Module 11
Hacking Machine Learning Models

Can you hack a model?

YES!!

Attack Types

- Model Evasion
- Poisoning
- Membership Inference
- Model Extraction / Inversion

White box attacks require access to the model, and black box only require access to the output.

Attack Surface

Phase	Description	Activity Type
Data Collection	Models require data to be trained. Data is usually collected from both public and platform sources with a specific model in mind. This is an ongoing process and data will continue to be collected from these sources.	Train
Data Processing	The collected data is processed in any number of ways before being introduced to an algorithm for both training and inference.	Train
Model Training	The processed data is then ingested by an algorithm and a model is trained.	Train
Model Validation	After a model is trained, it is validated to ensure accuracy, robustness, explainability, or any number of other metrics.	Train
Model Deployment	The trained model is embedded in a system for use in production. Machine learning is deployed in a wide variety of ways – inside autonomous vehicles, on a web API, in client-side applications.	Inference
System Monitoring	Once the model has been deployed, the “system” is monitored. This includes aspects of the system that may not related to the ML model directly.	Inference

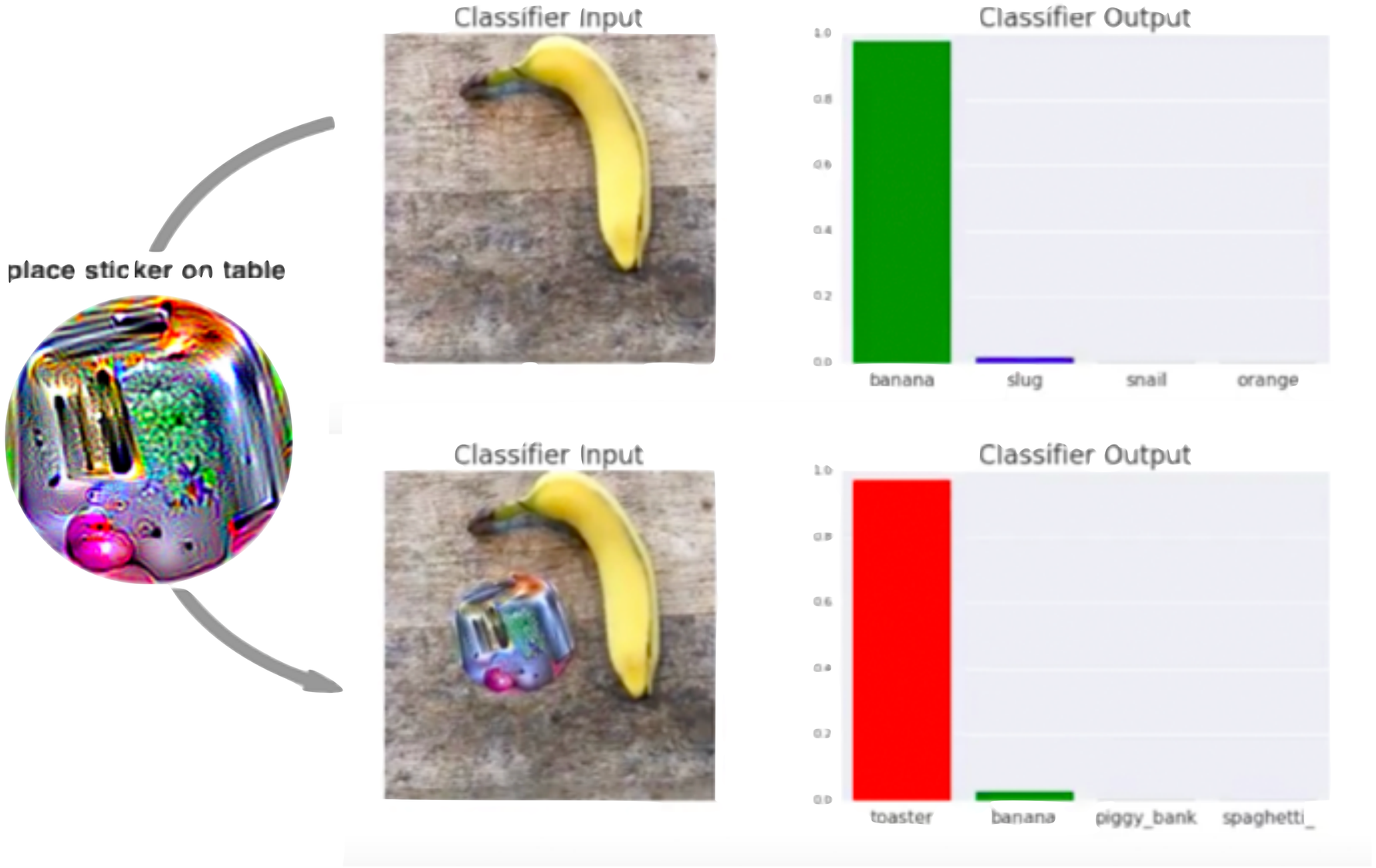
Attack Types

Attack	Description	Activity Type
Functional Extraction	An adversary successfully copies a model functionality.	Inference
Model Evasion	An adversary successfully causes a model to misclassify an input.	Inference
Model Inversion	An adversary successfully recovers internal state information from the model such as trained weights.	Inference
Membership Inference	An adversary successfully recovers data the model was trained on.	Inference
Model Poisoning	An adversary successfully alters inputs to the model at train time.	Train
System Compromise	Traditional attacks against infrastructure components.	N/A

Model Evasion

Deep Neural Networks are Easily Fooled

High Prediction Scores for Unrecognizable Images



https://gizmodo.com/this-simple-sticker-can-trick-neural-networks-into-thin-1821735479

