Adversarial Sample Detection for Deep Neural Network through Model Mutation Testing

Jingyi Wang[1,4]

With Guoliang Dong[2], Jun Sun[3,4], Xinyu Wang[2], and Peixin Zhang[2]

1 National University of Singapore

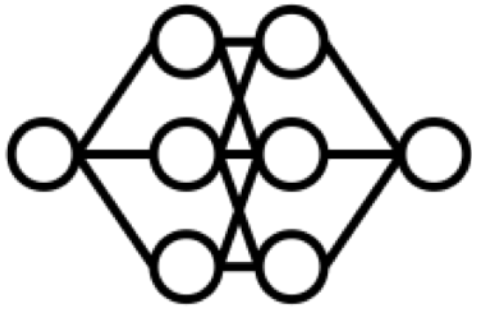
2 Zhejiang University

3 Singapore Management University

4 Singapore University of Technology and Design

ML achieves human-level performance

Deep learning models



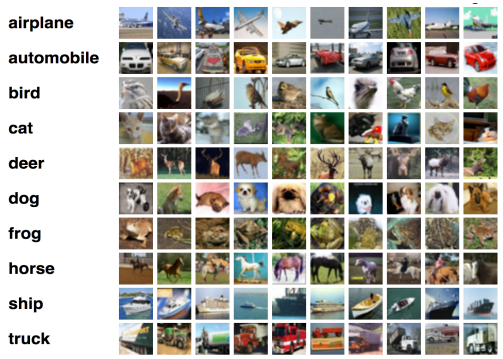


image classification

machine translation

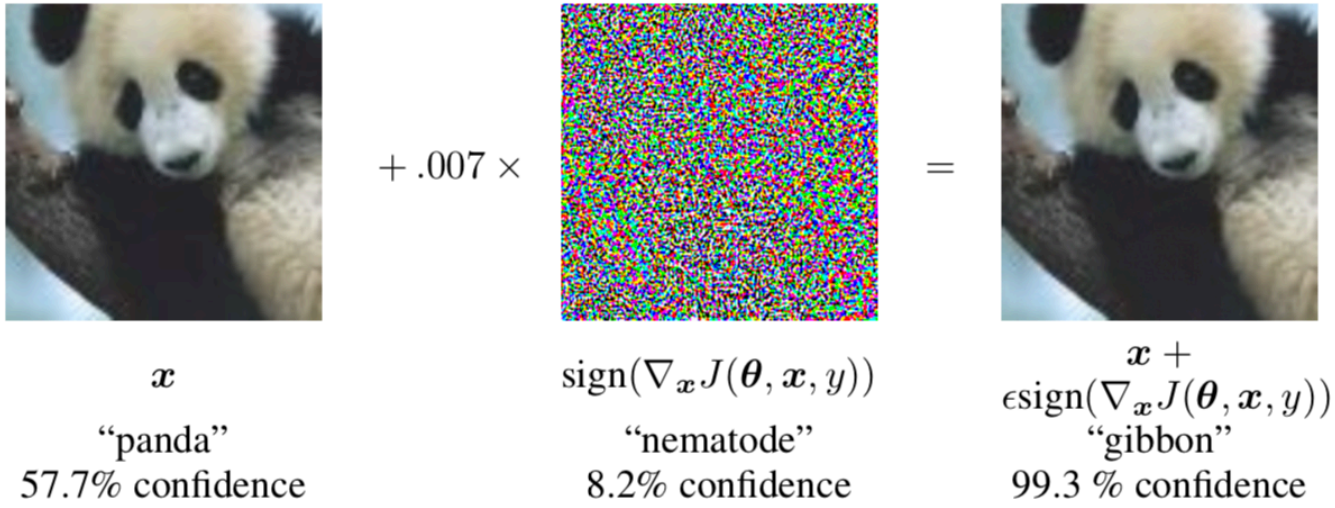
game playing

malware detection

Safety threats

Can make mistakes

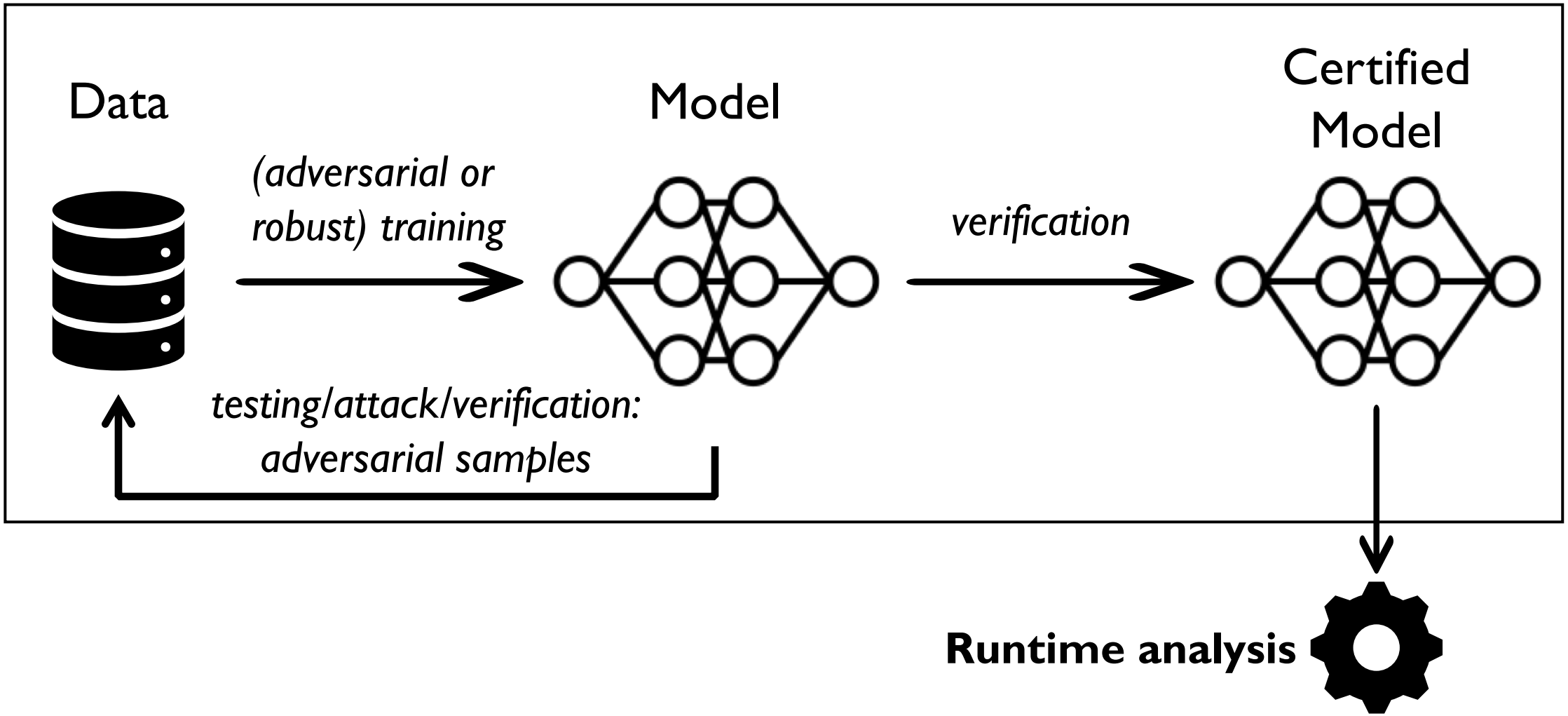
Vulnerable to adversarial attacks



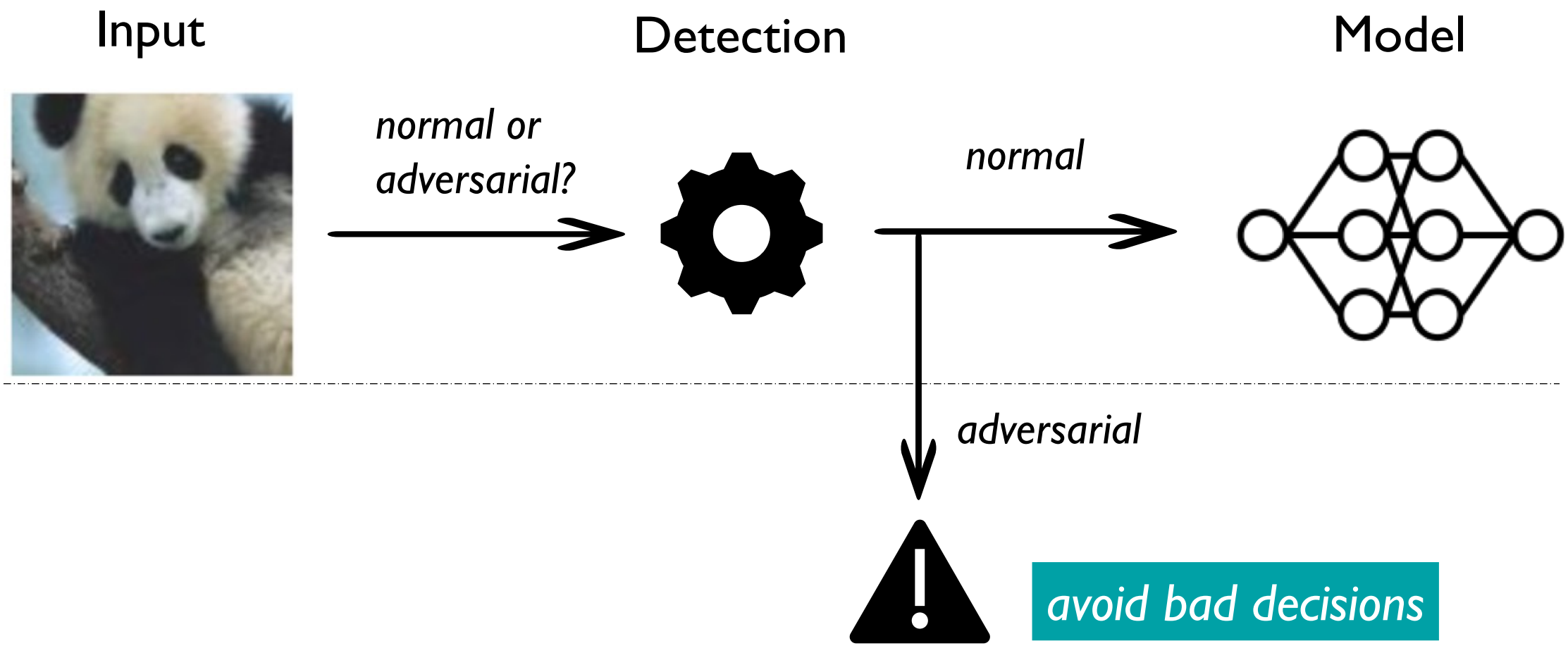
https://www.theregister.co.uk/2017/06/20/tesla_death_crash_accident_report_ntsb/

