Hands On Adversarial Machine Learning: Building Practical Attacks & Defenses

Dr. Yacin Nadji

Overview

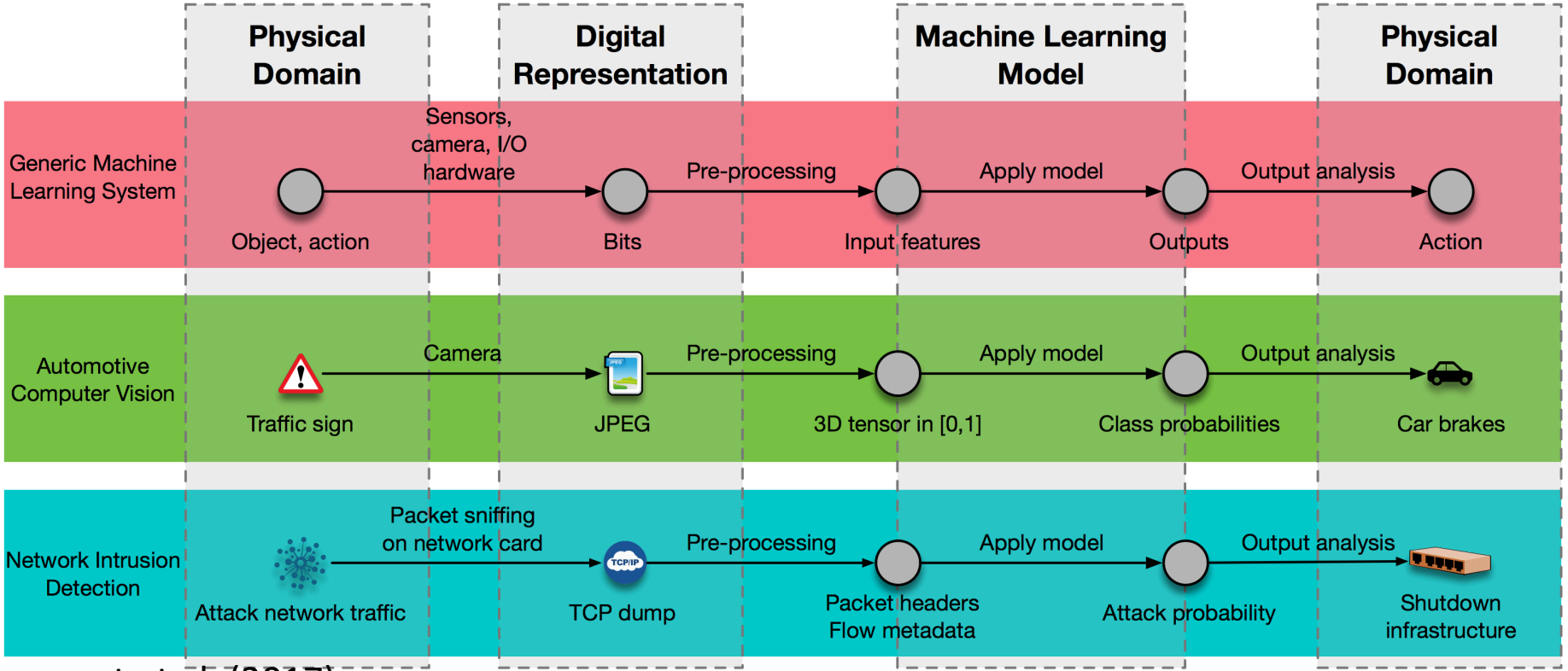
- Adversarial ML in a nutshell
- Monotonic classification
- Retraining w/ adversarial examples
- Whitebox attacks
- Whitebox defenses

ADVERSARIAL ML IN A NUTSHELL

Threat Model

- To defend against an adversary, we must understand them
- What are their *capabilities*?
  - Queryable model vs. model architecture and parameters
  - No training data vs. read training data vs. **write** training data
- *Goals*?
  - Force incorrect classifications (integrity)
  - Denial of service (availability)
  - Reverse engineer model and/or parameters (confidentiality)
  - Complete a patient's feature vector (privacy)
- *Attack surface*: where am I vulnerable?

