

# Trustworthy Machine Learning

Lecturer: Jingfeng Zhang

RIKEN-AIP

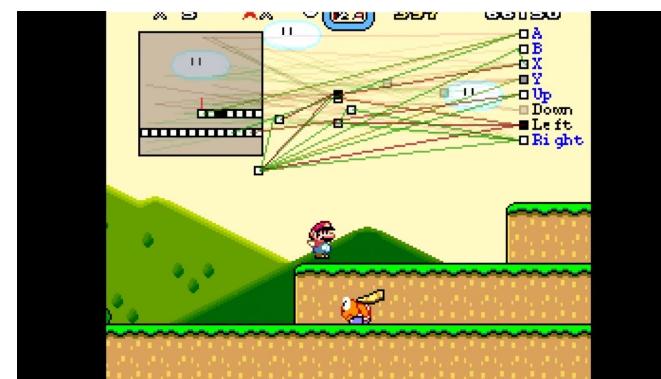
Homepage: <https://zjfheart.github.io>

# Machine learning (ML) models exceed human ability in many tasks.

## Image Classification



## Reinforcement Learning



## Natural language processing



Machine translation



Alexa, order me a large pizza!

# ML models are also in high-stake applications.



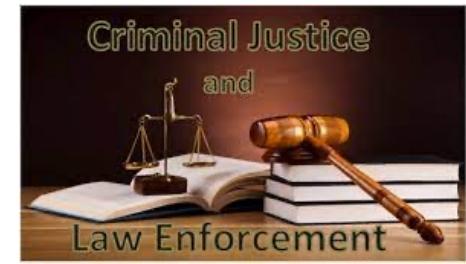
Education assessment



Credit



Health care



Criminal justice



Self-driving cars



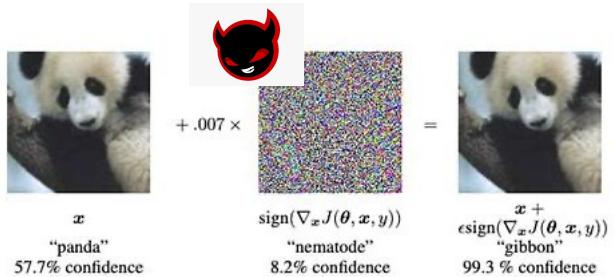
Robotic surgery



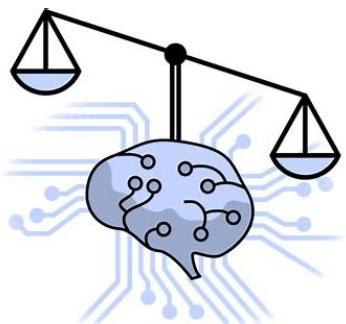
Content recommendations

ML models need TRUST!

# What is “trust” in ML?



Security



Fairness



Privacy



Interpretability

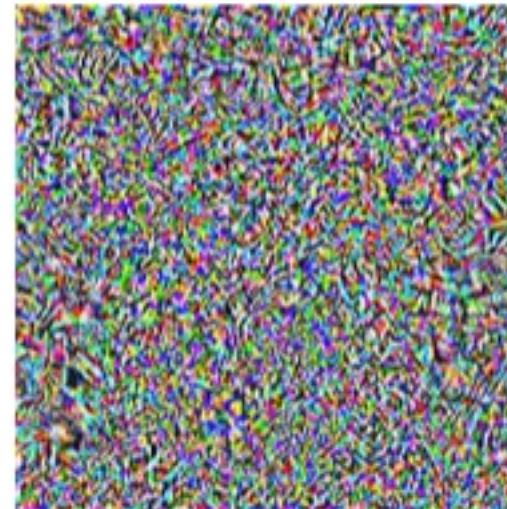
# An example---adversarial attacks!

**“pig” (91%)**



**+ 0.005 x**

**noise (NOT random)**



**=**

**“airliner” (99%)**



**AI makes the pig flying high!**

The images & the amusements come from Aleksander Madry's group.

# Examples---adversarial attacks pose **threat** to AI's deployment.

Glasses

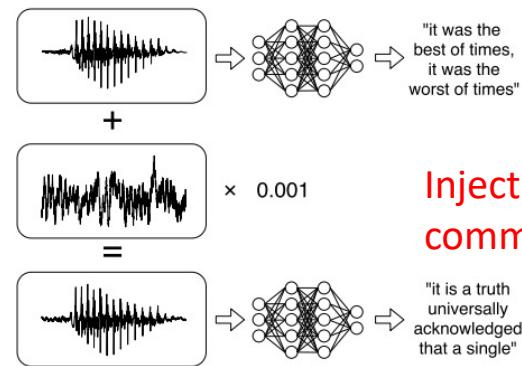


[Sharif Bhagavatula Bauer Reiter 2016]



Small stickers

[Eykholt Evtimov Fernandes Li Rahmati Xiao Prakash Kohno Song 2018]



Inject a hidden voice command

[Carlini Wagner 2018]



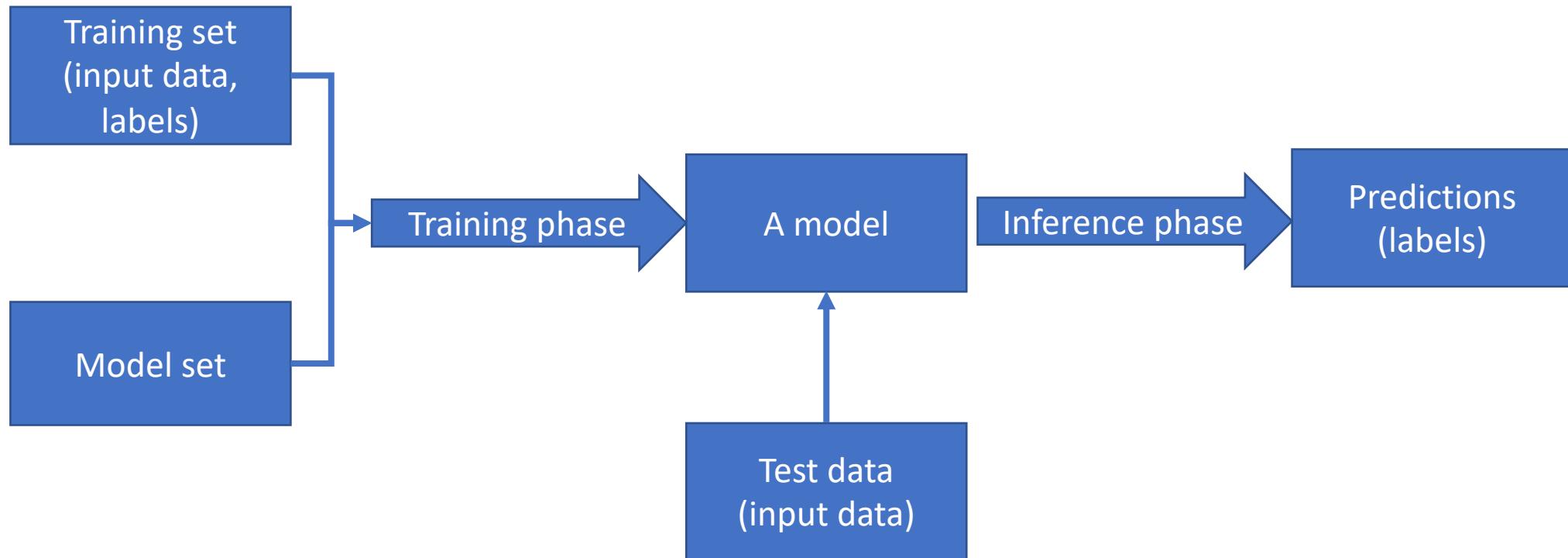
Add Human-imperceptible noises

[Mopuri Ganeshan Babu 2018]



THREAT!