An attack caused a model to label this image as a 45mph Speed Limit Sign

=

Ivan Evtimov et al. ."Robust Physical-World Attacks on Deep Learning Models" (2017)

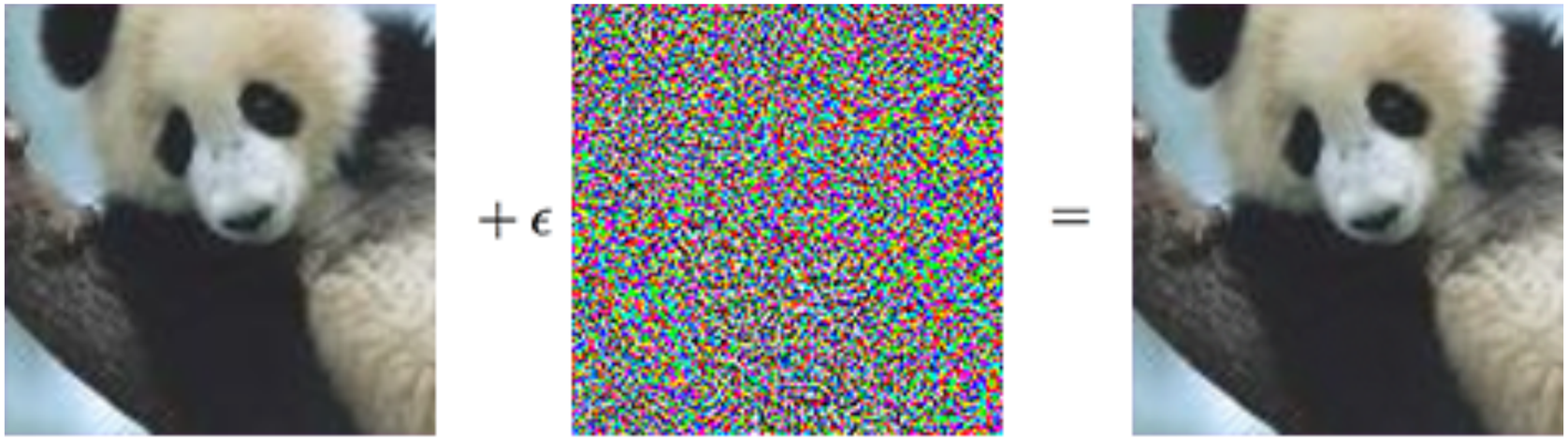
An attack caused a model to label this image as a Stop Sign

=

Ivan Evtimov et al. ."Robust Physical-World Attacks on Deep Learning Models" (2017)

Altering a Prediction

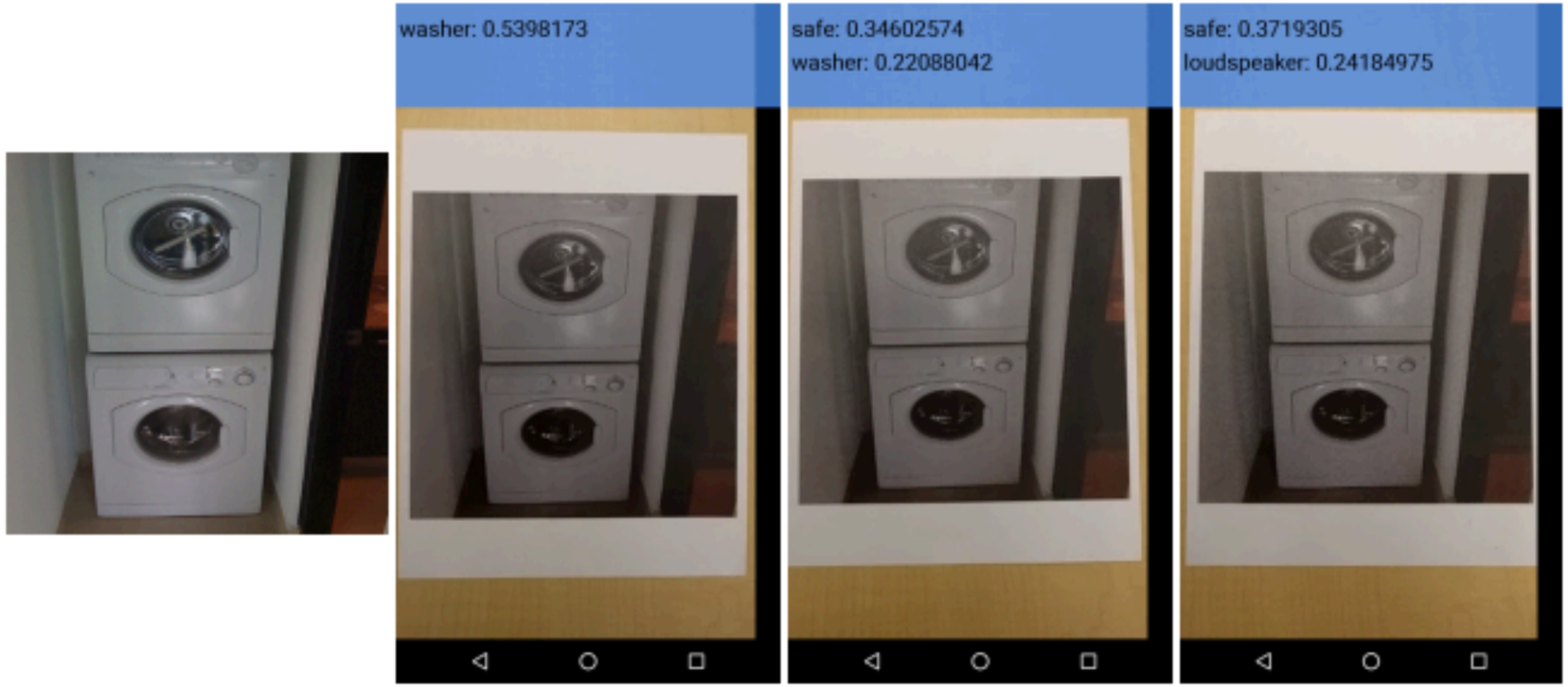
By adding small perturbations to an image, it is possible to completely alter the prediction.



