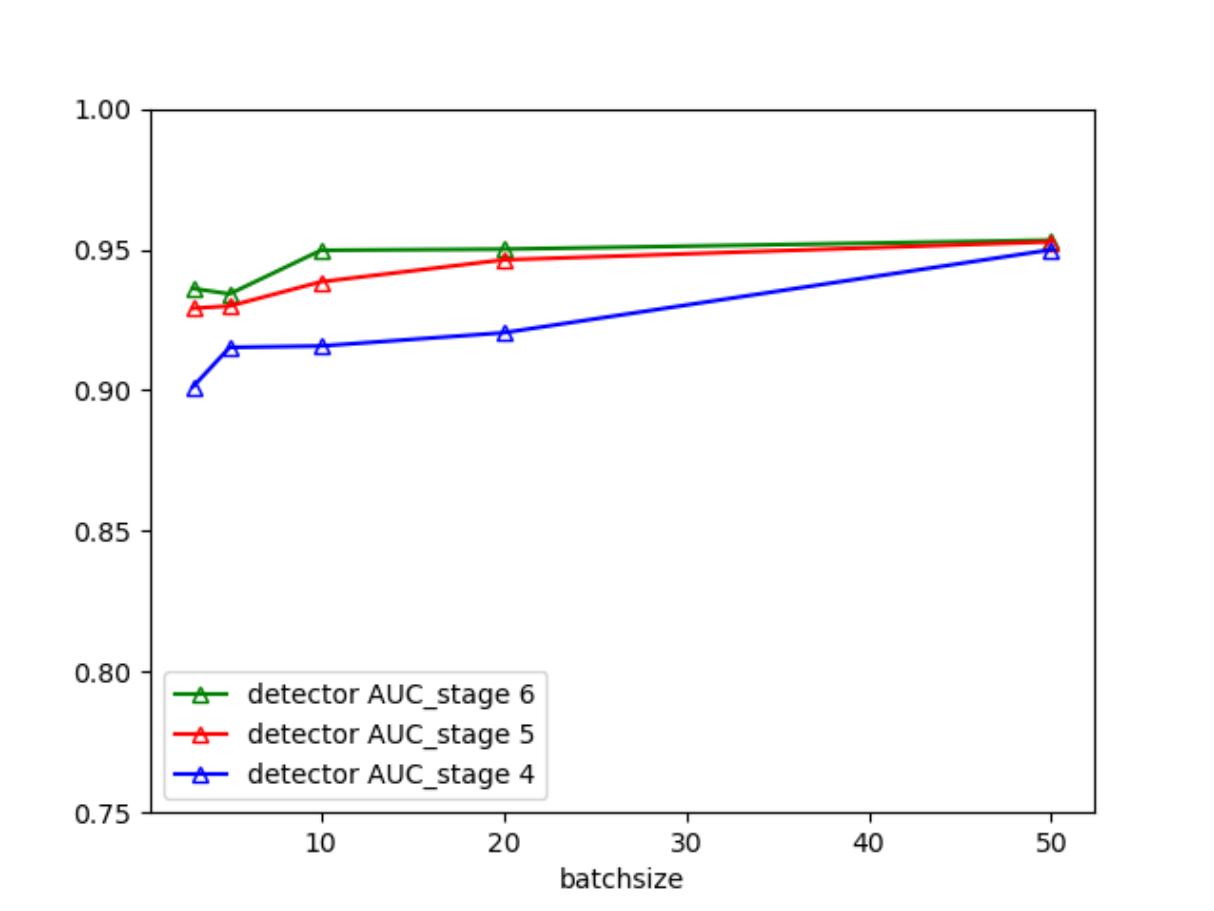


Figure 2

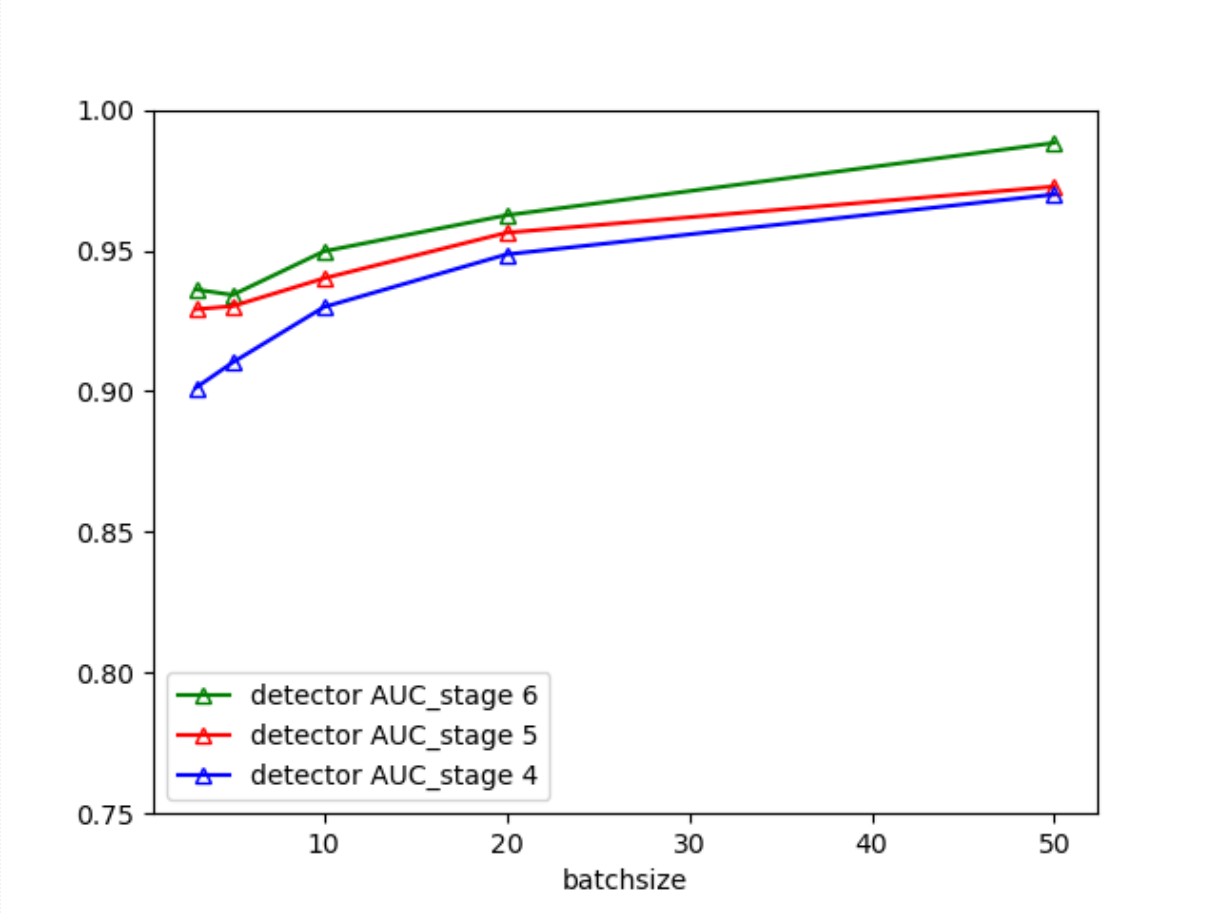
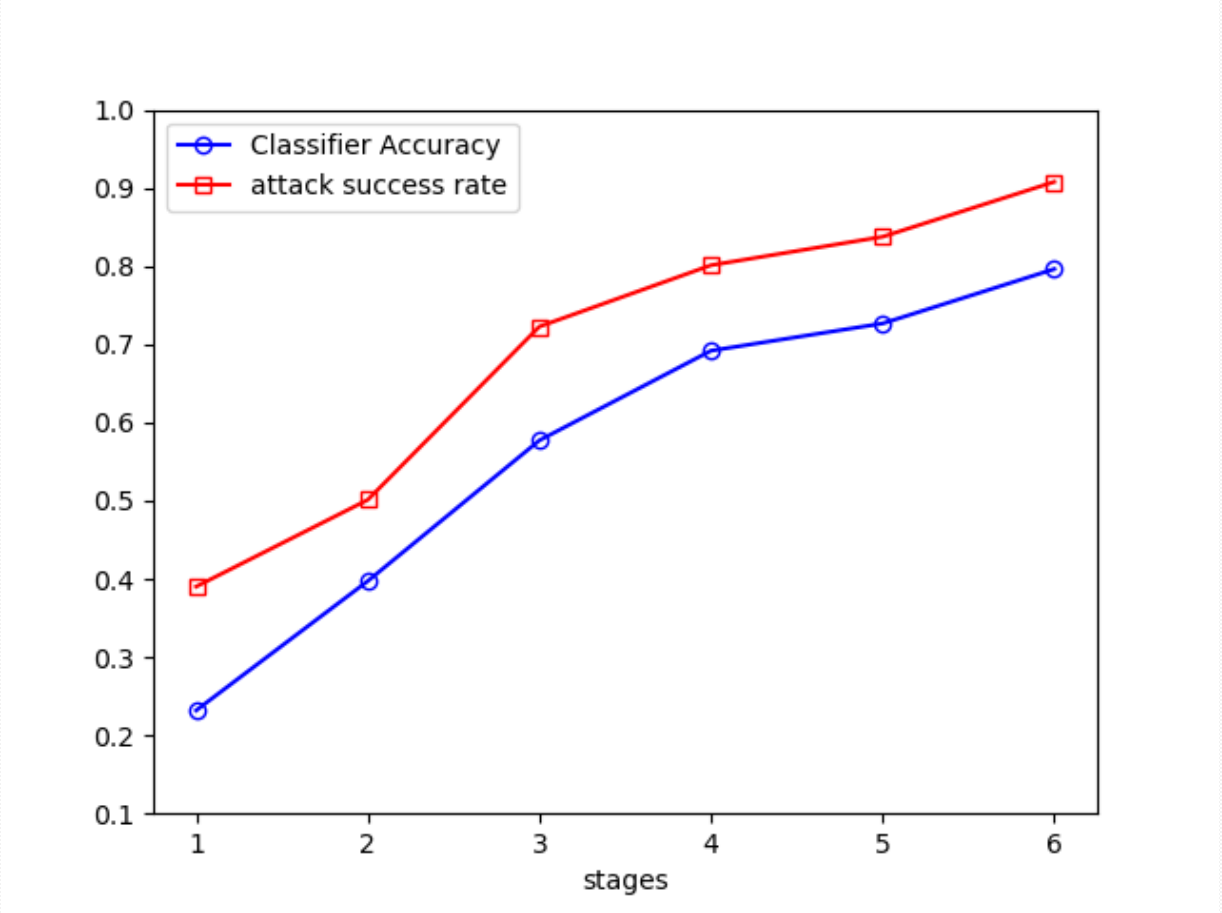


Figure 3



With Permission, Figures Reproduced From

- Y. Wang, D.J. Miller, G. Kesidis. When Not to Classify: Detection of Reverse Engineering Attacks on DNN Image Classifiers. In *Proc. IEEE ICASSP*, Brighton, UK, May 2019.