WHEN NOT TO CLASSIFY: DETECTION OF REVERSE ENGINEERING ATTACKS ON DNN IMAGE CLASSIFIERS

Yujia Wang, David J. Miller, George Kesidis

School of EECS, The Pennsylvania State University, University Park, PA 16802
{yjwang,djm25,gik2}@psu.edu

ABSTRACT

This paper addresses detection of a reverse engineering (RE) attack targeting a deep neural network (DNN) image classifier; by querying, RE's aim is to discover the classifier's decision rule. RE can enable test-time evasion attacks, which require knowledge of the classifier. Recently, we proposed a quite effective approach (ADA) to detect test-time evasion attacks. In this paper, we extend ADA to detect RE attacks (ADA-RE). We demonstrate our method is successful in detecting "stealthy" RE attacks before they learn enough to launch effective test-time evasion attacks.

1. INTRODUCTION

Recently, there has been great interest in identifying vulnerabilities in machine learning (ML)) systems. *Test-time evasion attacks* (TTEAs) [3, 4, 5, 6, 7, 8] add subtle perturbations to legitimate test-time samples[1] to "fool" a classifier into making incorrect decisions relative to those of a human being. Related work has demonstrated the fragility of DNNs for some domains in the presence of modest data perturbations, e.g. changing the tempo in music genre classification [12]. TTEAs should be taken seriously because they could allow illegitimate access to a building, data, or a piece of machinery. They could also lead e.g. to a radiologist looking at "doctored" cancer biopsy images (that trick automated prescreening systems). Test-time attacks require knowledge of the classifier under attack. RE attacks [10, 11] involve querying a classifier to *discover* its decision rule. Thus, one primary application of RE is to *enable* TTEAs.

Several recent RE attack works are [10] and [11]. [10] demonstrates that, with a *relatively* modest number of queries (perhaps $\sim$ ten thousand), using the classifier's answers on query examples as supervising ground truth labels, one can learn a surrogate classifier on a given domain that closely mimics an unknown (black box) classifier. One weakness of [10] is that it neither considers very large (feature space) domains nor very large networks (DNNs) – orders of magnitude more queries may be needed to reverse-engineer a DNN on a large-scale domain. However, a much more critical weakness stems from one of the greatest purported advantages in [10] – the authors emphasize their RE does not require *any* actual samples from the domain[2]. Their queries are *randomly* drawn, *e.g.* uniformly, over the given feature space. What was not recognized in [10] is that this random querying makes the attack *easily detectable* – randomly selected query patterns will typically look nothing like legitimate examples from any of the classes – they are very likely to be extreme outliers, of all the classes. Each such query is thus *individually* highly suspicious – thus, even ten, let alone ten thousand such queries will be trivially anomaly-detected as jointly improbable under a null distribution (estimable from the training set defined over all the classes from the domain). Even if the attacker employed bots, each of which makes a small number of queries, each bot's random queries should be easily detected as anomalous, likely associated with an RE attack. On the other hand, [11] propose an RE attack that does require some initial known data from the domain. It uses this to create more legitimate, "stealthier" queries, over a series of query stages, with the resulting labeled data used to train a substitute classifier used to launch a TTEA.

Recently, an approach was developed which achieves state-of-the-art results in detection of TTEAs, Anomaly Detection of Attacks (ADA) [7, 8]. Since this approach is an anomaly detector for the image domain of interest, it in principle should also be applicable to detect query images that are not representative of real images from the domain. However, since the querying in [11] is stealthy (as it is based on perturbations of real images from the domain), it is not obvious their querying is detectable. However, here we extend ADA to indeed detect the RE querying from [11], and thus demonstrate the potential to prevent TTEAs even before they are launched.

This paper is organized as follows. In Sec.2, we describe the reverse engineering attack of [11]. In Sec.3, we give background on ADA. In Sec.4, we discuss our extension of ADA for reverse engineering attacks. Experimental results

This paper is dedicated to the memory of our dear friend Jan Larsen. This research is supported by AFOSR DDDAS and Cisco Systems URP.

[1] Perturbation approaches are related to boundary-finding algorithms (neural network inversion) [9].

[2] For certain sensitive domains, or ones where obtaining real examples is expensive, the attacker may not have access to legitimate examples.

for DNN classifiers of images are given in Sec.5. Finally, conclusions are drawn in Sec.6.