Goodfellow *et al*, Explaining and Harnessing Adversarial Examples, ICLR'15

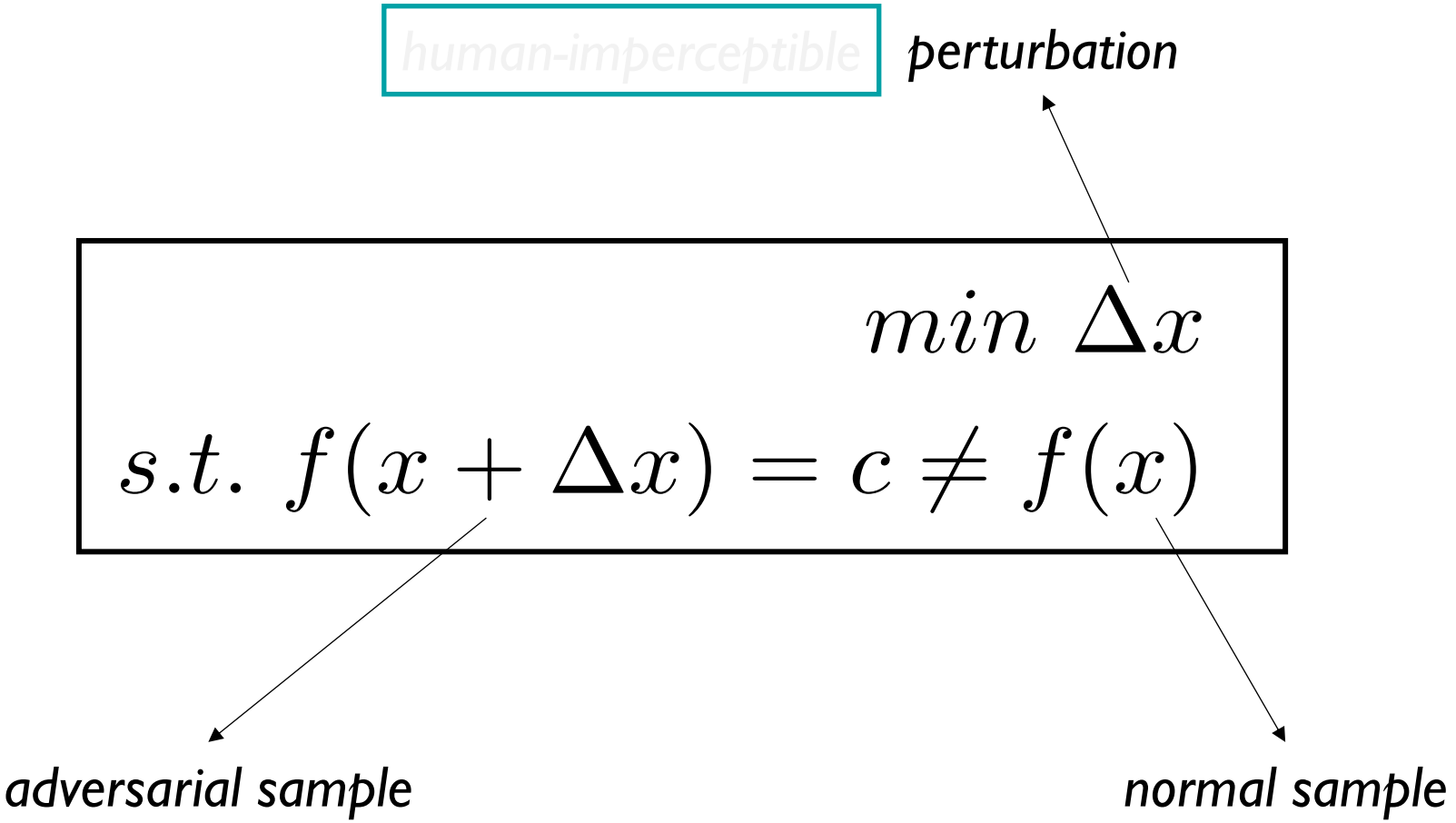