# ML pipeline



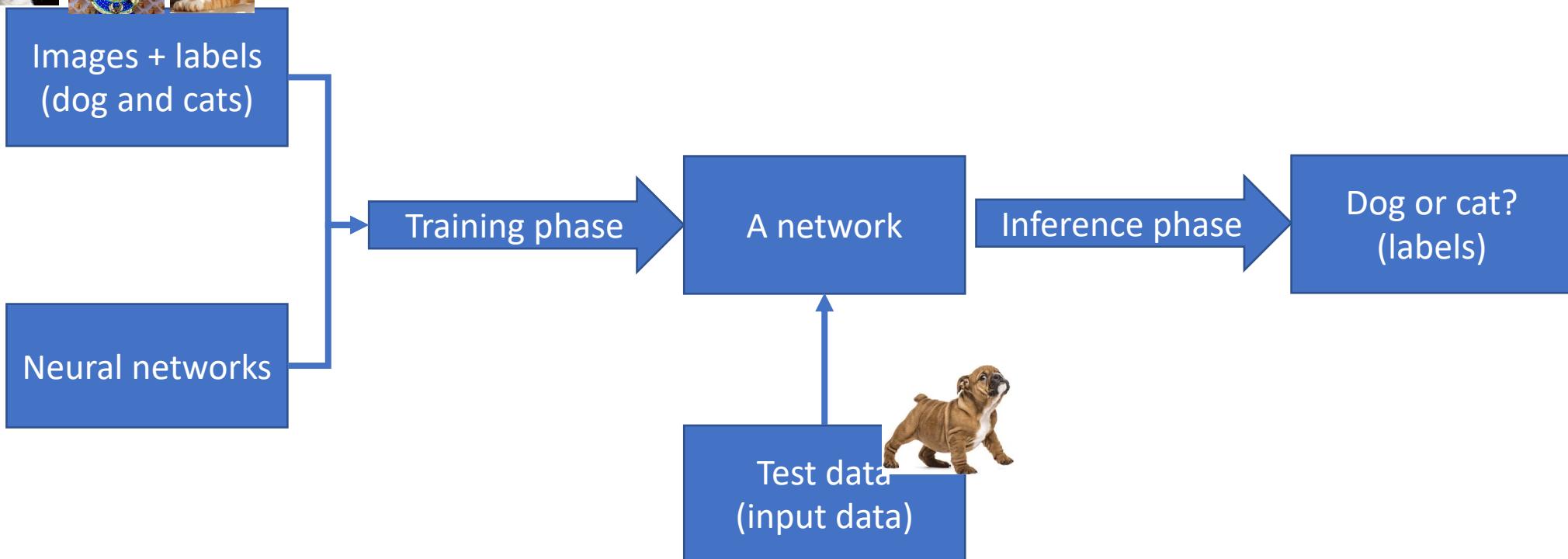
# ML for dog and cat classification



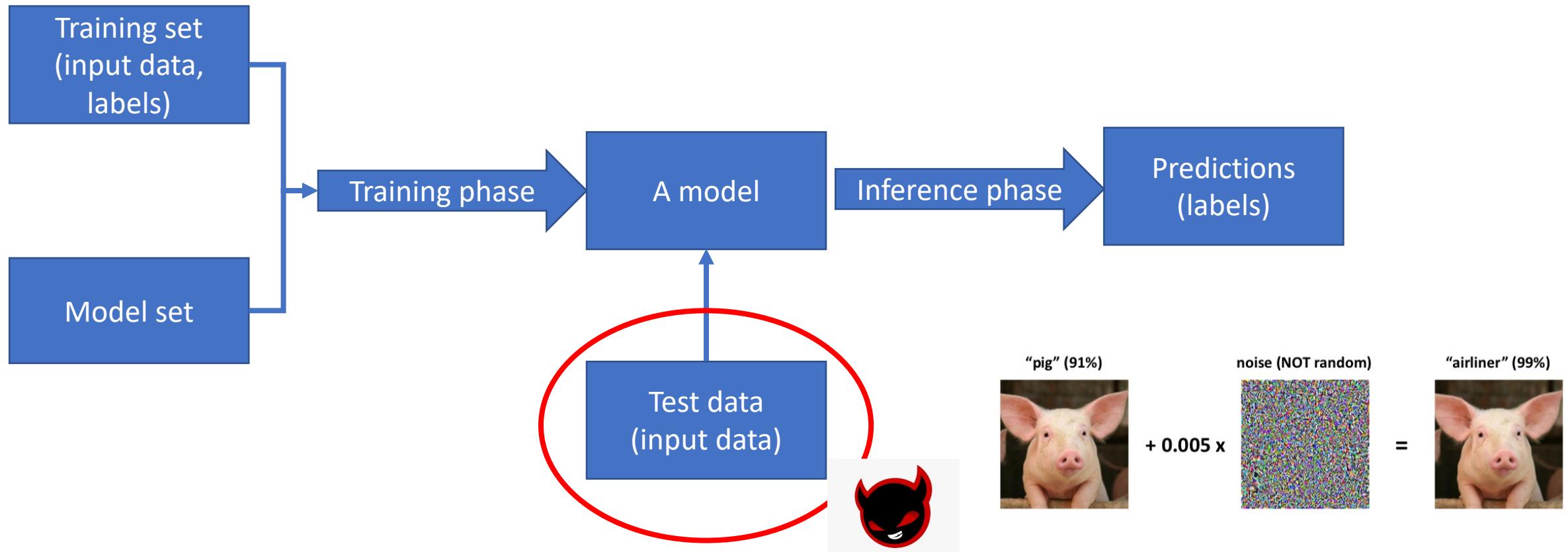
We labeled them as dog!



We labeled them as cat!



# Security: (Evasion) adversarial attack happens at inference phase

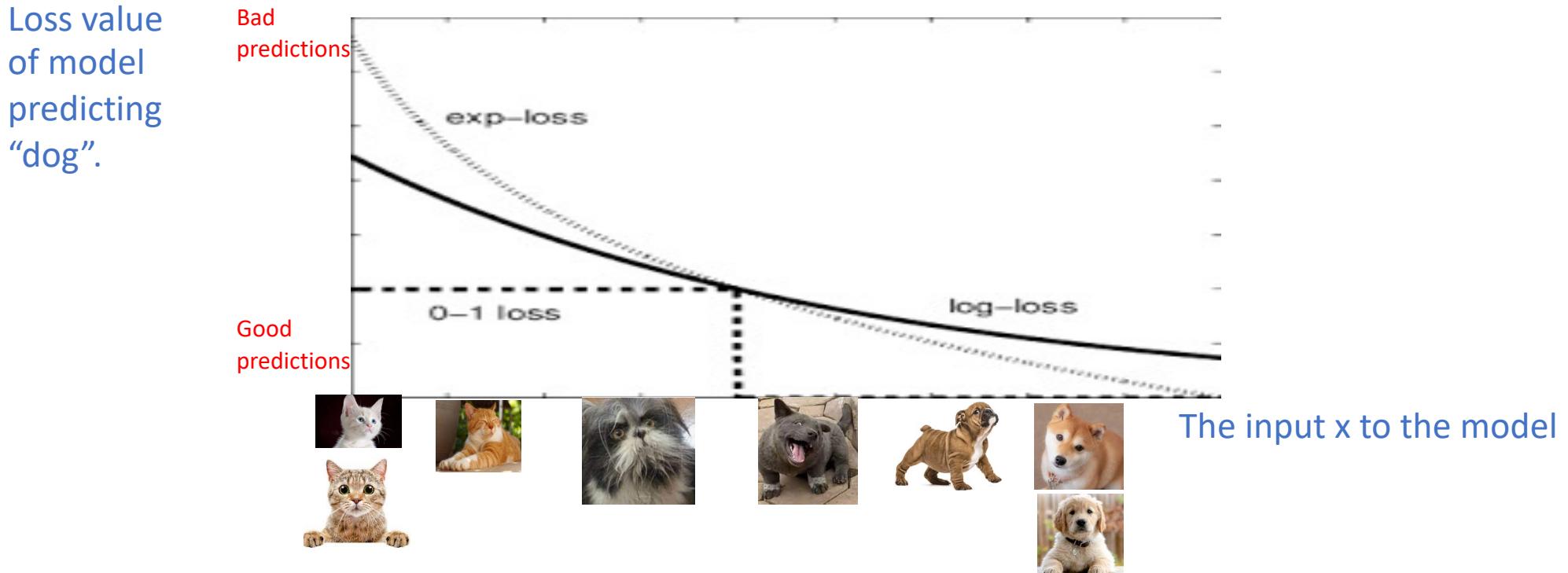


Adversarial attacker adds small (human-imperceptible) noise to test input data, which fools the model to make wrong predictions!

