David J. Miller

Zhen Xiang

George Kesidis

Adversarial Learning and Secure AI

CAMBRIDGE
UNIVERSITY PRESS & ASSESSMENT

Chapter 01

Overview of Adversarial Learning

Outline

- Introduction
- Risks of AI
- Types of attacks and defenses on AI
- Privacy Issues
- Connection between AI defense and robust AI

Introduction

- With the advent of vast amounts of compute and storage power in the mid 2000s, and with large dataset repositories available, deep (large) neural networks (DNNs) have demonstrated state-of-the-art performance in some application domains.
- Now (but not previous to 2000) AI is synonymous with deep learning and “an AI” typically connotes a DNN.
- But despite the intense media hype, AI is rarely solely relied upon, and sometimes not used by rule, in settings with high financial stakes or where safety and security are significant concerns.
- Why?

Risks of AI

- After 25+ years of intense R&D in cyber security, one typically does not write software and deploy it without, e.g.,
  - thoroughly understanding its behavior for **all possible inputs** not just those deemed operationally “typical” by the developers, and
  - testing the software for bugs, logical ones in particular.
- Though this may be difficult to do for large, complex code-bases, it’s next to impossible to do for commercial AIs which are
  - highly parameterized models,
  - trained (deeply learned) on vast datasets of high-dimensional data samples,
  - and with the training done in ad hoc fashion.
- For example, ~10^{23} iterations of gradient based optimization for training a DNN model with ~10^{11} parameters.

Risks of AI – reproducibility & explainability

- Moreover, the training is in many cases difficult to reproduce.
- For example, DNNs tend to be highly overparameterized, where easy-to-use random dropout is often employed to “spread out” the learning toward heuristically improving generalization performance.
- Explainable or Interpretable AI (XAI) is a research field spanning both theory and practice to better understand and predict AI behavior in complex settings.
- Unfortunately, there is little to no such theory that is of any practical use for commercial AIs.

Risks of AI – adversarial AI

- In the past ten years, researchers have rediscovered and extended techniques to probe complex decision boundaries of trained AI classifiers to demonstrate how they may be reverse engineered and induced to make errors.
- In addition, simple training-data poisoning techniques have been developed to reduce AI performance or to plant subtle Trojans/backdoors, the latter to simplify induction to errors at test time.
- One defense idea is to make the AI generally more robust to such threats.
- Another is to deploy AIs together with (post-deployment) defenses…

Reverse Engineering Attack

- Reverse-engineering attacks (REAs) probe the model, see Chapter 14.
- [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*.
- This is achieved by probing/querying the classifier.
- But probes are anomalous w.r.t. the training set.
- So, they can easily be detected as training-set outliers, see Chapter 4.