Explaining and Harnessing Adversarial Examples, Goodfellow et al., 2014

Altering a Prediction

Photos taken on a smartphone and printed out can be altered in this way.



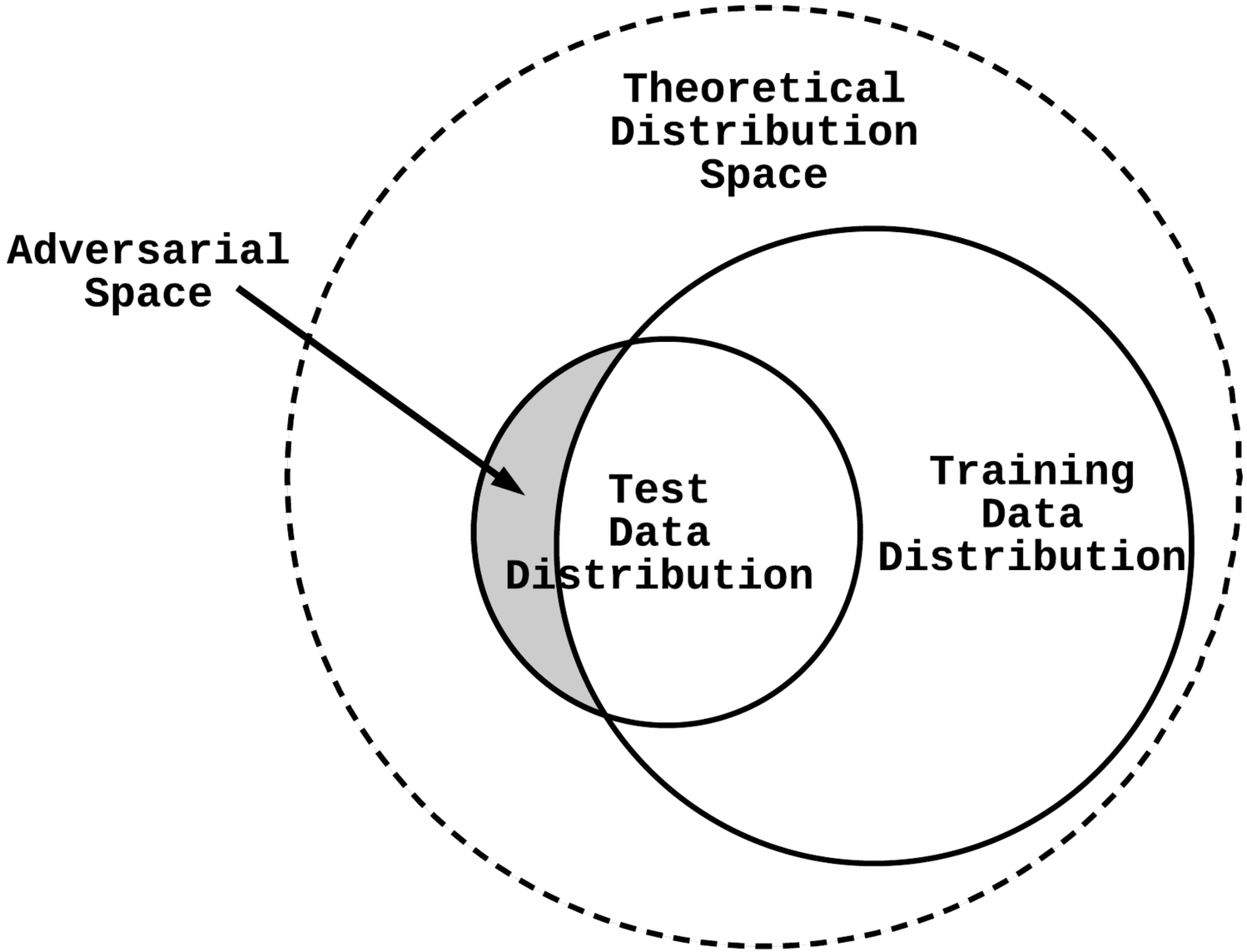
(a) Image from dataset (b) Clean image (c) Adv. image, $\epsilon = 4$ (d) Adv. image, $\epsilon = 8$

Explaining and Harnessing Adversarial Examples, Goodfellow et al., 2014

Model Evasion Attacks

- A model evasion attack is when a model is fed a carefully crafted input, known as an adversarial example, that is designed to be misclassified.
- This attack was first described in 2004.
- Model evasion attacks can be successfully executed either as a black or white box attack.
- The model's complexity is not a defense against an evasion attack.

https://towardsdatascience.com/evasion-attacks-on-machine-learning-or-adversarial-examples-12f2283e06a1



Clarence Chio, David Freeman. Machine Learning & Security. Pg. 319 (2018)

Introducing ART

- The Adversarial Robustness Toolbox (ART) is a collection of automated tools for attacking and defending ML models.
- It's like Metasploit for adversarial machine learning.
- Works with most ML frameworks such as scikit-learn, pytorch, XGBoost and others.
- Supports attack and defense, as well as black and white box attacks.
- Works with a variety of ML models from simple SVM to neural networks.

https://github.com/Trusted-AI/adversarial-robustness-toolbox

Other Adversarial Frameworks

There are a few other frameworks which can automate hacking ML models, or at least see how vulnerable a model is to adversarial attacks.