# The adversarial attack is against the model's will on the purpose! But what is model's will?

Let us use function  $f$  to denote model.

- What is model's will? Correctly label the test input data, i.e.,  $f(\text{dog}) = \text{"dog"}$ .
- Then, the model's will is to **minimize** the 0-1 loss  $\ell(f(\text{dog}), \text{"dog"})$ .



- In ML, we usually use the smoothed loss function, i.e.,  $\ell(f(x), y)$ , to upper bound the 0-1 loss. For example, log-loss and exp-loss can be differentiable!

# $L_p$ -norm bounded adversarial attacker: **maximize** the model loss!

## Attacker Objective:

$$\tilde{x} = \operatorname{argmax}_{\tilde{x} \in B_\epsilon(x_i)} \ell(f(\tilde{x}), y)$$

Find an **adversarial data**  $\tilde{x}$  within the  $L_p$  norm ball  $B_\epsilon(x)$  of **natural data**  $x$  that maximizes the loss  $\ell(f(\tilde{x}), y)$  within the norm ball constraint  $\epsilon$ .

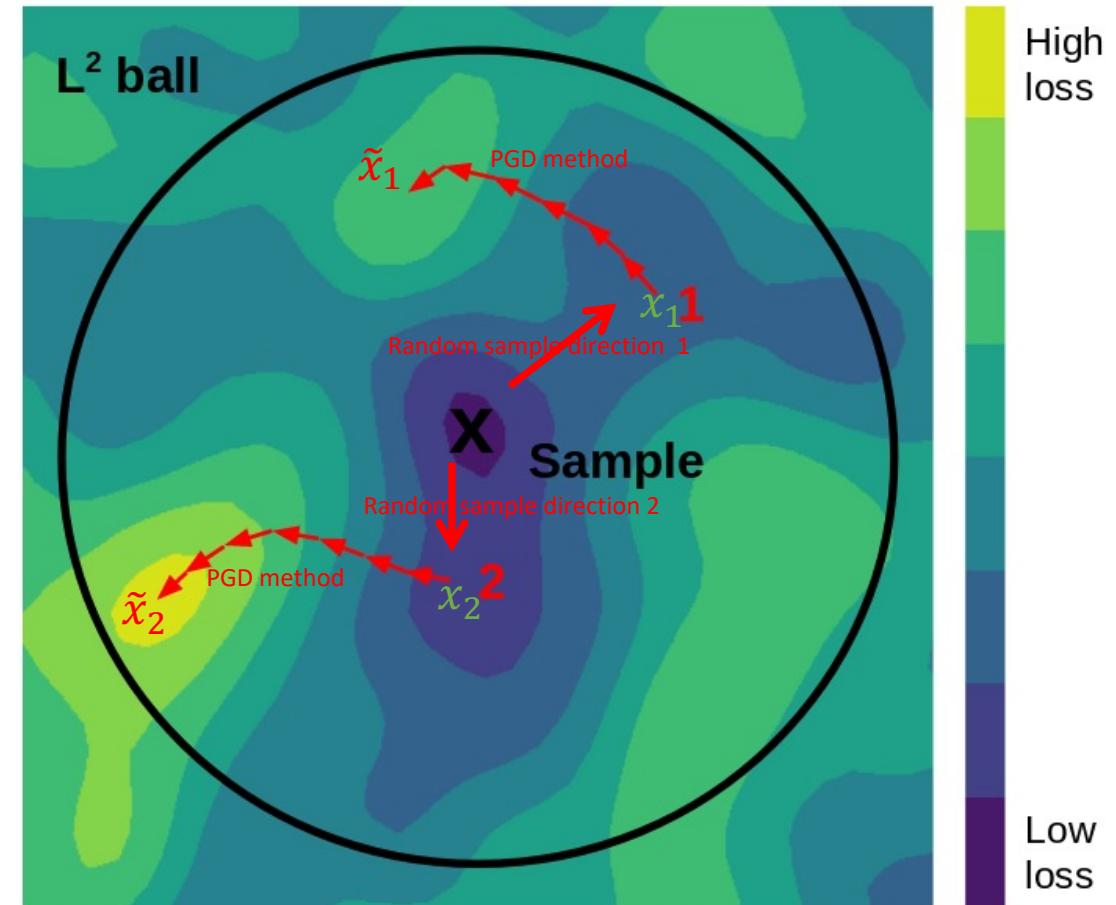
## A Typical Method:

**Projected gradient descent (PGD)** –given a starting point  $x^{(0)}$  and step size  $\alpha$ , PGD works as followed:

$$x^{(t+1)} = \Pi_{B(x^{(0)})} \left( x^{(t)} + \alpha \operatorname{sign} \left( \nabla_{x^{(t)}} \ell(f_\theta(x^{(t)}), y) \right) \right), t \in N$$

$\Pi_{B(x^{(0)})}$  projects adversarial data  $x^{(t)}$  back onto the norm ball if  $x^{(t)}$  exceeds the norm ball boundary;  $\alpha$  is a small step size;  $t$  is searching step numbers.

Images modified from <https://towardsdatascience.com/know-your-enemy-7f7c5038bd3>



# Different types of adversarial attacks

- Human imperceptible attacks,  
e.g., attackers use norm bound to measure  
imperceptability such as  $L_{\infty}$ ,  $L_2$  norm,  
Wasserstein norm.

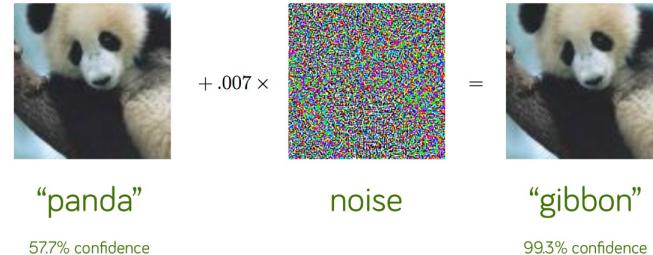


Image taken from <https://towardsdatascience.com/breaking-neural-networks-with-adversarial-attacks-f4290a9a45aa>

- Patch-based attacks. e.g.,  $L_0$  norm.

Image taken  
from [https://www.cs.cmu.edu/~sbhagava/  
papers/face-rec-ccs16.pdf](https://www.cs.cmu.edu/~sbhagava/papers/face-rec-ccs16.pdf)