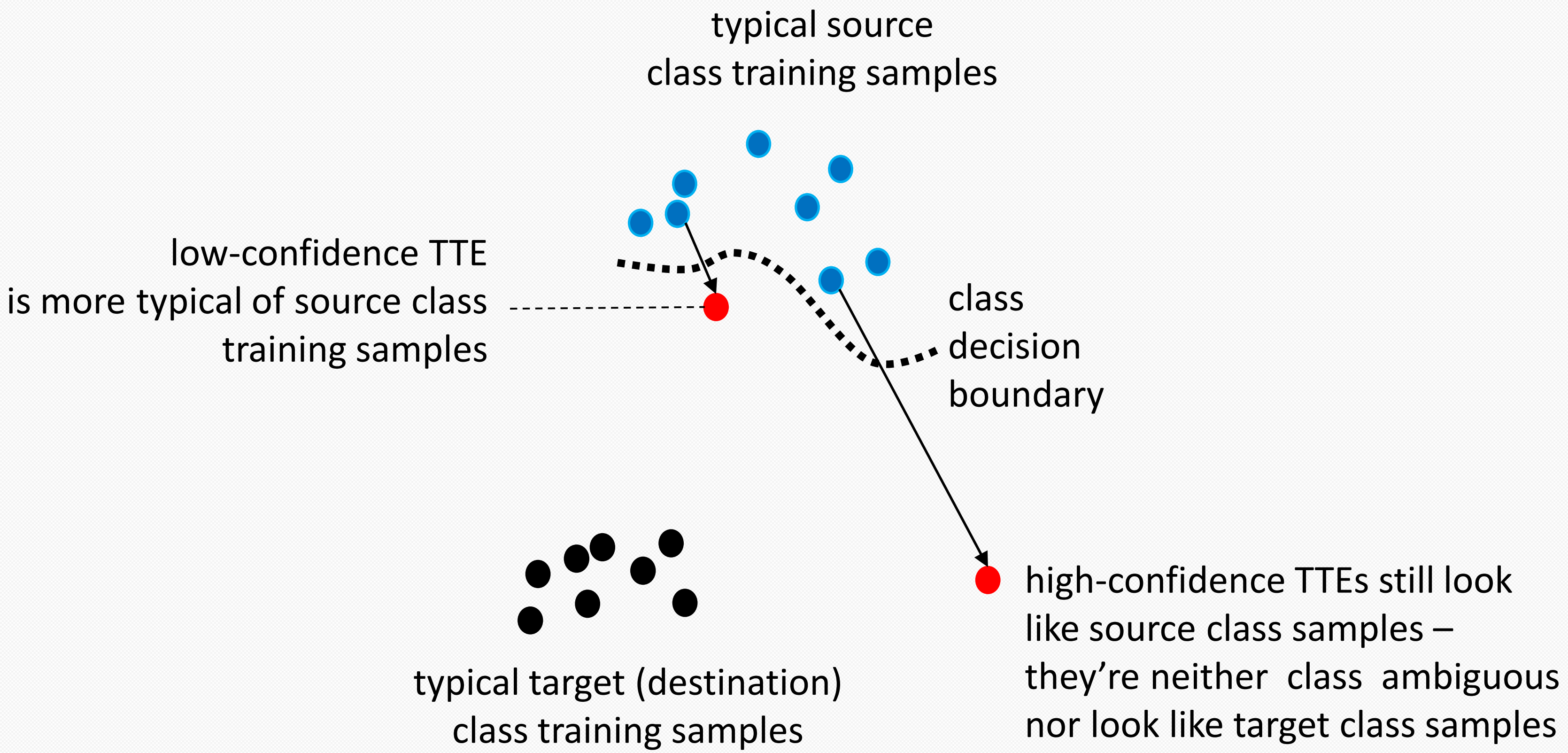
Data Privacy Issues

- REAs can also be used to try to discover something about the training dataset, which may pose a threat to privacy,
- e.g., when an AI is trained based on patient or census records.
- Here, the adversary may or may not have ”white box” access to the DNN, i.e., know the DNN model parameters.
- Note that reverse engineering a backdoor pattern obviously tells us something about the training dataset, i.e., how it was poisoned, see Chapters 6-8.

Test-Time Evasion Attacks

- A TTE is an operational/online input that produces an erroneous output and doesn’t rely on a planted backdoor.
- TTEs can be high or low confidence.
- TTEs can be informed by REAs on the model.
- Imperceptible TTEs demonstrate that DNNs are still far from achieving truly robust pattern recognition, e.g., akin to that of the human visual system.
- See Chapter 4.