Threat Modeling: Attack Surface



Papernot et al. (2017)

Attacks

- Training phase
  - Inject adversarial inputs, or
  - Alter the training data
- Inference phase
  - Inputs from outside expected train/test distribution
  - *White box*:
    - Attacker has insider knowledge (e.g., model params, model itself, etc.)
    - Focus on feature space where model has high error
  - *Black box*:
    - *Oracle* model from cryptography
    - More realistic (think MLaaS)

MONOTONIC CLASSIFIERS

Monotonic Classifiers

- Stream data, e.g.,
  - Stream of text data
  - Stream of system calls
- Intuition: Adding additional benign system calls should never make a malicious binary “flip” to benign
- Consider negative feature coefficients

DIRECT ATTACK

Change a feature vector for a malicious binary such that it is classified as benign. Can you create a function that always successfully “flips” from malicious to benign? Post before-and-after classification results to group chat.

INDIRECT ATTACK (PART 1)

Identify `terms` that are more likely to be associated with benign software than malicious software.

POLLING QUESTION

WHICH OF THE FEATURES BELOW WOULD BE A GOOD CANDIDATE FOR ABUSE? (MULTIPLE CHOICE, MULTIPLE ANSWERS ACCEPTABLE)

a.) `ntwritefile`
b.) `regcreatekeyex`
c.) `ntdisplaystring`
d.) `ntsettimerex`
e.) `ntreadfile`

INDIRECT ATTACK (PART 2)

Write a function that alters a malicious program's call trace so the classifier labels it benign. Describe how your function works. Post before-and-after classification results to group chat for all ground truth cases (use `collections.Counter()`).

MONOTONIC DEFENSE

Change the classifier’s coefficient for your abused feature to not be negative. See if your function from the previous exercise still work. Post your findings to group chat.

RETRAINING W/ ADVERSARIAL EXAMPLES

Retraining as a Defense

- Adversarial examples
  - Generate
    - Easy in specific cases
    - Harder to generalize
  - Retrain
    - Easy! If you have examples

BREAK DGA CLASSIFIER

Generate a function that adds additional domain names to confuse the classifier between two DGA families. Post before-and-after classification results. Make use of the `pandas.crosstab()` (see `_cv()` function).

FIX DGA CLASSIFIER

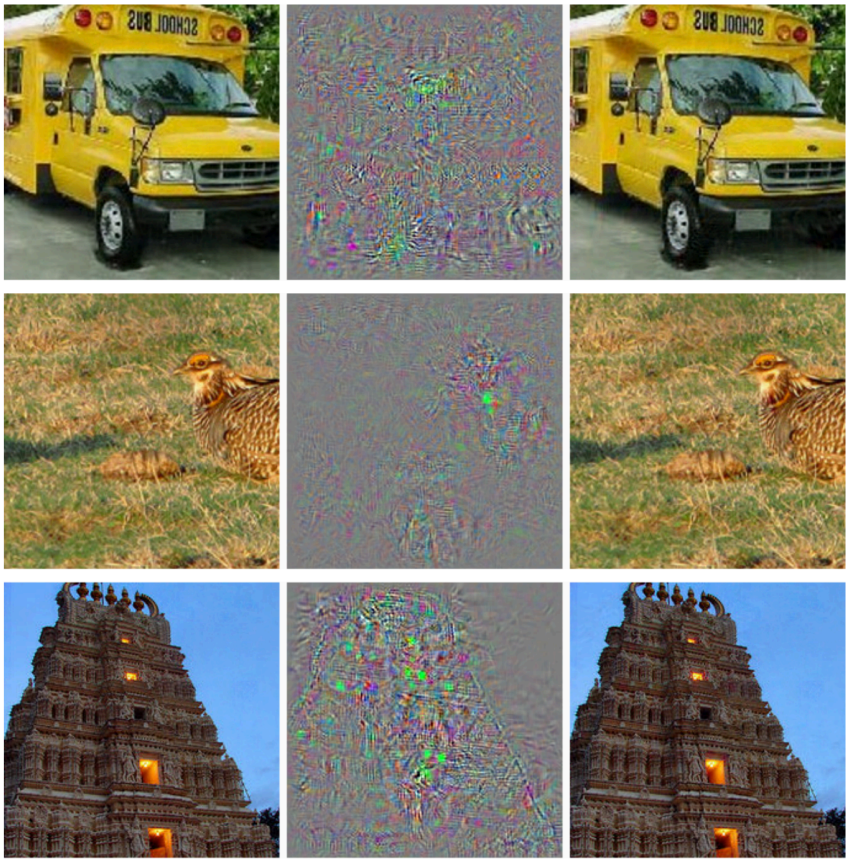
Using your adversarial examples from the previous exercise, retrain the model and demonstrate it is resistant to your function from the previous exercise. Post relevant confusion matrices to the chat.