2. RE ATTACK GIVEN DOMAIN SAMPLES

The RE procedure in [11] is summarized as follows. First, the adversary collects a small set of representative labeled samples from the input domain as an initial training set S_0 and uses this to train an initial substitute classifier. Then, there is stagewise data collection and retraining, over a sequence of stages. In each, the adversary augments the current training set by querying the classifier with the stage's newly generated samples [11], i.e.,

$$S_{k+1} = \{\underline{x}+\lambda\cdot\text{sgn}(\nabla(\max_c P_s^{(k)}[C=c|\underline{x}])) : \underline{x} \in S_k\}\cup S_k$$

where k is the current stage index and $P_s^{(k)}[C=c|\underline{x}]$ is the current substitute class posterior model. The substitute classifier is then retrained using S_{k+1}. Each successive stage crafts query samples closer to the classifier's true boundary, which is helpful for RE learning but which also makes these samples less class-representative and thus more detectable. Once a sufficiently accurate substitute classifier is learned, the adversary can launch a TTEA using one of the existing TTEA attacks, e.g. [4, 3, 5]. Here, one starts with an original image from the domain, from a source class c_s, perturbs the image, using the substitute classifier's gradient information, to push across the decision boundary to a destination class, $c_d \neq c_s$. The perturbed image is then submitted to the actual classifier as a TTEA instance.

3. DETECTION OF TEST-TIME EVASIONS (ADA)

3.1. Basic ADA

ADA detection is grounded in the premise that an attack example in general will exhibit too much atypicality (evaluated on null distributions estimated from the class training sets) w.r.t. c_d and too little null atypicality w.r.t. c_s[3]. Given a test sample $\underline{x}$, basic ADA works as follows:

1. Determine the MAP (destination) class under the deep neural network: $c_d = \text{argmax}_c P[C=c|\underline{x}]$.
2. Compute $\underline{z} = \underline{g}_k(\underline{x})$, the vector of outputs from the k-th layer of the DNN.
3. Estimate the source class c_s based on the null model: $c_s = \text{argmax}_{c\neq c_d} f_{\underline{Z}(k)|c}(\underline{z})$.
4. Form two probability vectors $P(k)$ and $Q(k)$ where $P(k) = \{p_0 f_{\underline{Z}(k)|c_d}(\underline{z}), p_0 f_{\underline{Z}(k)|c_s}(\underline{z})\}$ and $Q(k) = \{q_0 P[C = c_d|\underline{x}], q_0 P[C = c_s|\underline{x}]\}$. p_0 and q_0 are normalizers to make $P(k)$ and $Q(k)$ probability mass functions.
5. Report a detection if $D_{\text{KL}}(P(k)||Q(k)) > t$ where $D_{\text{KL}}(\cdot||\cdot)$ is the Kullback Leibler(KL) Divergence. The KL distance will be large when $\underline{x}$ exhibits atypicality w.r.t. the null of c_d and typicality w.r.t. the null of c_s.

[3]This premise is plausible because the attacker tries to be stealthy – to fool the classifier while not fooling a human being (or an anomaly detector). In so doing, the perturbed image, while classified to c_d, still has to "look" like it comes from c_s.

3.2. Ultimate ADA Method Development: L-AWA-maxKL

The ultimate ADA method is based on the following extensions/improvements.

Maximizing KL distance over multiple layers: Rather than measure KL distance at one layer, we can compute KL distance at different layers and detect based on the *maximum* KL distance over these layers.

Null modelling for Different Neuron Activations: It was demonstrated that Gaussian mixture modelling is suitable for sigmoidal and linear layers, with log-Gaussian mixture modelling appropriate for RELU layers[7, 8].

Exploiting source class uncertainty and class confusion: Basic ADA hard-estimates c_s. More information is preserved if one reflects source class uncertainty, via the probabilities

$$\text{P}[C_s = c] = \frac{f_{\underline{Z}|c}(\underline{z})}{\sum_{c'\neq c_d} f_{\underline{Z}|c'}(\underline{z})} \quad \forall c \neq c_d.$$

Going further, if we have knowledge of the class confusion matrix $[\text{P}[C^* = i|C = j]]$, then a tuple (c_s, c_d) with a small class confusion probability $\text{P}[C^* = c_d|C = c_s]$ may indicate an attack. As a result, we weight KL distance by $\frac{1}{\text{P}[c_d|c_s]}$. This increases the decision statistic for those pairs that are unlikely to occur. Combining both techniques, we construct an average weighted ADA decision statistic via

$$\sum_{c\neq c_d} \text{P}[C=c]\frac{D_{\text{KL}}(P^{(c)}||Q^{(c)})}{\text{P}[C^*=c_d|C=c]}.$$

We can evaluate this statistic at different layers and then apply a max rule over the layers.