High & Low Confidence TTEs



typical source
class training samples

low-confidence TTE
is more typical of source class
training samples

class
decision
boundary

high-confidence TTEs still look
like source class samples –
they’re neither class ambiguous
nor look like target class samples

typical target (destination)
class training samples

Examples of TTEs

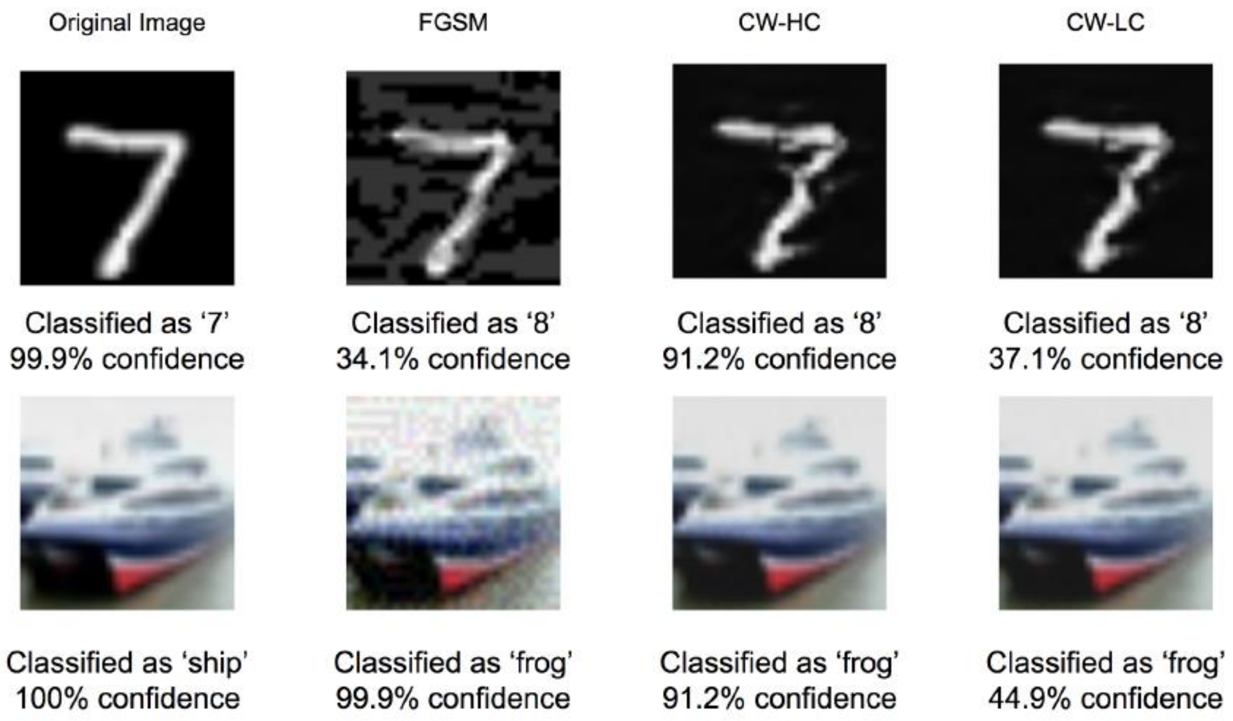


Figure 1.3 Examples of clean images, FGSM attack images [82], CW high confidence (CW-HC) images and CW low confidence (CW-LC) images [33], from MNIST [141] (first row) and CIFAR-10 [129] (second row) datasets. HC TTEs typically have much larger adversarial perturbations compared to LC TTEs. Reprinted from [284] with permission.

Data Poisoning Attacks

- DPs typically don’t rely on knowledge of model to be trained.
- Error generic DPs (or just DPs) aim to decrease model accuracy (e.g., by label flipping), see Chapter 13.
- DPs to plant backdoors (called backdoor attacks or Trojans) are triggered at test time (easier to trigger than TTEs), see Chapters 5-11.

Trojans in AIs

- Training data is poisoned by samples with backdoors incorporated.
- The backdoor pattern is incorporated into clean samples from one (source) class and labelled as belonging to another (target) class.
- Such poisoned samples are inserted into the training dataset.
- This can happen through insecure supply chains or insiders.
- The attacker need not know the AI architecture or training method.
- Only modest poisoning needed to effectively plant the backdoor.
- Accuracy of classifier or regressor on clean samples should **not** be significantly impacted (else the attack is easily detected).
- Backdoors have been successfully planted for a wide variety of AI architectures and in various application domains.