CLEVERHANS & WHITEBOX ATTACKS

DNN Attacks

- Lots of focus on DNNs. Why?
  - Lots of space for hidden perturbations
  - Model works directly on input image/audio/etc.
- Lots of work: [25, 26, 30, 51, 52, 53, 56, 66, 67]
- Very, very good w/ incremental improvements:
  - Misclassify ~90% of MNIST
  - 4% feature perturbation —> 97% attack success rate
- Specific “junk” class doesn’t help [52]
- Model output surface poorly understood due to linear extrapolation

Whitebox (Direct)



Whitebox (Indirect) [57]



(a) Image from dataset



(b) Clean image



(c) Adv. image, $\epsilon = 4$

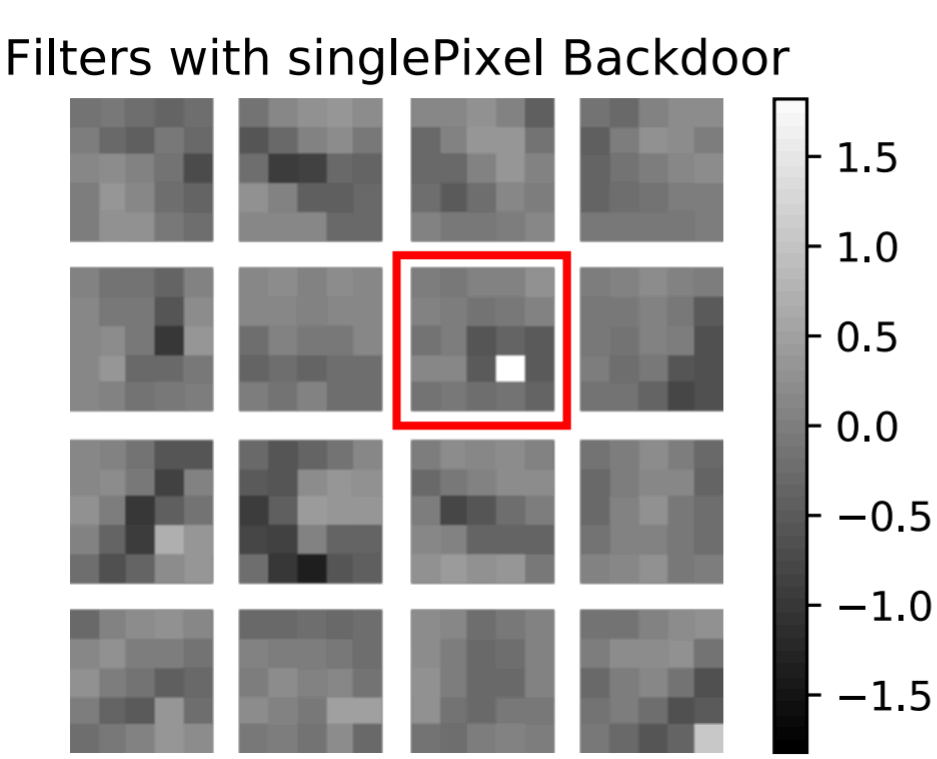


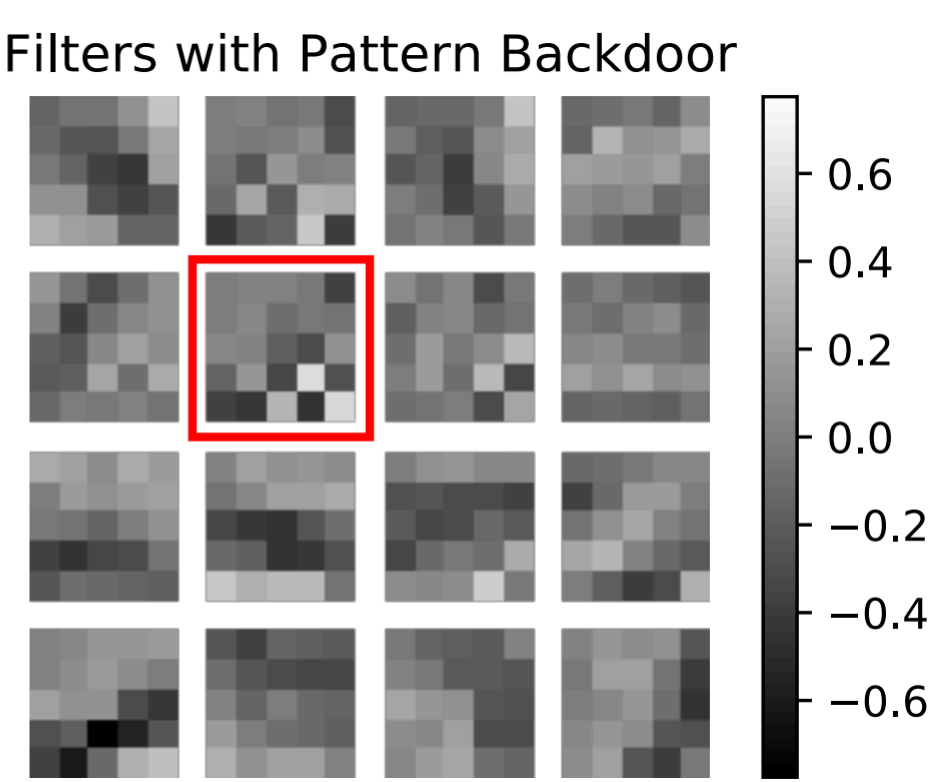
(d) Adv. image, $\epsilon = 8$

Gu et al. (BadNets)

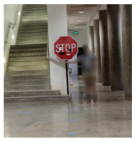


Gu et al. (BadNets)





Eykholt et al. (2018)

Distance/Angle	Subtle Poster	Subtle Poster Right Turn	Camouflage Graffiti	Camouflage Art (LISA-CNN)	Camouflage Art (GTSRB-CNN)
5′ 0°					
5′ 15°					
10′ 0°					
10′ 30°					
40′ 0°					
Targeted-Attack Success	100%	73.33%	66.67%	100%	80%

cleverhans

- Collection of adversarial example generation
- DNN-focused
- Some techniques could be adapted to other classification scenarios

RUN MNIST_EXAMPLE ATTACK

Post the test accuracies reported to group chat. Hint: set nb_epochs=1 if you TensorFlow does not detect GPUs.

POLLING QUESTION

SELECT THE MOST TRUE CHOICE BELOW

a.) Blackbox attacks are easier to perform and require a less sophisticated attacker than whitebox attacks
b.) Whitebox attacks are easier to perform and require a less sophisticated attacker than blackbox attacks
c.) Blackbox attacks are easier to perform but require a more sophisticated attacker than whitebox attacks
d.) Whitebox attacks are easier to perform but require a more sophisticated attacker than blackbox attacks
e.) Whitebox and blackbox attacks are equally easy to perform

POLLING QUESTION

SELECT THE MOST TRUE CHOICE BELOW

a.) I would expect to see more blackbox attacks than whitebox attacks in the wild

b.) I would expect to see more whitebox attacks than blackbox attacks in the wild