Image taken from <https://arxiv.org/pdf/1712.09665.pdf>

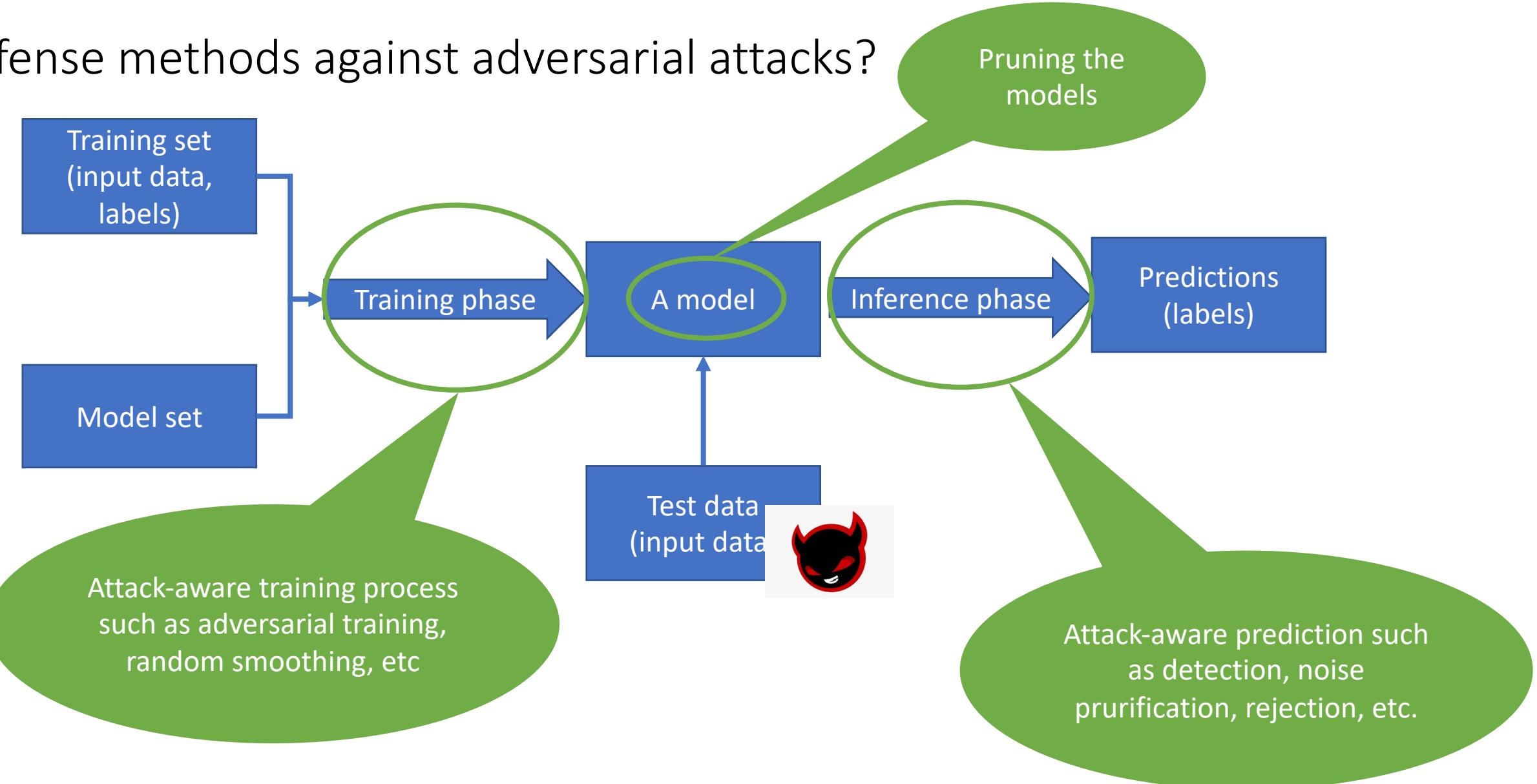
Others, such as rotation attacks, out-of-distributions attacks, etc

What if attacker is not allowed to access model's parameter?

- Black-box attacker: query the model's predictions only.
- Grey-box attacker: Know some training data.  
Train a substitute model.  
Perform the transfer-based attacks.

Reading: [Papernot et al., Practical Black-Box Attacks against Machine Learning.](#)

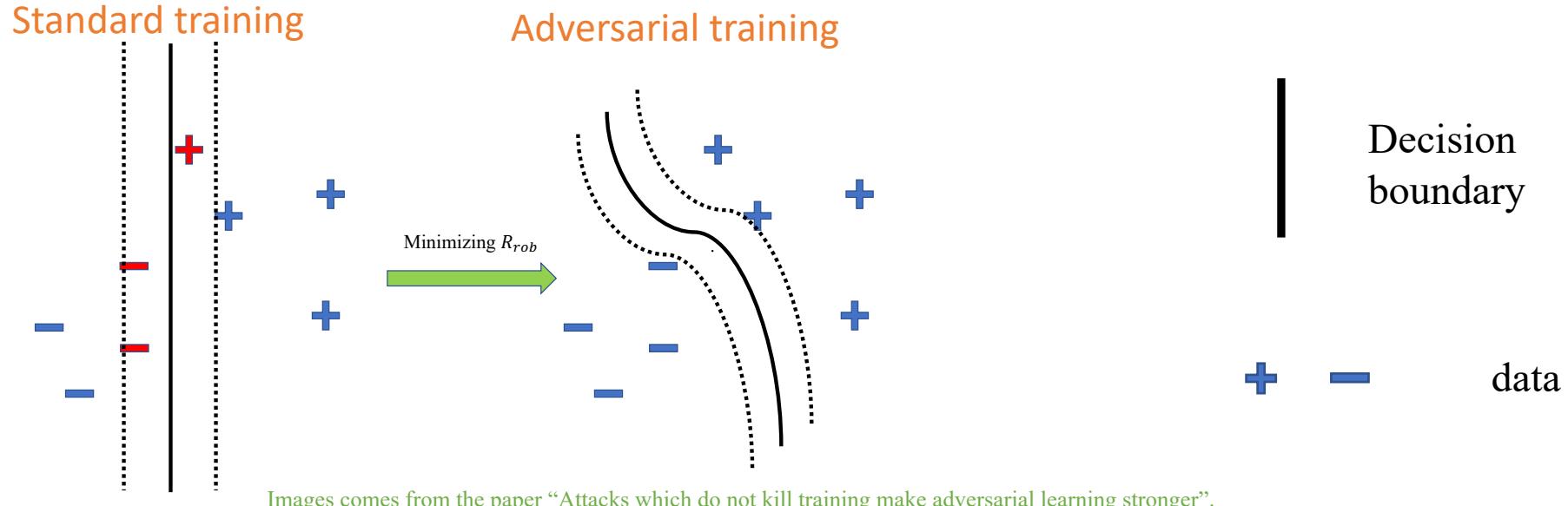
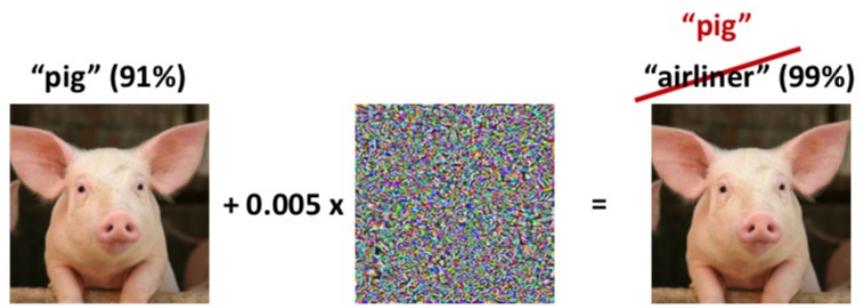
# Defense methods against adversarial attacks?



# One defense example: adversarial training (AT)

Given the knowledge that the test data may be adversarial,  
AT carefully *simulates some adversarial attacks during training*.

Thus, the model has already seen many adversarial training data in the past,  
and hopefully it can generalize to adversarial test data in the future.



**AT's Purpose 1:** correctly classify the data.

**AT's Purpose 2:** make the decision boundary thick so that no data lie nearby the decision boundary.

# AT's basic formulations and the corresponding AT's improvements

Minimax formulation:

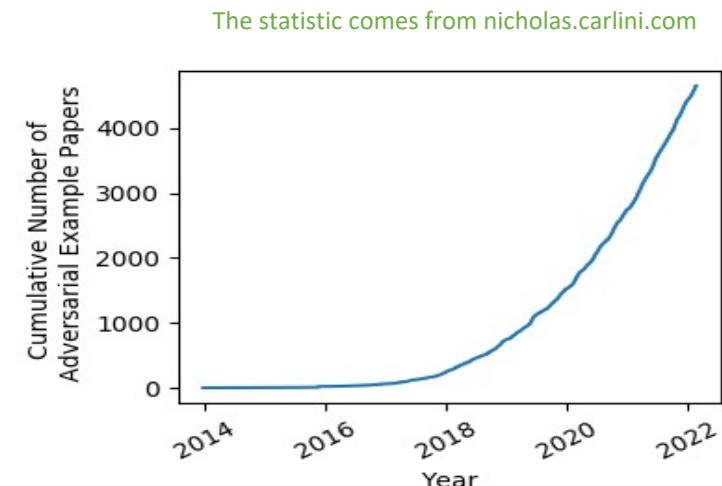
$$\min_f \frac{1}{n} \sum_{i=1}^n \ell(f(\tilde{x}_i), y_i), \text{ where } \tilde{x}_i = \operatorname{argmax}_{x \in B_\epsilon(x_i)} \ell(f(x), y_i)$$