Exploiting local features: Rather than jointly null-model all features from a layer, instead we can null-model all possible feature *pairs* within this layer. This accounts for possible sparsity of an attack's anomalous signature within a layer[4]. For layer l with N features, there are $N_l = \binom{N}{2}$ feature pairs. For each, denoted (Z_i, Z_j)(ith and jth features from layer l), we can evaluate average weighted ADA statistics. Moreover, each of these low-order AW-ADA statistics can be *weighted* by the magnitude of the DNN weights from Z_i and Z_j to the next layer of the DNN. The DNN weightings are properly normalized and denoted β_i and β_j. The feature pairs with higher β_i and β_j have a stronger impact in classifier decision-making and thus atypicalities involving them should be given greater weight. Accordingly, for each layer we form a weighted aggregation of all low-order AW-ADA statistics, expressed for

[4]Joint atypicality of all features in a layer may be weak if only a few features exhibit strong atypicality.

layer l as *L-AWA-ADA*$^{(l)}$:

$$\frac{1}{N_l}\sum_{(i,j)}\beta_i\beta_j\sum_{c\neq c_d}\mathrm{P_{ij}}[C=c]\frac{D_{\mathrm{KL}}(P_{ij}^{(c)}||Q^{(c)})}{\mathrm{P}[C^*=c_d|C=c]}.$$

Here P_{ij} and $P_{ij}^{(c)}$ are feature-pair dependent since they are calculated using null density modelling $f(\cdot)$, which is feature-pair dependent.β_i is the sum of the magnitudes of the DNN weights that conduct from feature i in layer l to all neurons in the next layer, $l+1$, normalized by the maximum such sum over all features in layer l. $1/N_l$ is a necessary normalizer to compare distance statistics across layers fairly, since different layers have different numbers of feautures (neurons). Again we apply a max-KL rule on *L-AWA-ADA*$^{(l)}$ statistics, with the resulting method dubbed *L-AWA-ADA-maxKL*. This is the ultimate ADA detection method (achieving the best results), described in more detail in [7, 8].

4. PROPOSED DETECTION APPROACH FOR REVERSE ENGINEERING ATTACKS

Since in RE attacks the attacker submits batches of query images to the classifier, we modify L-AWA to jointly exploit batches of images in seeking to detect attacks (in this case RE query attacks, not TTEA attacks). Several schemes for *aggregating* L-AWA-ADA decision statistics, produced for individual images in a batch, are investigated: i) arithmetically averaging the L-AWA statistic over all images in a batch; ii) *maximizing* the L-AWA statistic over all images in a batch; iii) Dividing a batch into mini-batches, for example a batch of 50 images 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*). This latter scheme will be seen to perform the best.

5. EXPERIMENTAL SETUP AND RESULTS

We experimented on MNIST. This is a dataset with 60,000 grayscale images, representing the digits 0 through 9. There are 50,000 training images and 10,000 test images. As a DNN classifier, we used Lenet-5. We also used the Lenet-5 structure for training the RE attacker's substitute network. For S_0 we used 150 MNIST samples(15 from each class). We applied 5 stages of retraining (6 training stages) of the substitute DNN and chose $\lambda = 0.1$. The number of queries generated by the 5 stages were: 150,300,600, 1200 and 2400. Fast gradient sign method(FGSM) is used to craft adversarial samples. We used mini-batches of size 5 in experiments. Two maxpooling layers and the penultimate layer were used in generating the ADA detection statistics.For evaluating RE detections ROC-AUC, we used two data sources i) the 10,000(non-query) test images and ii)the query images produced in a given stage. For a given batch size, to crate a pool of samples used for evaluating ROC-AUC, we randomly drew batches from the two sources many times with replacement. The number of samples created was batch-size dependent. As one example for batch size 20, we created 427 non-query batches(samples) and 361 query batches(samples). We evaluated detection accuracy for stages 4-6 in our experiments. The reason is as follows: the substitute classifier's accuracy and the resulting success rate of TTEA attacks both grow with the stage number; by stage 4, these accuracies are 0.69 and 0.8, respectively, as shown in Figure 3. Figure 1 shows that good detection accuracy is achieved using the arithmetic averaging scheme, with the ROC AUC increasing with batch size and with the attack stage, as expected(slightly inferior performance is achieved by max rule). However, the ROC AUC appears to asymptote at about 0.95, which we would not expect – we would hope perfect detection accuracy could be approached with increasing batch size, especially in the latter stages. This better behaviour is exhibited by the mini-batch scheme with union detection rule in Figure 2. Thus, this latter aggregation scheme is the most promising one.