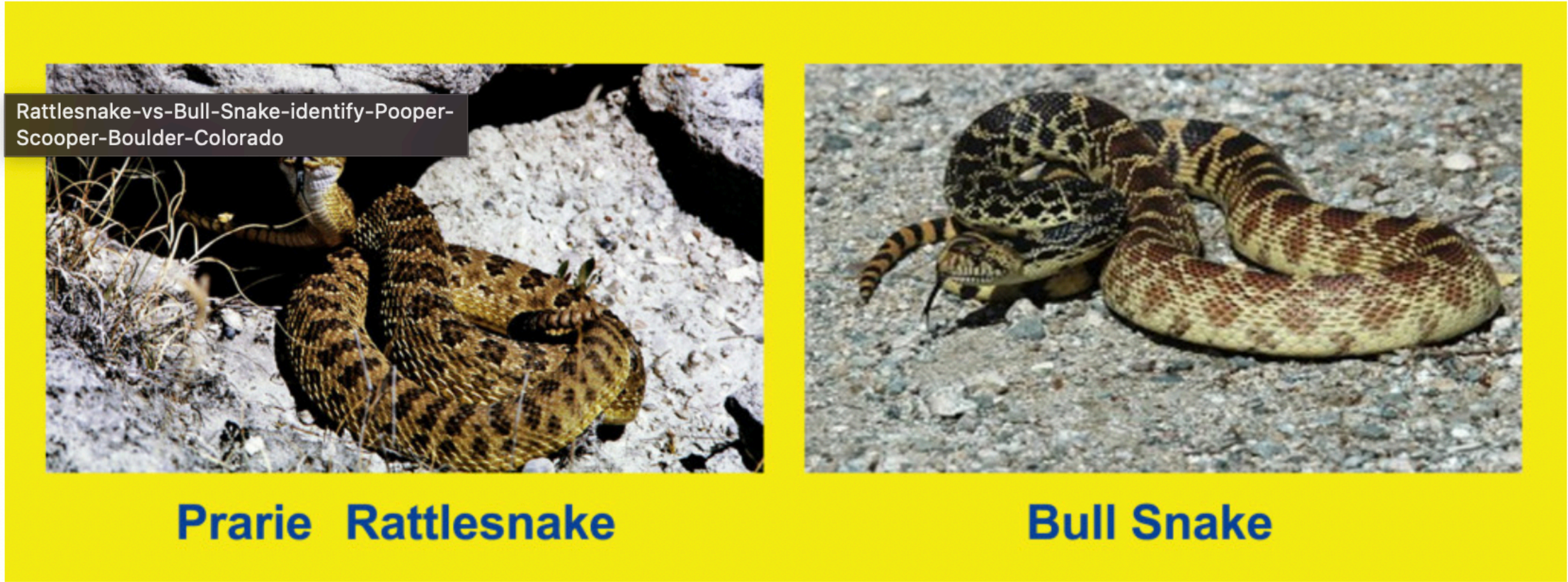
- Counterfeit: Command line security assessment tool (https://github.com/Azure/counterfit)
- TextAttack: Generates adversarial examples for NLP Models (https://github.com/QData/TextAttack)
- Cleverhans is built by Google and part of Tensorflow. (https://github.com/tensorflow/cleverhans) *
- Deep-pwn: Billed as Metasploit for machine learning: (https://github.com/cchio/deep-pwning) *

* Projects may no longer be supported or developed.

Functional Extraction

Functional Extraction

- Mimicry: the action or art of imitating someone or something



http://poofree.com/colorado-rattlesnakes-and-your-furry-friend-duty-free-pets-colorado-pooper-scooper/rattlesnake-vs-bull-snake-identify-pooper-scooper-boulder-colorado/

Functional Extraction: So What?

- Having a “copy” of a model to play with allows attackers to better attempt other attack methods
  - Evasion - Attackers can develop tactics to evade the defensive models using their copy
  - Poisoning - Attackers can practice poisoning their model and establish some good baseline idea of what it would take to move boundaries on the target model

http://poofree.com/colorado-rattlesnakes-and-your-furry-friend-duty-free-pets-colorado-pooper-scooper/rattlesnake-vs-bull-snake-identify-pooper-scooper-boulder-colorado/

Model Inversion
&
Membership Inference

Inversion/Inference Example

- A facial recognition model “ W “ is produced using a training procedure “ Y “ and data “ X “.
- Model is released to the public for use (white box)
- Given the model and possible output labels can training data be recovered?

Inversion/Inference Example

Figure 1: An image recovered using a new model inversion attack (left) and a training set image of the victim (right). The attacker is given only the person's name and access to a facial recognition system that returns a class confidence score.

https://www.cs.toronto.edu/~toni/Courses/Fairness/Lectures/ML-and-DP-v2.pdf

Inversion/Inference So What

- Sensitive Information
  - Credit Card Numbers
  - Addresses
  - Survey data

risk-taking behaviors [17]. To support the analysis, FiveThirtyEight commissioned a survey of 553 individuals from SurveyMonkey, which collected responses to questions such as: "Do you ever smoke cigarettes?", "Have you ever cheated on your significant other?", and of course, "How do you like your steak prepared?". Demographic characteristics such as

and 11 variables, including basic demographic information and responses to questions such as, "How happy are you in your marriage?" and "Have you watched X-rated movies in the last year?" We discarded rows that did not contain re-

https://www.cs.toronto.edu/~toni/Courses/Fairness/Lectures/ML-and-DP-v2.pdf

Poisoning

Defined

- **Poisoning Attack:** Used with online learning systems. Injecting data to cause a model to modify its decision boundary in a particular direction.