Outer minimization                                    Inner maximization

[Madry Kakelov Schmidt Tsipras Vladu 2019]

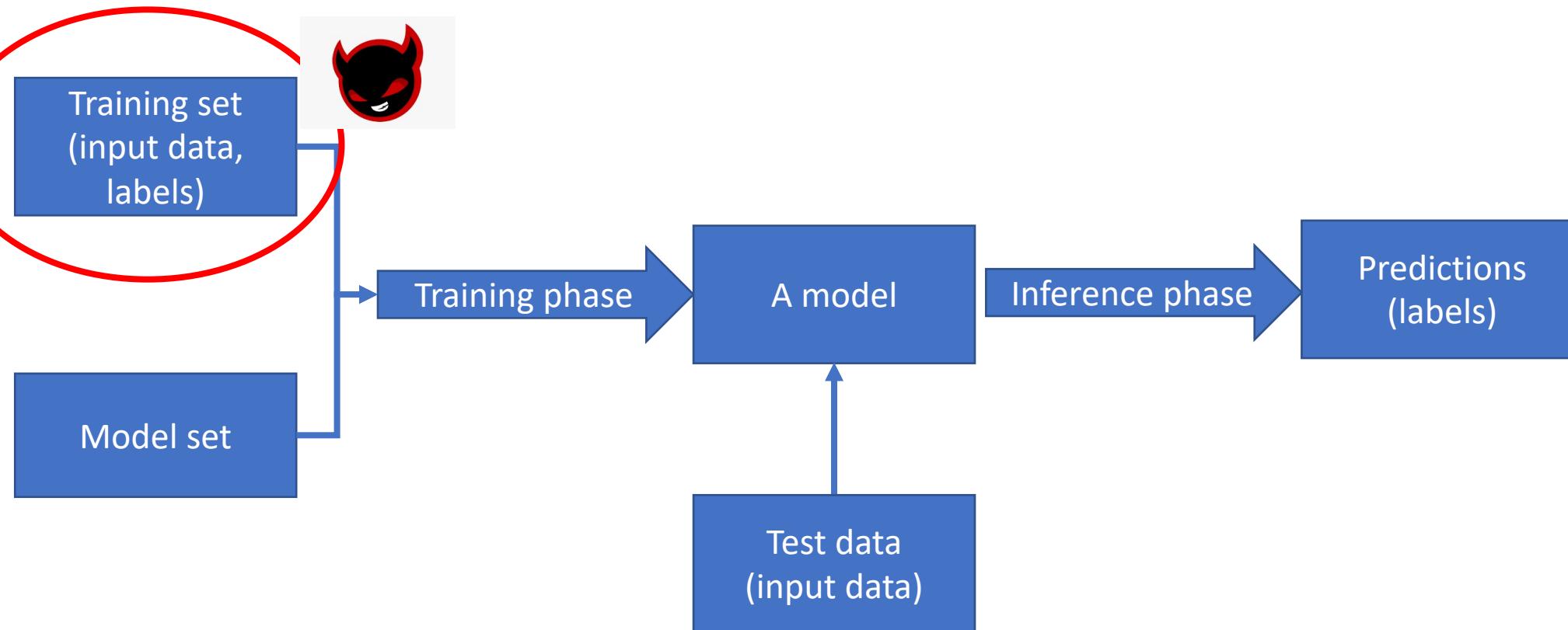
AT's improvements/modifications, intriguing findings & interesting applications

- 1 Collecting/generating more/smarter training data
- 2 Simulating smarter attacks
- 3 Designing smarter learning objective
- 4 Designing/learning smarter network structures
- 5 Leveraging smarter tricks
- 6 Discovering some intriguing findings
- 7 Developing some applications
- 8 Other directions such as smarter attacks, detections.



Refer to a video: <https://www.youtube.com/watch?v=3Z8bUgn41Fk>

# Security: (Poisoning) attack happens at training phase

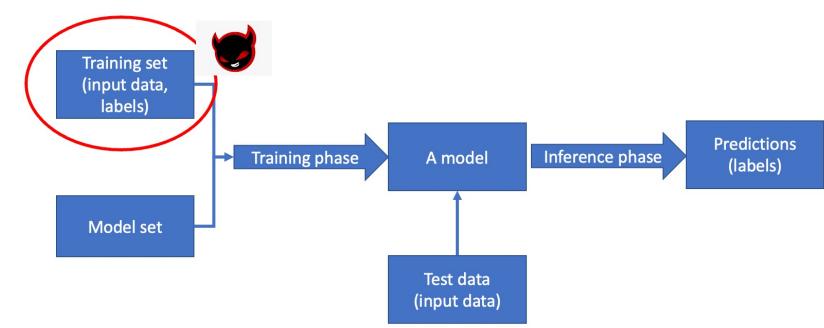


Adversarial attacker adds small (human-imperceptible or human-perceptible) noise to training data, which fools the training phase to generate the “bad” model!

The attacker is against the learning's will on the purpose.

- In the previous slides, the model is denoted as a function  $f: x \rightarrow y$ .
- Similarly, the learning is also denoted as function  $A: D \rightarrow f$ , in which  $D$  is a training dataset, and  $f$  is a model.
- What is the learning's will? **Usually**, return a good model that has small **natural** generalization loss, i.e.,  $E_{x \sim D}[\ell(f(x), y)]$ .
- **Sometimes**, it also needs a different will---small **robust** generalization loss (for security purpose), i.e.,  $E_{x \sim D}[\max_{\tilde{x} \in B_\epsilon(x)} \ell(f(\tilde{x}), y)]$ , where  $B_\epsilon$  is  $\epsilon$  norm ball.

# What can the poisoning attacker do?



Flipping labels  
(naturally noisy labels,  
or flipping-label  
attacks)

Reading: Biggio et al., Support vector machines under adversarial label noise.  
Zhao et al., Efficient label contamination attacks against black-box learning models

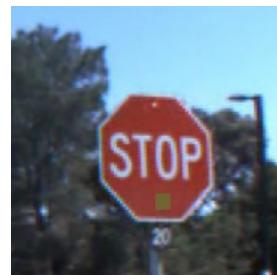
Facilitate

Label-noisy  
robust training  
research



Perturbing input  
data

Backdoor  
Trigger

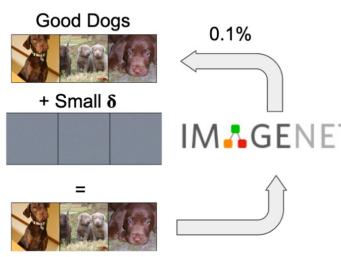


After training



Images come from Gu et al., BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain

Clean-label  
targeted attack



After training

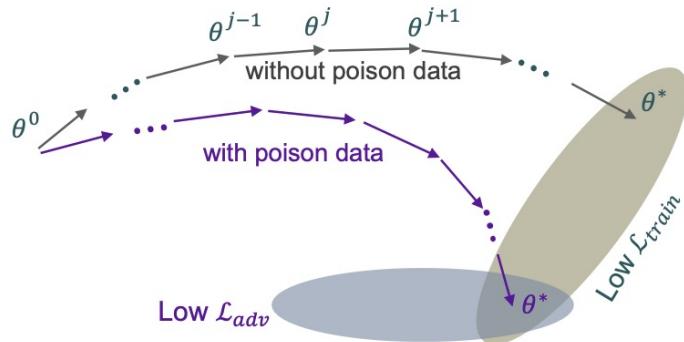


Oh! This is a dog! (wrong prediction)

Images come from Geiping et al., WITCHES' BREW: INDUSTRIAL SCALE DATA POISONING VIA GRADIENT MATCHING

# One poisoning example---clean-label targeted attack

- Attacking a learning algorithm is more challenging!



It is not just fooling a single model (such as adversarial attack), but fooling a series of models in the learning sequences.

The learning algorithm A converges to a bad model region!

The image comes from Huang et al, MetaPoison: Practical General-purpose Clean-label Data Poisoning.