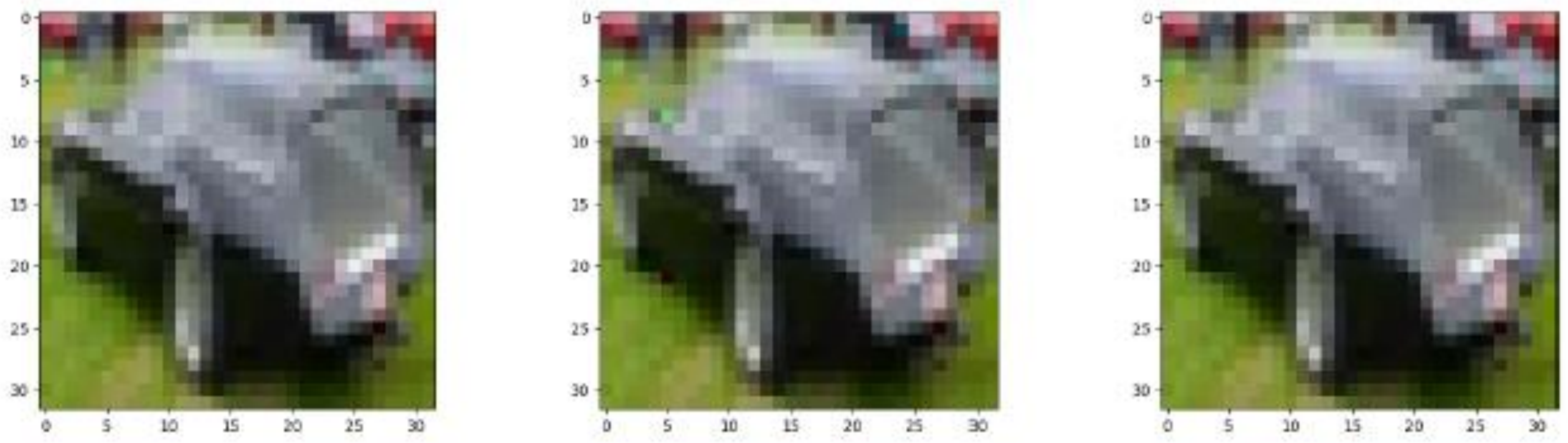
Trojans in AIs – online triggers

- A test-time source sample with the backdoor physically or digitally incorporated “triggers” the backdoor so that classifier infers the target class, see chapter 10.
- Or a triggerless backdoor can be covertly planted, e.g., in a deep regression system, see Chapter 12.
- Backdoor triggering is simpler than crafting a TTE (without backdoor data poisoning).

Effective Trojans

- Note that it would be easy to detect backdoor poisoning if the AI’s accuracy on some known clean samples was significantly affected by it.
- Backdoor pattern should be innocuous - the sample should “nominally appear” to belong to the source class and should not appear to have been tampered with.
- That is, the backdoor should evade detection by cursory examination by a human or by simple automated means.
- Hence our focus is on poisoned samples either with imperceptible (small magnitude) or perceptible but scene-plausible backdoors.
- Backdoor attacks may exhibit “collateral damage” where the backdoor pattern also causes non-source-class samples to be classified to the target class.
- The attacker can mitigate collateral damage by poisoning non-source-class samples with the backdoor pattern but without changing the class label.