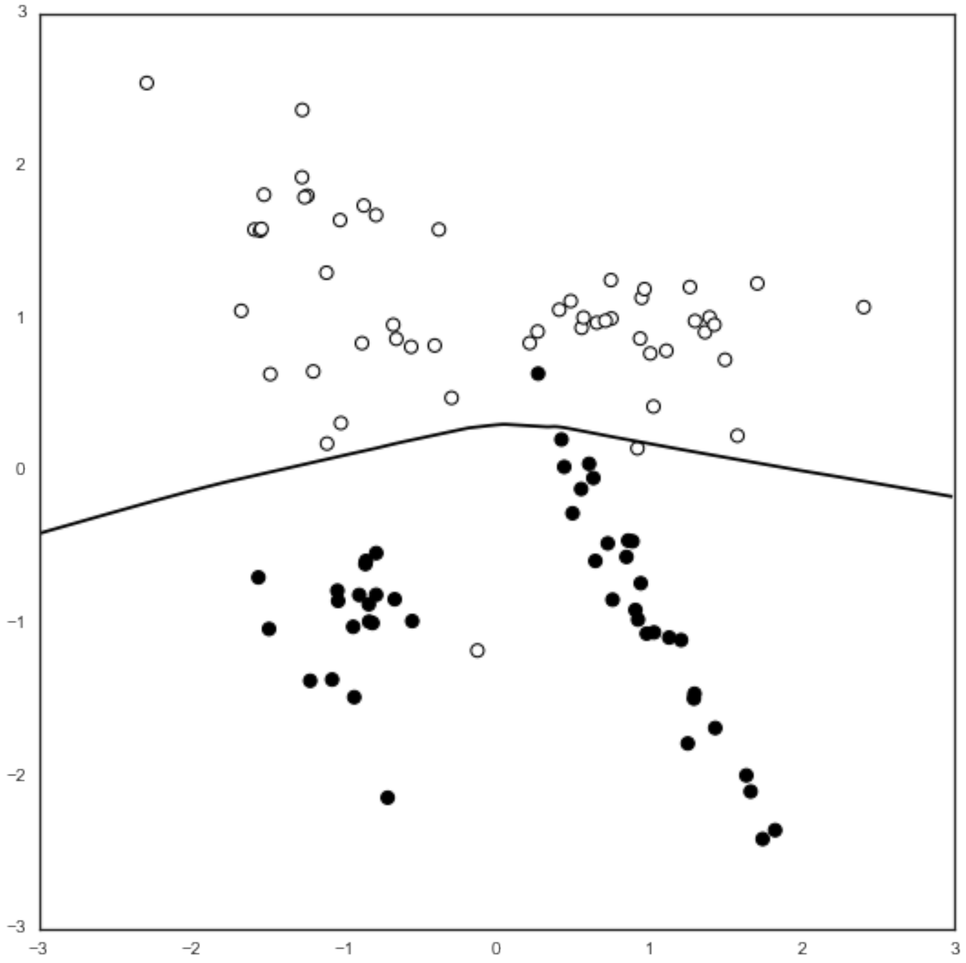
Poisoning Attack

- Online learning systems automatically adjust model parameters over time based on input
- Poisoning attacks, an actor injects new data into a retraining set with the intent of altering the decision boundaries.

Poisoning Attack

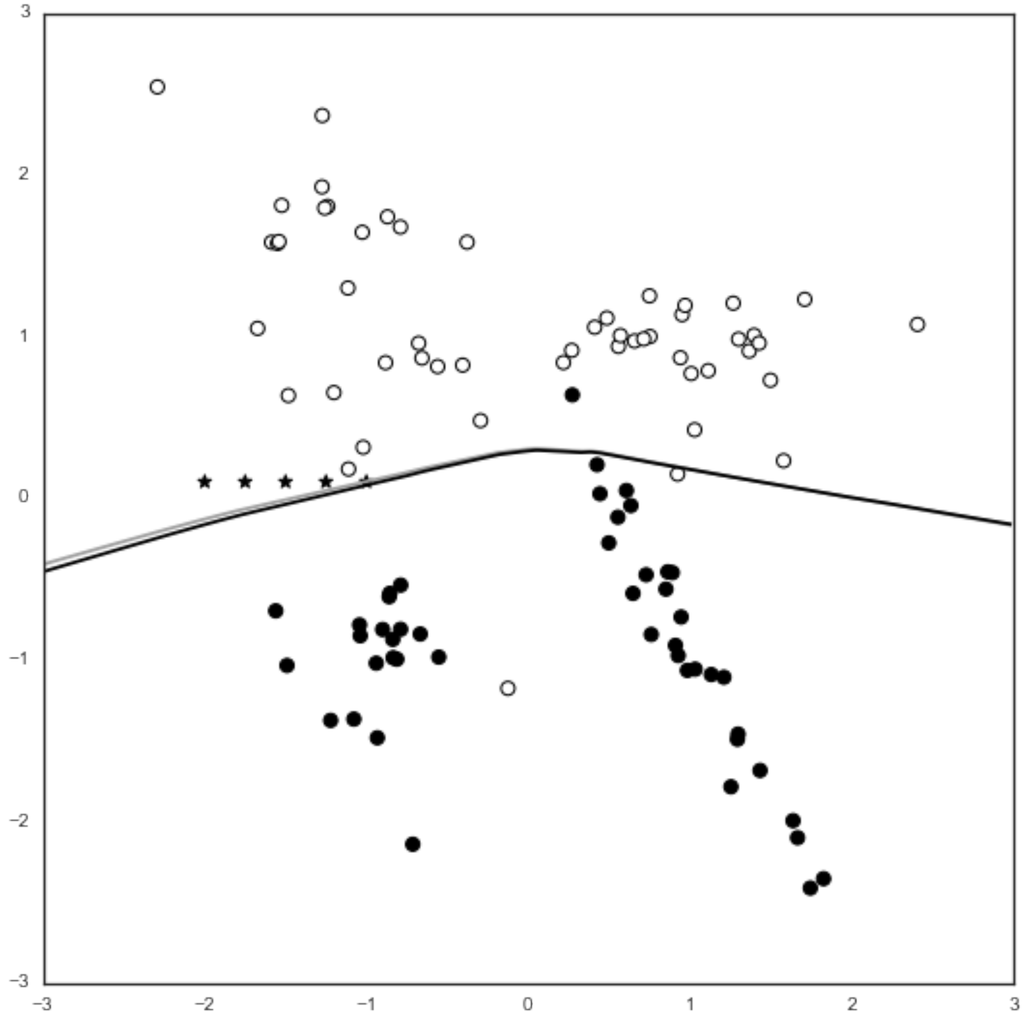
Clarence Chio, David Freeman. Machine Learning & Security. Pg. 322 (2018)