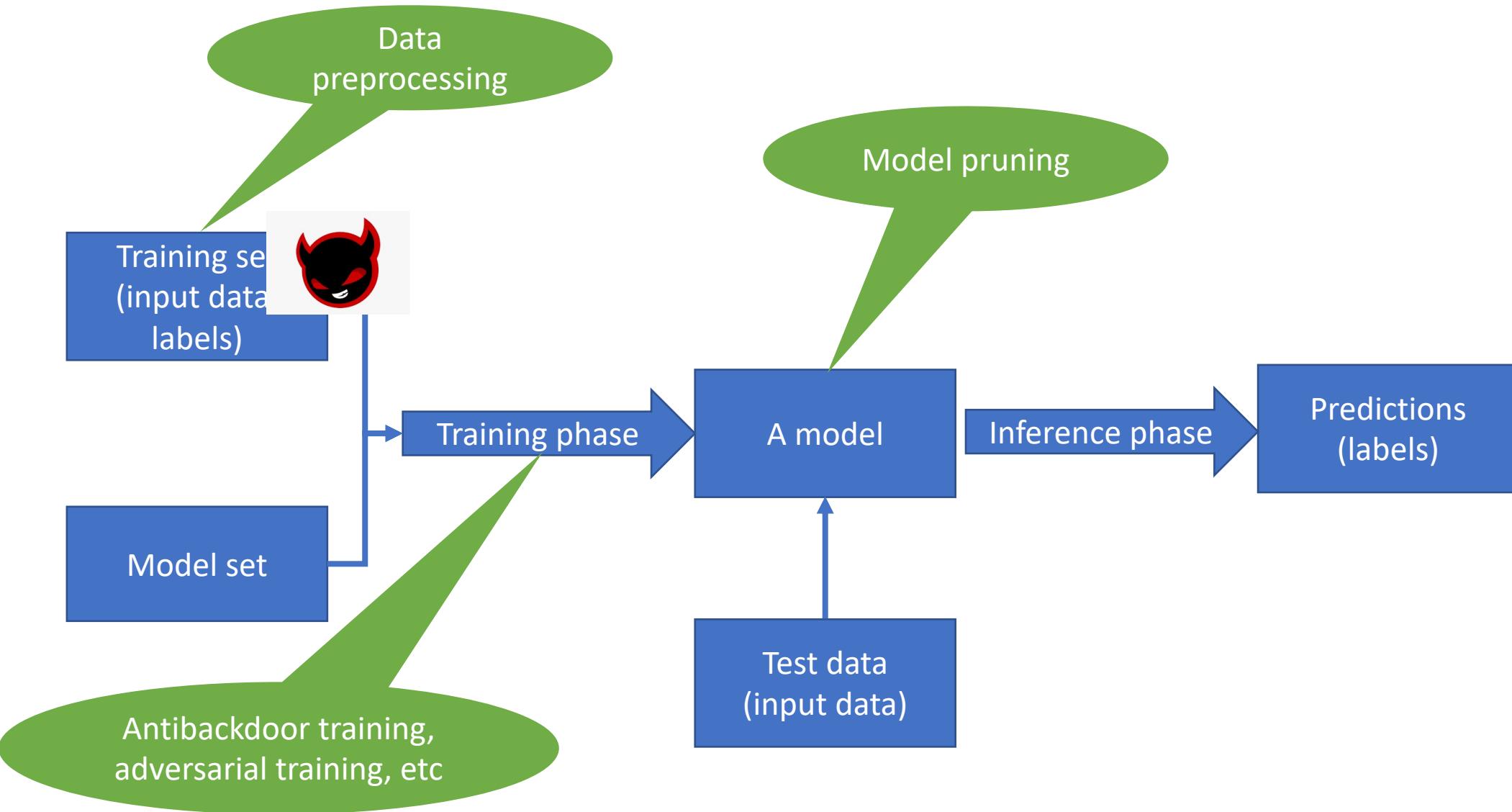
- What is clean-label targeted attack?

- 1 poisoned data (e.g., images) appear to be unmodified and labeled correctly.
- 2 The perturbed images often affect classifier behavior on *a specific target instance ( $x_{tar}$ )* of a learned model, without affecting behavior on other inputs,
- 3 The clean-label attacks are insidiously hard to detect.

## clean-label targeted attack

- Performing poisoning attack has to unroll the whole training process (constrained bilevel optimization), which is computationally intractable and costly!
- **Then how?** Just use **a single model** (a pretrained feature extractor) to present all!
- Feature collision:  $x_{poi} = \operatorname{argmin}_x [\|f(x) - f(x_{tar})\|^2 + \beta \|x - x_{nat}\|^2]$ , where  $x_{poi}$  is generated poisoned data,  $x_{tar}$  is a *specific* target instance in the test dataset,  $x_{nat}$  is original benign data.  
Shafahi et al. Poison frogs! targeted clean-label poisoning attacks on neural networks
- Gradient alignment (Witches Brew): Matching gradients between poisoned data and target data.  $x_{poi} = \operatorname{argmin}_{x_{poi} \in B(x_{nat})} \text{ML}[\nabla_\theta L(f(x_{tar}), y_{adv}), \nabla_\theta L(f(x_{poi}), y_{true})]$ , where ML is similarity loss, such as cosine similarity  $(a, b) = \frac{a \cdot b}{\|a\| \|b\|}$ ;  $y_{adv}$  is attacker-chosen label (wrong).  
Geiping et al., WITCHES' BREW: INDUSTRIAL SCALE DATA POISONING VIA GRADIENT MATCHING

# Defense against poisoning attacks



# Privacy

Two different notions of privacy.

- Protect data privacy from **machine**.  
How to achieve this? Data poisoning!

Reading: Zhiqi et al. Human-imperceptible privacy protection against machines, ACM MM 19 best paper award  
Huang et al. Unlearnable examples: Making personal data unexploitable, ICLR21 Spotlight

- Protect data privacy from **people**. How to achieve this?

# A head-scratching questionnaire!

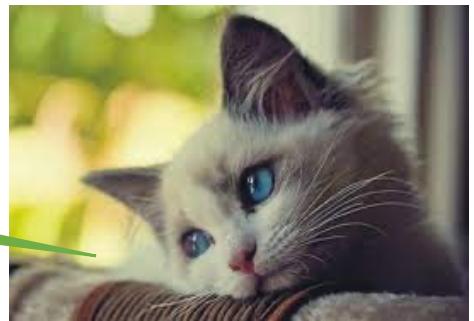
- Suppose you want to collect answers of a very embarrassing question, for example, whether you conduct improper behaviors on the train in the past three months. (Yes/No)



How?

This question is important on the population level, but very embarrassing on the individual level. Therefore, people tend to **lie** in this question.

What can I do to get  
the true statistics?



# We need a private learning process!

- We introduce randomness, i.e., plausible deniability for each individual.

- Step 1: The subject individual flips a coin twice.

- Step 2:

- a. If first coin was tail, report true answer.

- b. report YES, if second coin heads; report NO, if second coin tails.



We collect  $N$  samples, in which  $N_{yes}$  and  $N_{no} = N - N_{yes}$ .

We want to calculate the true estimated portion  $P$  of people who conduct improper behaviors. **How?**

# Differential privacy

We collect N samples, in which  $N_{yes}$ ;  $N_{no} = 1 - N_{yes}$ .

We want to calculate the true estimated portion P of people conducting improper behaviors. **How?**

First\second	Head	Tail
Tail	True answers	True answers
Head	Yes	No

People who truly commit crime (P) have 3/4 chances to report “Yes”, i.e.,  $\frac{3}{4} P$ .