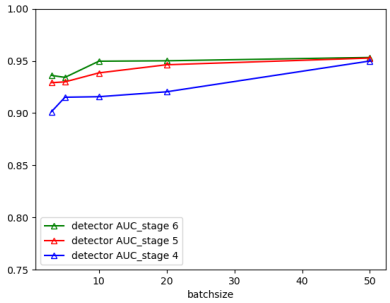

Fig. 1: RE detection ROC AUC at different stages versus batch size for arithmetic averaging scheme.

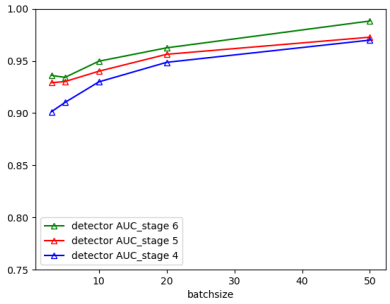

Fig. 2: RE detection ROC AUC at different stages versus batch size for mini-batch union aggregation scheme.

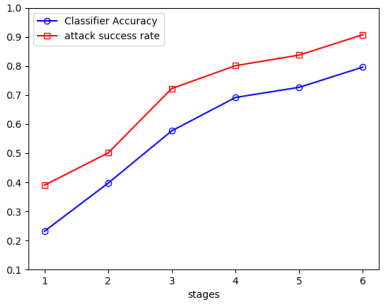

Fig. 3: attack sucess rate and classifier accuracy versus RE stage

6. CONCLUSION

We have developed an anomaly detection scheme that is very effective at detecting "stealthy" RE attacks on DNN image classifiers. This is potentially important to protect black box classifier information and to prevent TTEAs. Detection of other types of attacks, for other application domains, may be considered in future.

7. REFERENCES

[1] B. Miller, A. Kantchelian, S. Afroz, R. Bachwani, E. Dauber, L. Huang, M.C. Tschantz, A.D. Joseph, and J.D. Tygar, "Adversarial active learning," in *Proc. Workshop on Artificial Intelligence and Security (AISec)*, 2014.

[2] D.J Miller, X. Hu, Z. Qiu, and G. Kesidis, "Adversarial learning: a critical review and active learning study," in *Proc. IEEE MLSP*, Tokyo, Sept. 2017.

[3] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. 1st IEEE European Symp. on Security and Privacy*, 2016.

[4] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[5] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," in *IEEE Symposium on Security and Privacy*, 2017.

[6] N. Carlini and D. Wagner, "Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Method," in *Proc. ACM AISec, Dallas*, Nov. 2017.

[7] D.J. Miller, Y. Wang, and G. Kesidis, "When Not to Classify: Anomaly Detection of Attacks (ADA) on DNN Classifiers at Test Time," http://arxiv.org/abs/1712.06646, Dec. 18, 2017.

[8] D.J. Miller, Y. Wang, and G. Kesidis, "Anomaly Detection of Attacks on DNN Classifiers at Test Time," in *Proc. IEEE MLSP*, Sept. 2018.

[9] D.T. Davis and J.-N. Hwang, "Solving Inverse Problems by Bayesian Neural Network Iterative Inversion with Ground Truth Incorporation," *IEEE Trans. Sig. Proc.*, vol. 45, no. 11, Nov. 1997.

[10] F. Tamer, F. Zhang, A. Juels, M.K. Reiter, and T. Ristenpart, "Stealing Machine Learning Models via Prediction APIs," in *USENIX Security Symposium*, Austin, TX, Aug. 2016.

[11] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z.B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *ACM Asia Conference on Computer and Communications Security*, 2017.

[12] C. Kereliuk, B.L. Ahrendt, and J. Larsen, "Deep learning, audio adversaries, and music content analysis," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2015 IEEE Workshop on* (pp. 1-5). IEEE.