Poisoning Attack



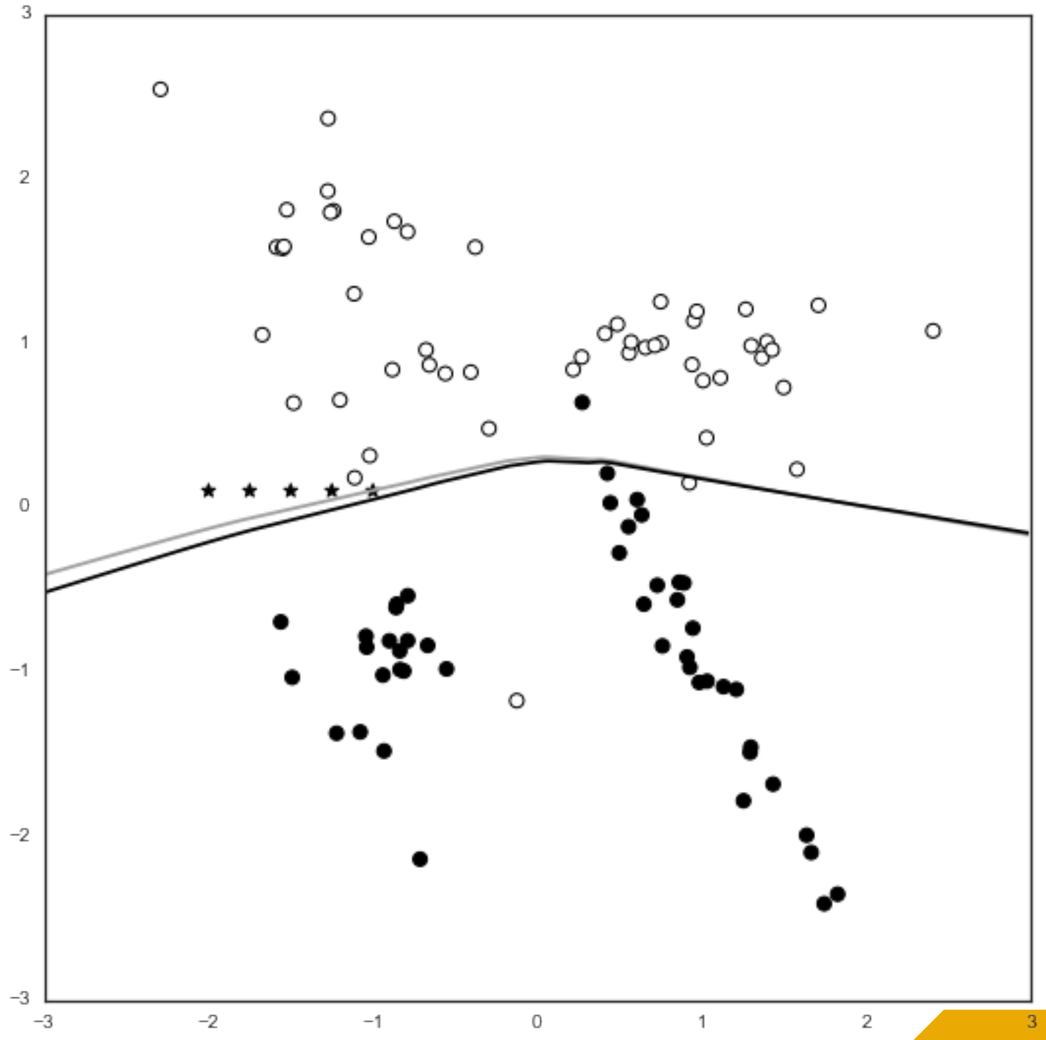
Clarence Chio, David Freeman. Machine Learning & Security. Pg. 322 (2018)