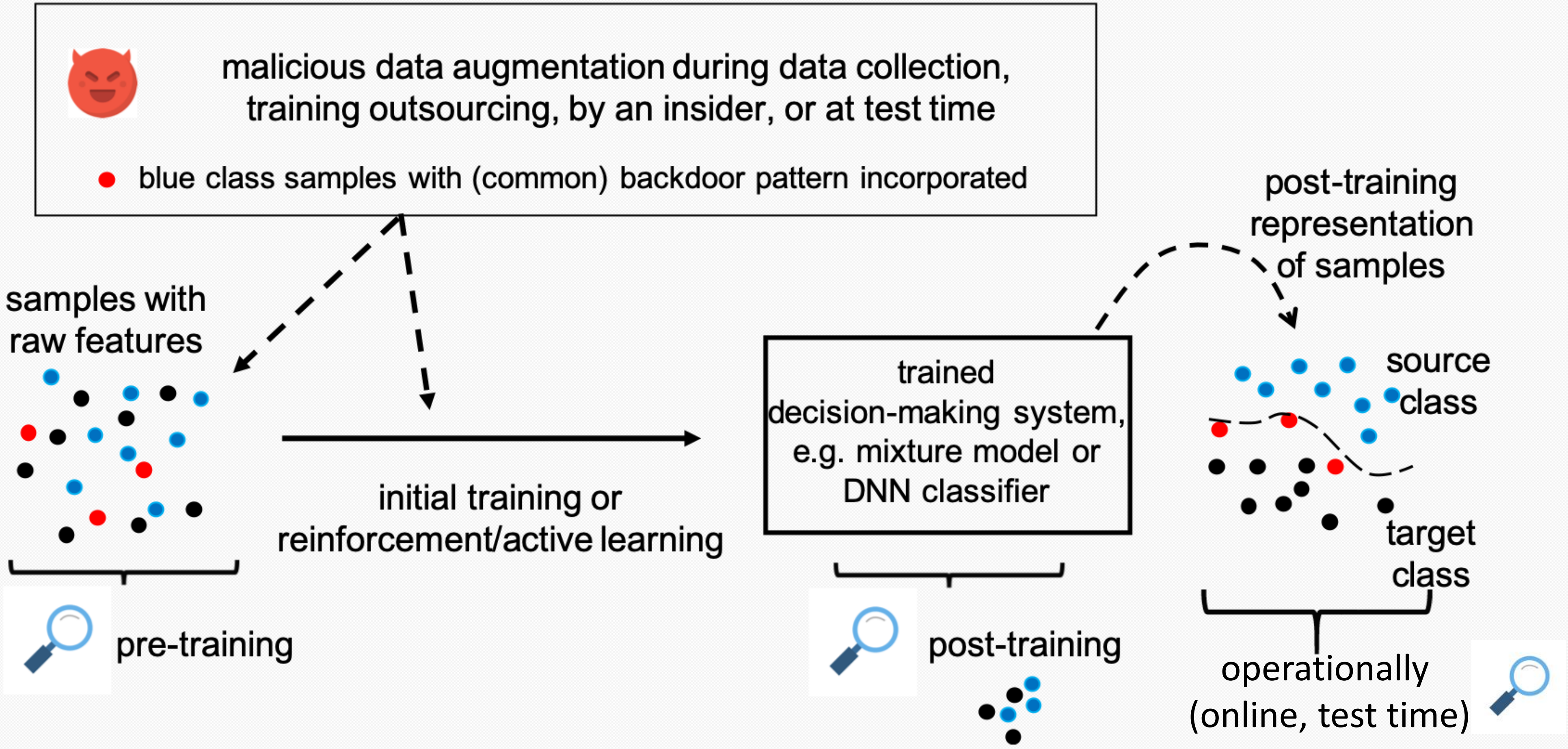
Imperceptible Trojan examples



(a) (b) (c)

Fig. 6. Examples of backdoor patterns applied to CIFAR-10 images. (a) Original automobile image. (b) Automobile with sparse, pixel-wise perturbation $(||\underline{v}^*||_2 = 0.6)$. (c) Automobile with global perturbation $(||\underline{v}^*||_2 = 0.2)$.