	Head	Tail
Tail	True ★	True ★
Head	Yes ★	No

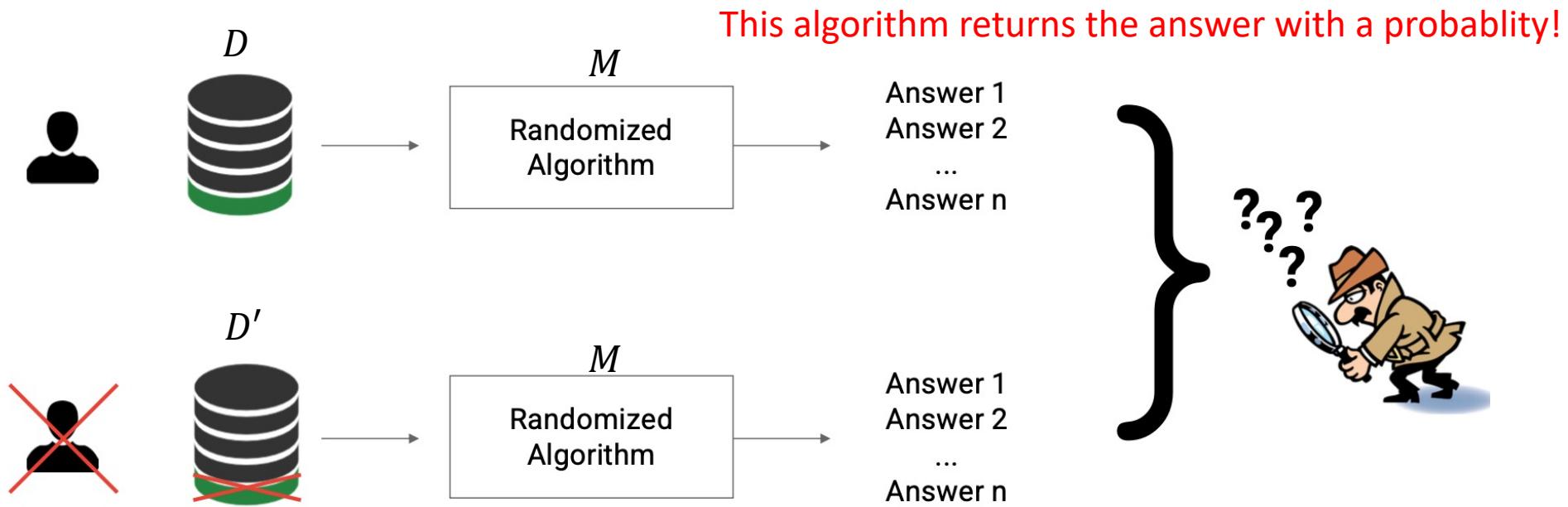
People who do not commit crime (1-P) have 1/4 chances to report “Yes”, i.e.,  $\frac{1}{4} (1 - P)$ .

Answer:

$$\frac{3}{4} P + \frac{1}{4} (1 - P) = \frac{N_{yes}}{N}$$

	Head	Tail
Tail	True	True
Head	Yes ★	No

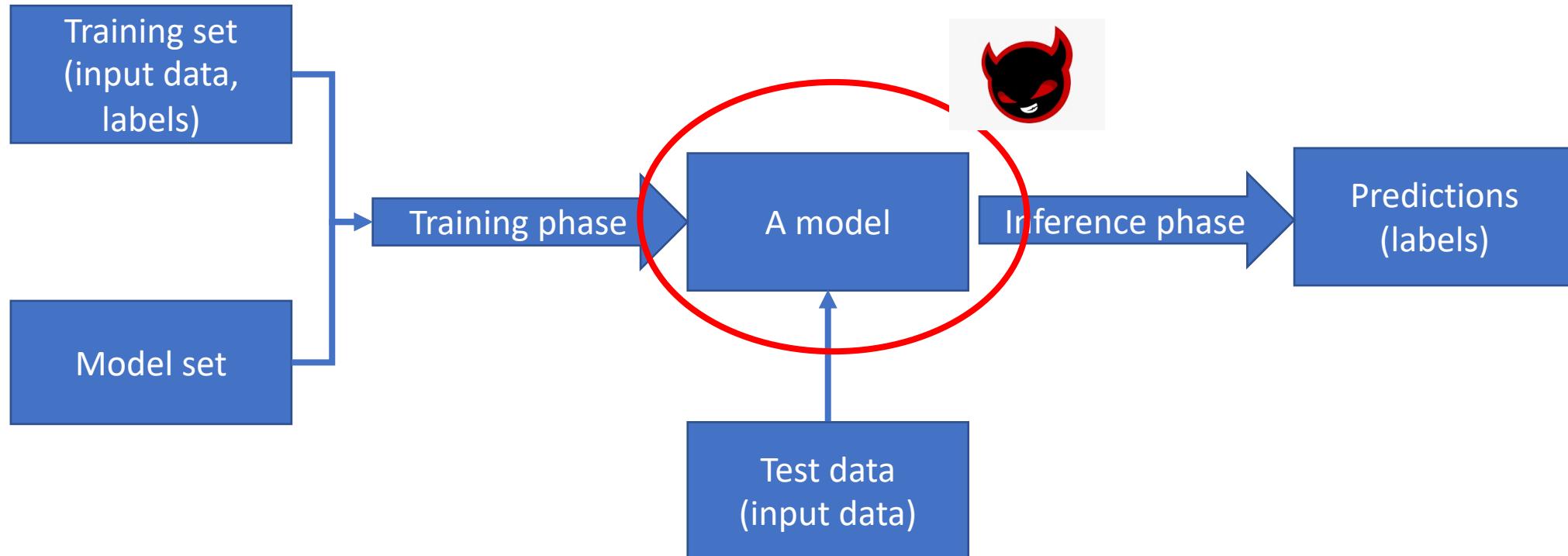
What is differentially private algorithm? --- a randomized algorithm.



$$\Pr[M(D)] \leq e^\epsilon \Pr[M(D')]$$

$D$  and  $D'$  has only one record different!

# Examples of privacy attacks in ML



Model inversion attack: Given a trained model, **recover the private dataset** used to train the model.



Fredrikson et al. Model inversion attacks that exploit confidence information and basic countermeasures

Membership inference attack: Given a trained model, **detect whether the data is used** to train the model.

Reza et al., Membership Inference Attacks against Machine Learning Models

# Fairness---various descriptions

- Proportional fairness: You get what you deserve.

Reading: Zhang et al. [Hierarchically fair federated learning](#), a tech report.

*A model may have bias towards sensitive attributes, such as gender, race, religion.*

- Individual fairness: Two similar individuals should be classified similarly.
- Group fairness: Model's outcome should be the same across different groups.

For example, there exists demographic parity:  $P(\text{guilty} | \text{black}) \neq P(\text{guilty} | \text{white})$ .

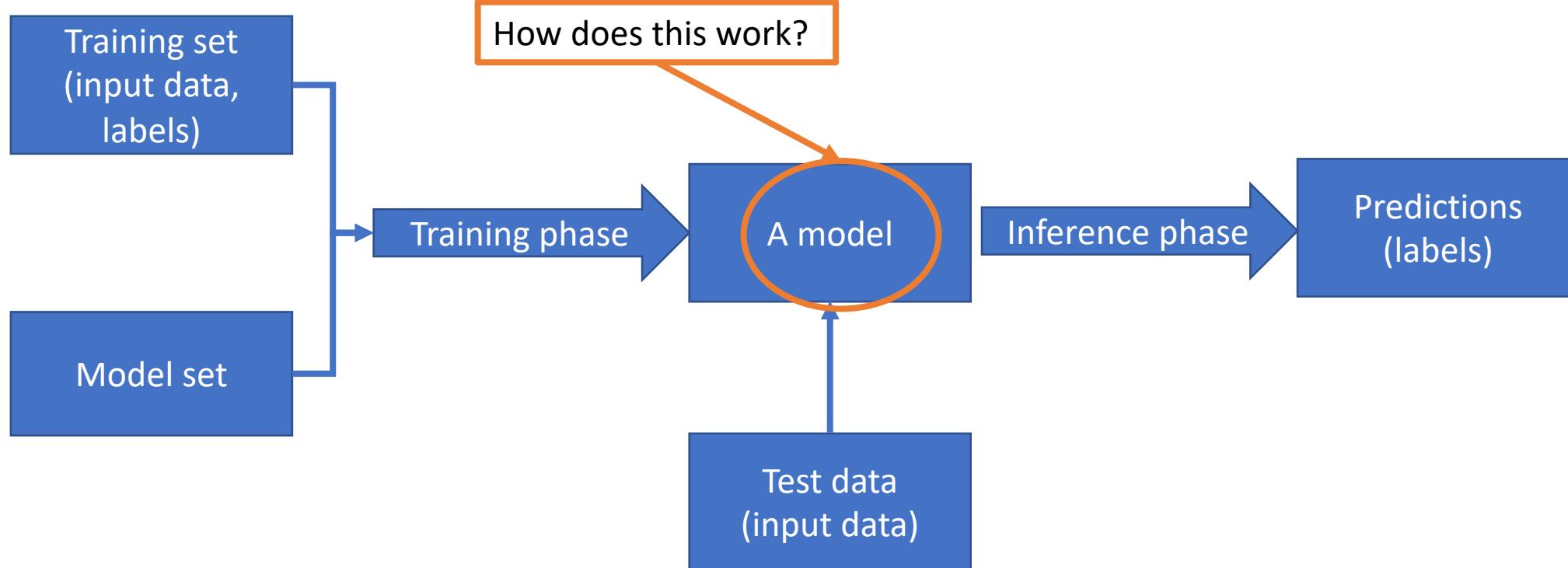
Reading: 1 [Dwork et al., Fairness Through Awareness](#).