Poisoning Attack



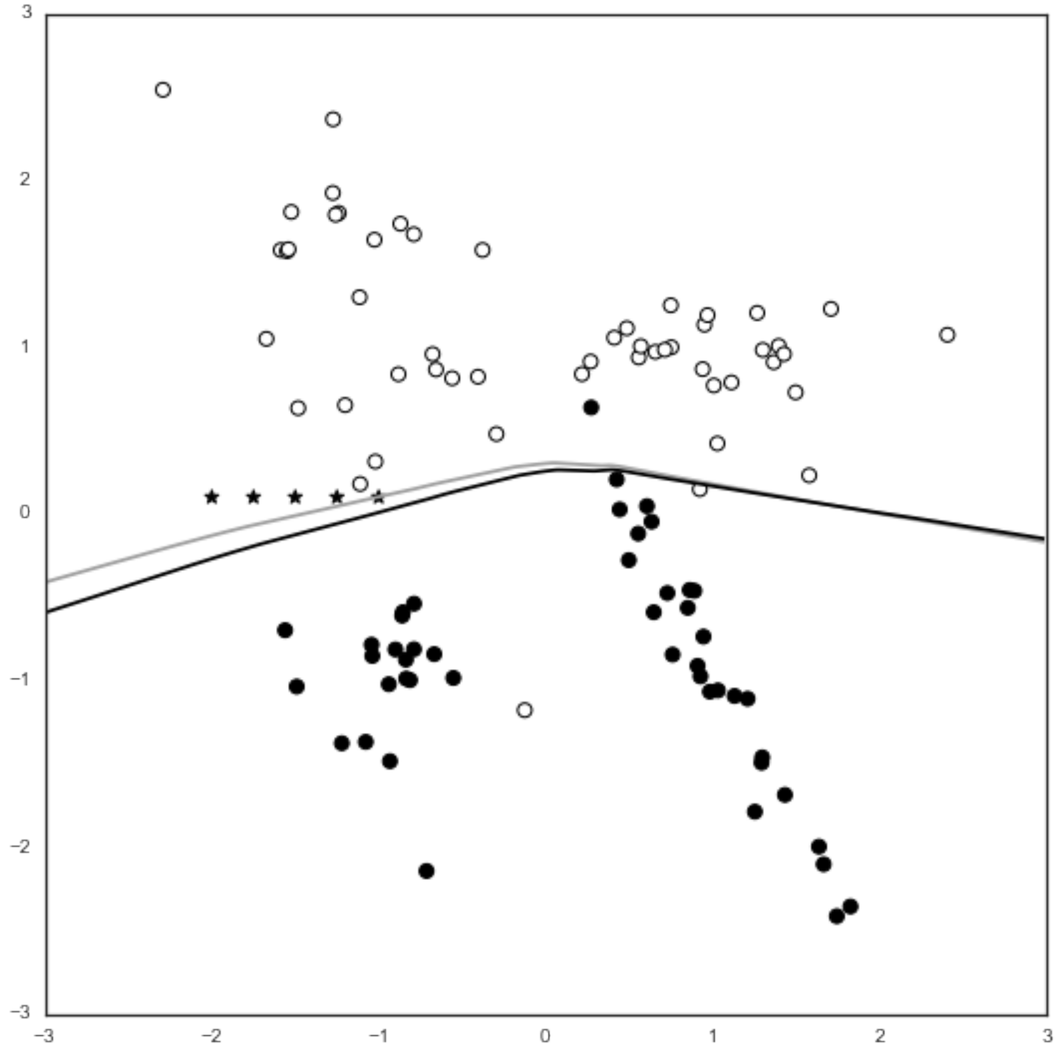
Clarence Chio, David Freeman. Machine Learning & Security. Pg. 322 (2018)

Poisoning Attack



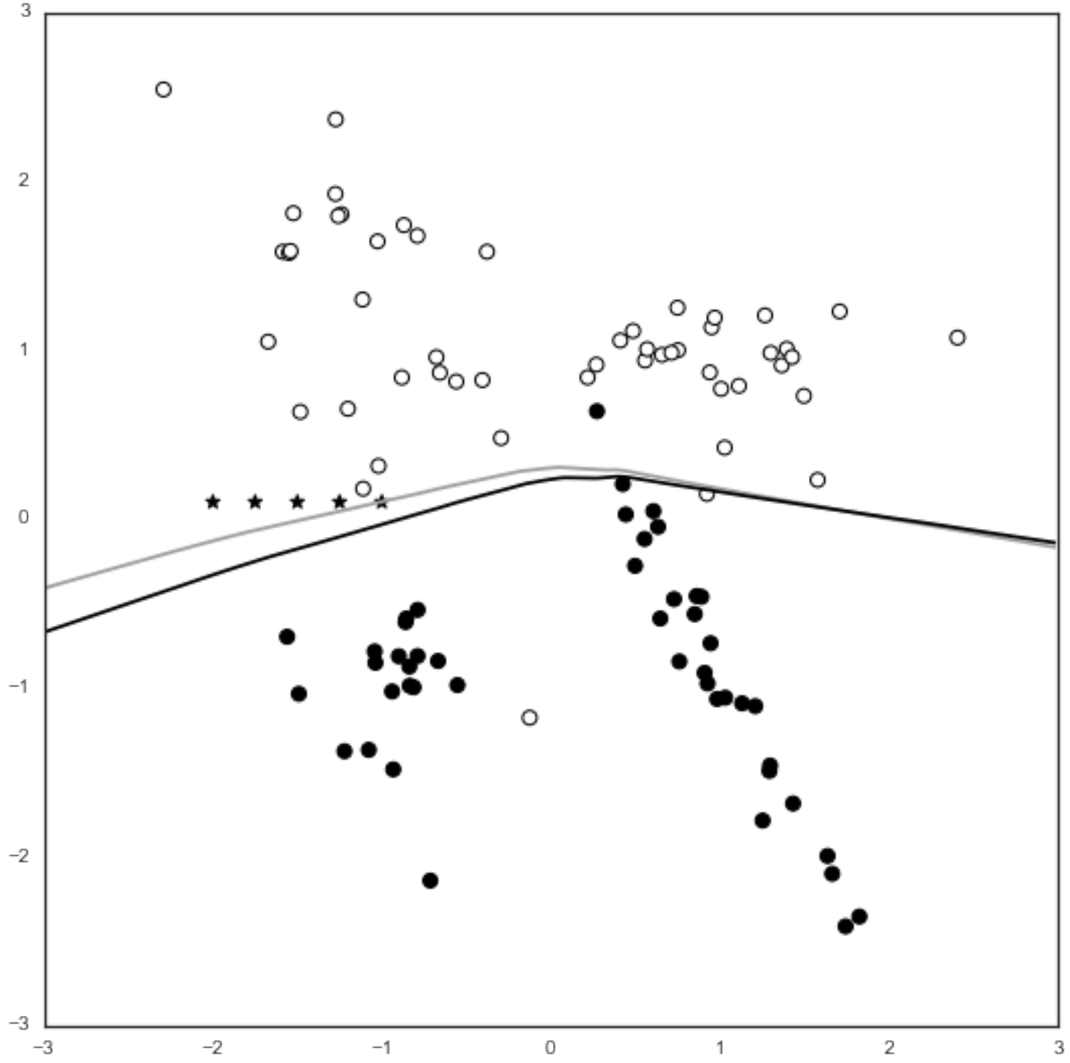
Clarence Chio, David Freeman. Machine Learning & Security. Pg. 322 (2018)