Trojan defense scenarios


malicious data augmentation during data collection, training outsourcing, by an insider, or at test time
● blue class samples with (common) backdoor pattern incorporated
samples with raw features
initial training or reinforcement/active learning
trained decision-making system, e.g. mixture model or DNN classifier
post-training representation of samples
source class
target class
pre-training
post-training
operationally (online, test time)

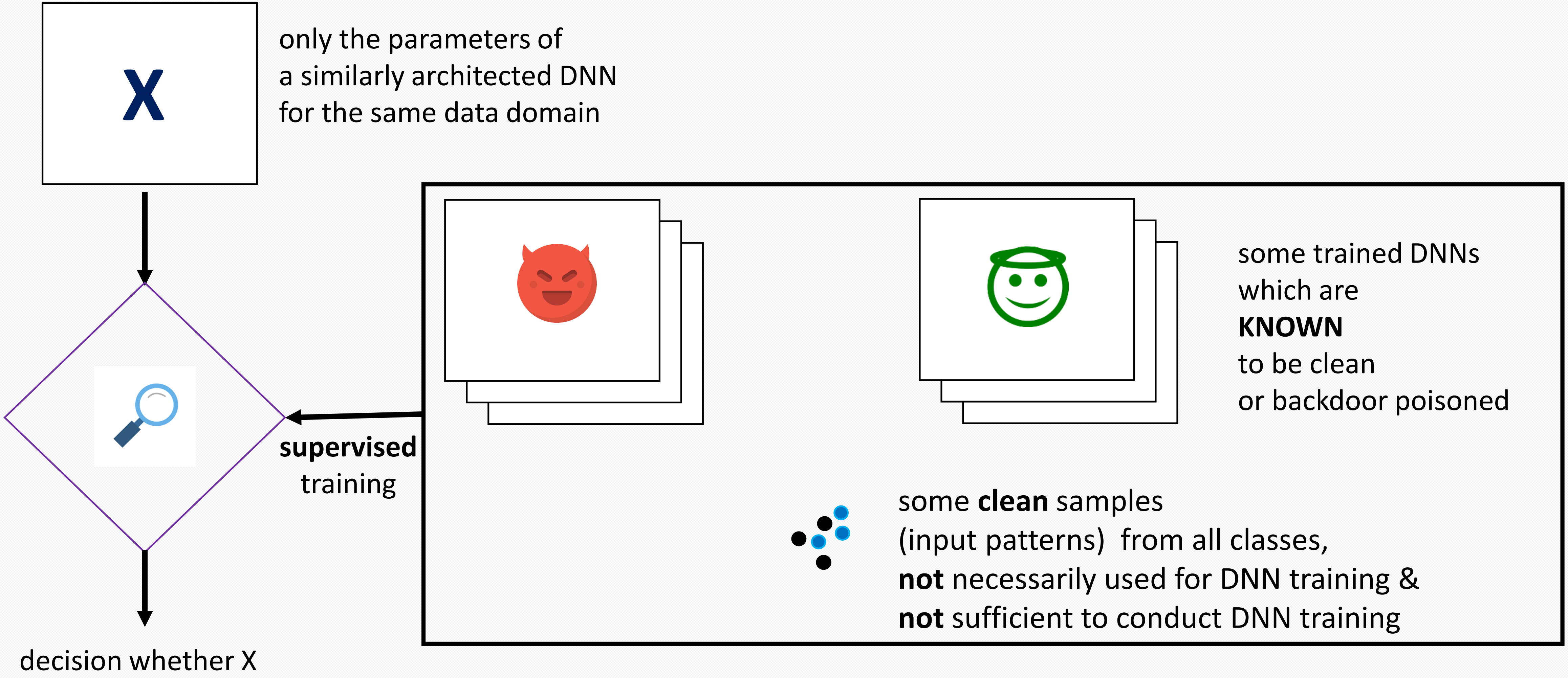
Trojan defense scenarios

- Pre-training or during training
  - Goal is to cleanse the potentially poisoned training dataset.
  - Here, no small clean dataset is assumed available.
- Post training and pre-deployment
  - A small clean dataset is sometimes assumed to be available with representatives from each class.
  - There may be both detection and mitigation objectives.
- During test time (operation time, online)
  - Detect backdoor triggers.
  - Can leverage pre-deployment defenses, e.g., Reverse Engineering Defenses.
  - Can attempt to correct class decision (mitigation).