2 [Barocas et al](#), Fairness and Machine Learning: limitations and opportunities, <https://fairmlbook.org>



[COMPAS](#) software  
used in US courts

# Interpretability—how to explain a ML model to human



What is interpretability? Understand how the **model** works towards a task.

# Interpretability---two example descriptions

How certain attributes influence the predictions? (saliency maps)

Test input      Attention map



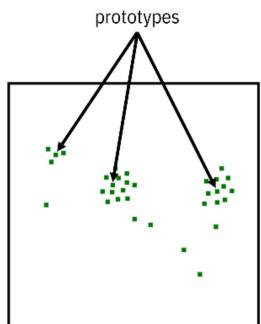
Mobile home  
(incorrect prediction)



Palace  
(incorrect prediction)



How certain training examples influence the predictions? (prototype)



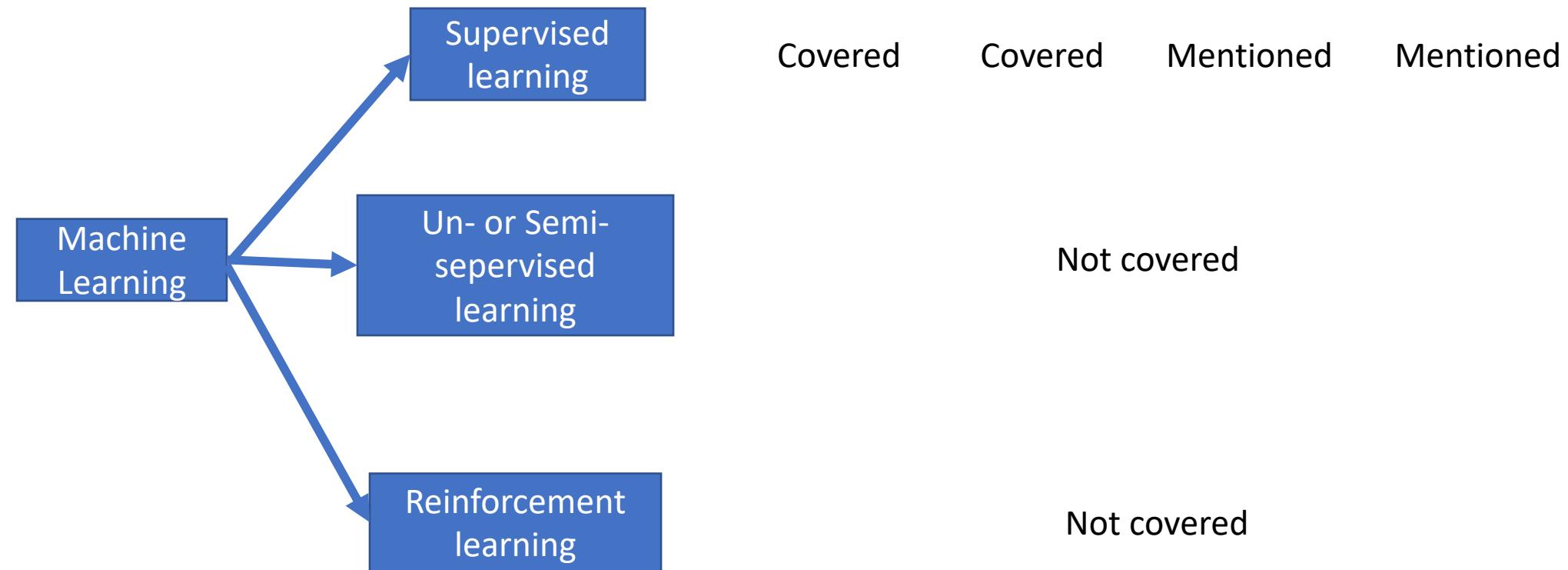
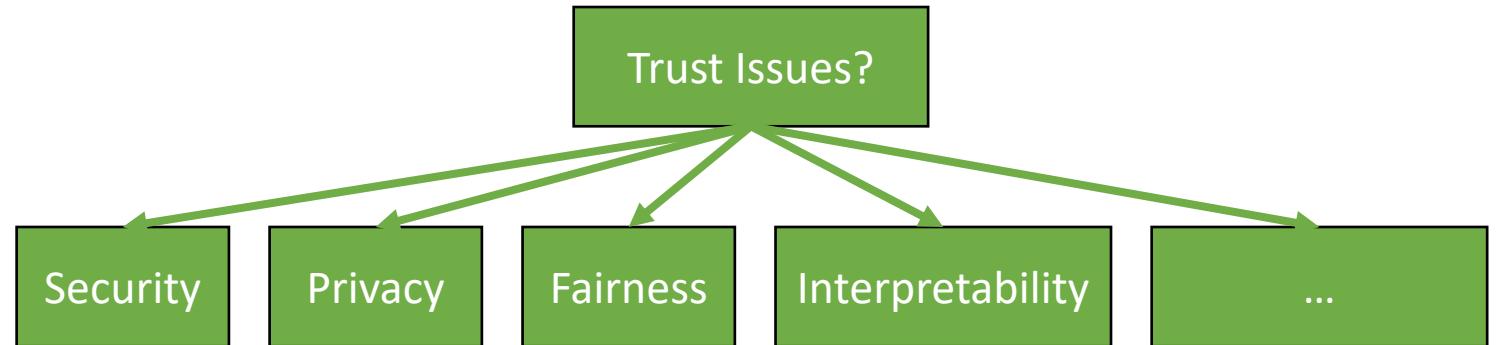
Test input



Most influential training images



# This lecture's scope



# Homework (10 points)

- 1 Write 1-2 pages essay (5 points).

Describe an ML application in the real world and discuss its “Trust” issues.

-Evaluation metric: clarity (2 points), relation to “Trust” (2 points), “wow!” factor (1 point).

- 2 Try coding! (5 points)

Run python code in the github <https://github.com/zjfheart/Friendly-Adversarial-Training>

-Use “smallcnn” network structure! E.g., specifying --net “smallcnn”

-Only run “python FAT.py”

-If you have GPUs, run CIFAR-10; if you do not have GPUs, modify code to run the MNIST dataset.

Report adversarial training’s results of natural accuracy and robust accuracy of  $\epsilon = \frac{2}{255}, \frac{4}{255}, \frac{8}{255}$  on CIFAR-10 or results of  $\epsilon = 0.1, 0.2, 0.3$  on MNIST. The  $\epsilon_{train} = \epsilon_{test}$  is specified as  $L_\infty$  norm bound.

# References

- 1 ECE1784H/CSC2559H: Trustworthy Machine Learning by Nicolas Papernot. <https://www.papernot.fr/teaching/f21-trustworthy-ml.html>
- 2 Varshney, K. R. (2022). Trustworthy Machine Learning. Independently Published. <http://www.trustworthymachinelearning.com>.
- 3 Solon Barocas and Moritz Hardt and Arvind Narayanan (2021). Fairness and Machine Learning. <http://www.fairmlbook.org>
- 4 Christoph Molnar (2022). Interpretable Machine Learning: A Guide For Making Black Box Models Explainable. Independently published. <https://christophm.github.io/interpretable-ml-book/>