Poisoning Attack



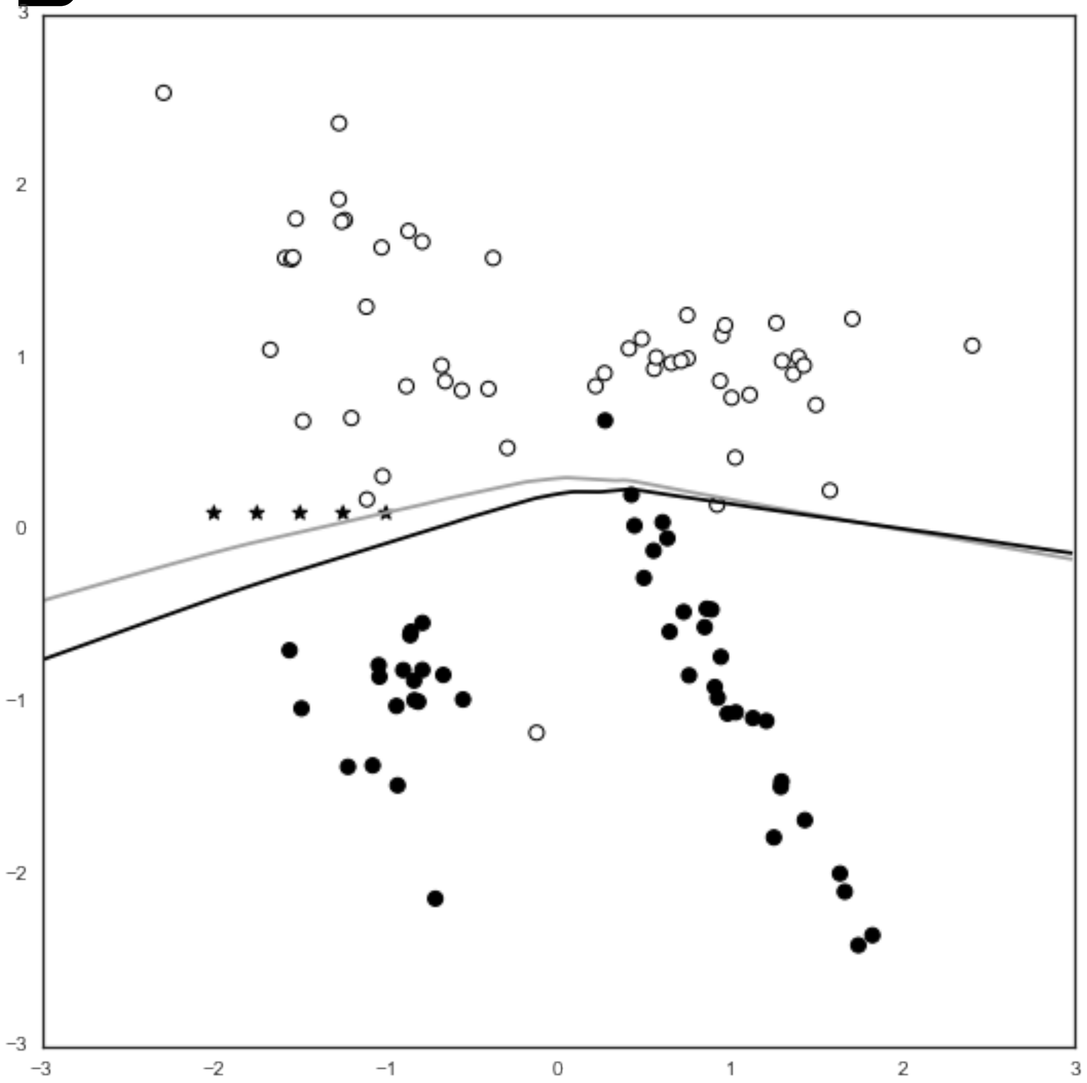
Clarence Chio, David Freeman. Machine Learning & Security. Pg. 322 (2018)

Poisoning Attack



Clarence Chio, David Freeman. Machine Learning & Security. Pg. 322 (2018)

Poisoning Attack



Clarence Chio, David Freeman. Machine Learning & Security. Pg. 322 (2018)

Poisoning Attacks

- Require access to either the predictions or the probabilities for an effective attack
- Longer periods between retraining
- Periodically analyzing retraining data to detect "boiling frog" attacks
- Avoiding real time online learning systems unless absolutely necessary

Additional Readings

- Alexey Kurakin et al. "Adversarial Examples in the Physical World" (2016)
- Anish Athalye et al. "Synthesizing Robust Adversarial Examples" (2017)
- Ivan Evtimov et al. "Robust Physical World Attacks on Machine Learning Models" (2017)
- Weilin Xu et al. "Automatically Evading Classifiers: A Case Study on PDF Malware Classifiers" (2016)

In Class Exercise

Please complete Worksheet 11: Attacking AI

Questions?