Post-training but pre-deployment defense scenarios for Trojans in AIs

- Detect whether a trained AI has a Trojan/backdoor
- This problem is **fundamental** to explainable AI (XAI)
- Defenses include those which are
  - Supervised, e.g., IARPA TrojAI 2019 scenario given DNNs that are known to be poisoned and others that are known to be clean
  - Unsupervised, but with sensitive hyperparameters
  - Unsupervised

IARPA TrojAI 2019 supervised problem



X

only the parameters of
a similarly architected DNN
for the same data domain

some trained DNNs
which are
KNOWN
to be clean
or backdoor poisoned

some **clean** samples
(input patterns) from all classes,
not necessarily used for DNN training &
not sufficient to conduct DNN training

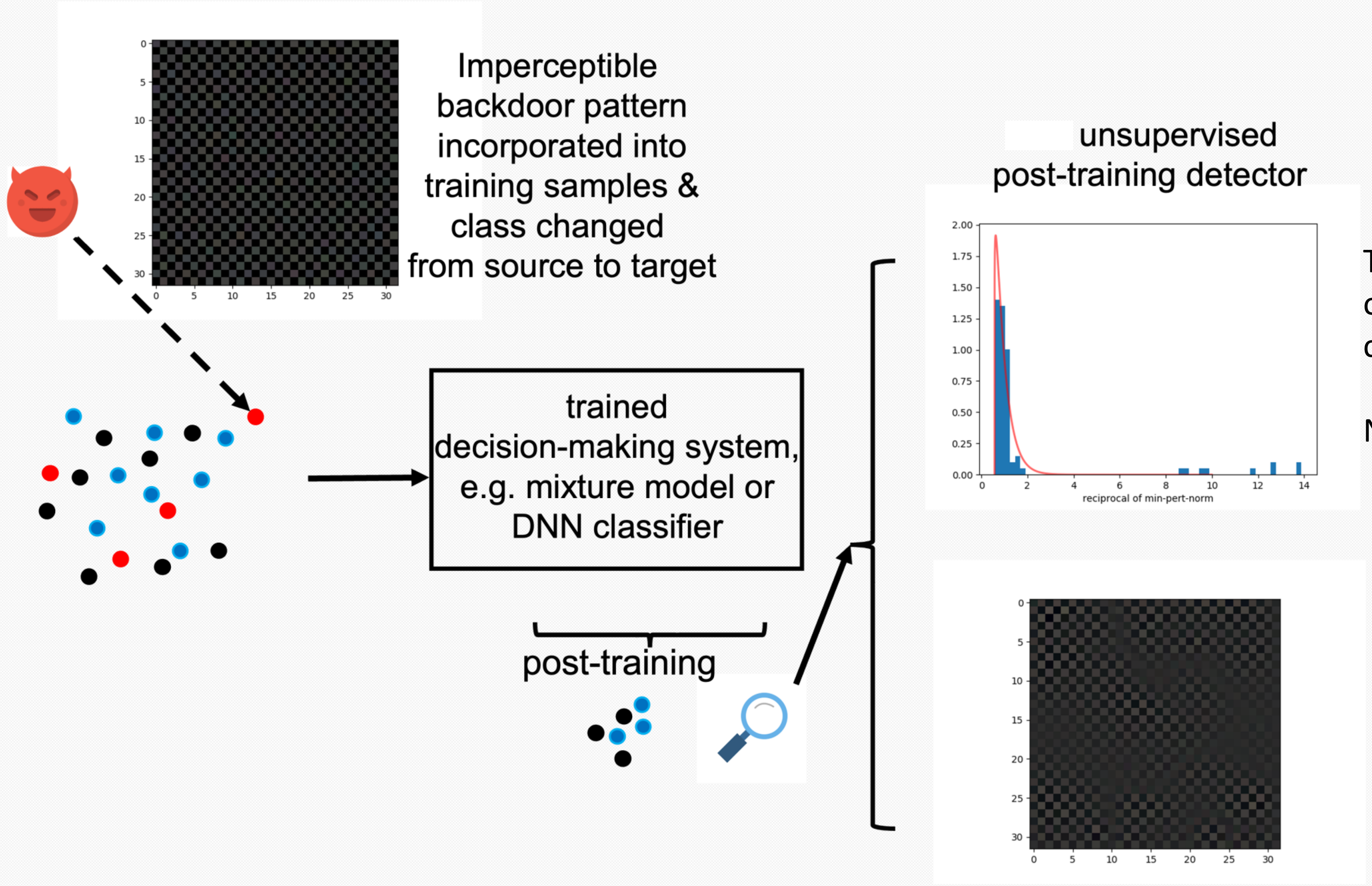
supervised
training

decision whether X

Unsupervised PT Defense of Classifiers - Overview

- Reverse engineering defenses (REDs) attempt to reconstruct the backdoor pattern, e.g.,
  - by additive perturbations of input or embedded features – e.g. I-PT-RED
  - by patch or ”blended” perturbations of input features – e.g., NC, ABS, P-PT-RED
  - by transferability of *sample-specific* perturbations, associated with a ”universal” detection threshold – T-PT-RED
- Non-RE based defenses, e.g., META, FP, UnivBD/BM
- Ensemble defenses

I-PT-RED



Imperceptible backdoor pattern incorporated into training samples & class changed from source to target

trained decision-making system, e.g. mixture model or DNN classifier

post-training

unsupervised post-training detector

Target & source classes of the backdoor attack clearly identified.

Note collateral damage.

reverse-engineered backdoor pattern without access to training dataset

Post-training & Pre-deployment scenarios

- Note that the previous slide indicates reverse engineering by the defender of the backdoor pattern that was used to poison the training dataset.
- REDs have been developed for various attack configurations and various methods of backdoor pattern incorporation, see Chapters 6-8.