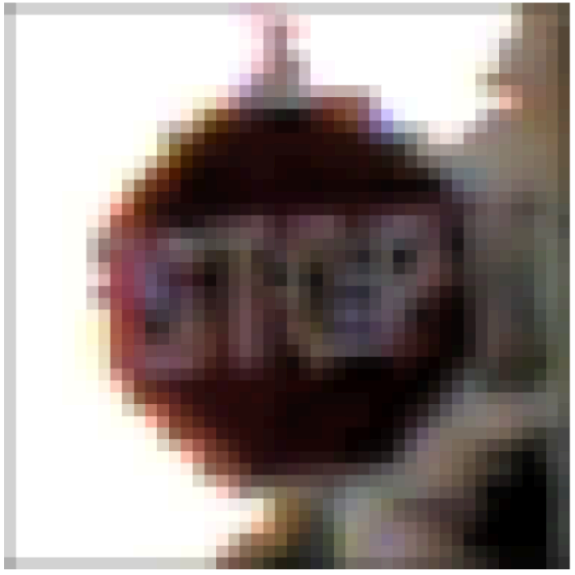
c.) I would expect to see roughly the same amount of whitebox and blackbox attacks in the wild

WHITEBOX DEFENSE: JPEG DEFENSE

JPEG Defense (Das et al. [2017])

Benign Image

Classified as
Stop Sign



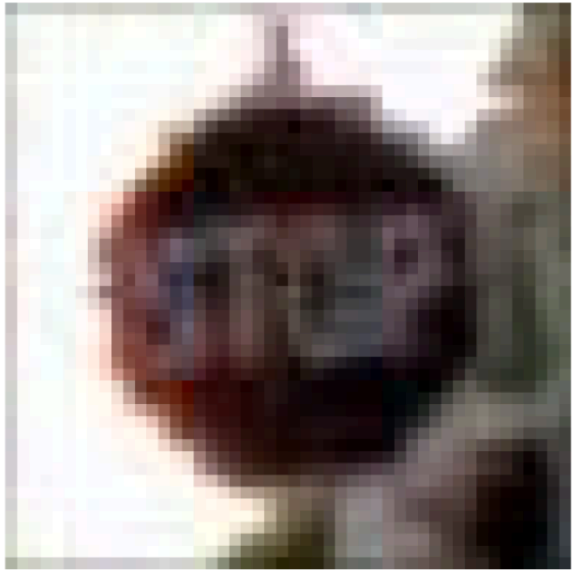
Adversarially Perturbed Image

Misclassified as
Max Speed 100



JPEG Compressed Adversarial Image

Recovers classification of
Stop Sign



JPEG Defense

- Implement JPEG defense w/ MNIST
- Recommendations:
  - Use `mnist_tutorial` code from `cleverhans`
  - `tf.image.{decode,encode}_jpeg`
  - `tf.image.convert_image_dtype`
  - Try w/ different levels of compression
- Post test accuracy & compression quality

JPEG Defense

- What was the most “effective” quality level?
- What happened when quality was increased?
- Worse or better than image example?
- Why?
- Post to group chat

References

- Papernot et al. “SoK: Towards the Science of Security and Privacy in Machine Learning” https://arxiv.org/pdf/1611.03814.pdf (see for [#] citations)
- Chen et al. “Practical Attacks Against Graph-based Clustering” https://arxiv.org/pdf/1708.09056.pdf
- Meng et al. “MagNet: a Two-Pronged Defense against Adversarial Examples” https://arxiv.org/pdf/1705.09064.pdf
- Gu et al. “BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain” https://arxiv.org/pdf/1708.06733.pdf [code]
- Das et al. “Keeping the Bad Guys Out: Protecting and Vaccinating Deep Learning with JPEG Compression” https://arxiv.org/pdf/1705.02900.pdf
- Eykholt et al. “Robust Physical-World Attacks on Deep Learning Visual Classification” https://arxiv.org/pdf/1707.08945.pdf
- Zhang et al. “DolphinAttack: Inaudible Voice Commands” https://arxiv.org/pdf/1708.09537.pdf
- Cleverhans (adversarial attack generation methods): https://github.com/tensorflow/cleverhans
- More code: https://en.wikipedia.org/wiki/Adversarial_machine_learning#Software