- Also, there are defenses which are agnostic to the backdoor pattern and method of incorporation, see Section 6.4.4 and Chapter 9.

Robust AI versus Secure AI

- Phenomena similar to backdoors may occur naturally, see “intrinsic” backdoors of Chapter 11.
- For another example, green grass may trigger classification to the “cow” class of a classifier of animal images if, in training samples, cows are commonly present in a pasture and grass is relatively uncommon in images of other animals.
- How to address intrinsic backdoors, and issues with AIs which are effectively exploited by TTEs, are problems of robust learning.
- Techniques similar to those used to address attacks can be used to improve robustness.

Unlearning training samples

- Post-training “unlearning” of certain training samples is a process through which the DNN is modified to minimize or eliminate the marginal effects of certain training samples.
- Unlearning is motivated by, e.g.,
  - the problem of data poisoning (see Chap. 9),
  - the need to manage the capacity of a dynamically fine-tuned DNN under active learning (see Chap. 12), or
  - data privacy.
- See, e.g., T. Shaik, X. Tao, H. Xie, L. Li, X. Zhu and Q. Li. Exploring the Landscape of Machine Unlearning: A Comprehensive Survey and Taxonomy. https://arxiv.org/pdf/2305.06360.pdf, 2023.

With Permission, Figures Reproduced From

- D.J. Miller, Y. Wang, G. Kesidis. Anomaly Detection of Attacks (ADA) on DNN Classifiers at Test Time. *Neural Computation* **31**(8), Aug. 2019.
- D.J. Miller, Z. Xiang and G. Kesidis. Adversarial Learning in Statistical Classification: A Comprehensive Review of Defenses Against Attacks. *Proceedings of the IEEE* **108**(3), March 2020.
- H. Wang, Z. Xiang, D.J. Miller and G. Kesidis. Anomaly Detection of Test-Time Evasion Attacks Using Class-Conditional Generative Adversarial Networks. *Computers and Security* **124**, Jan 2023.