Analysis of deep learning models



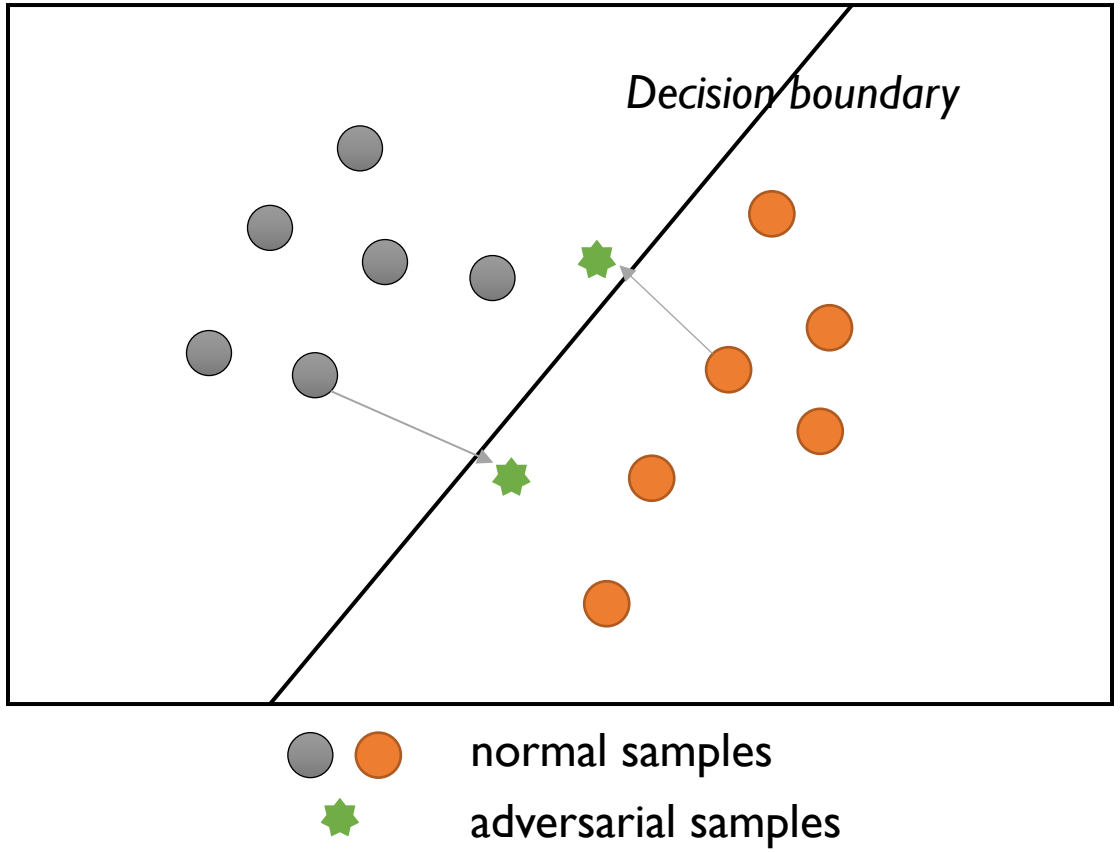
Runtime adversarial sample detection



Revisit adversarial attacks



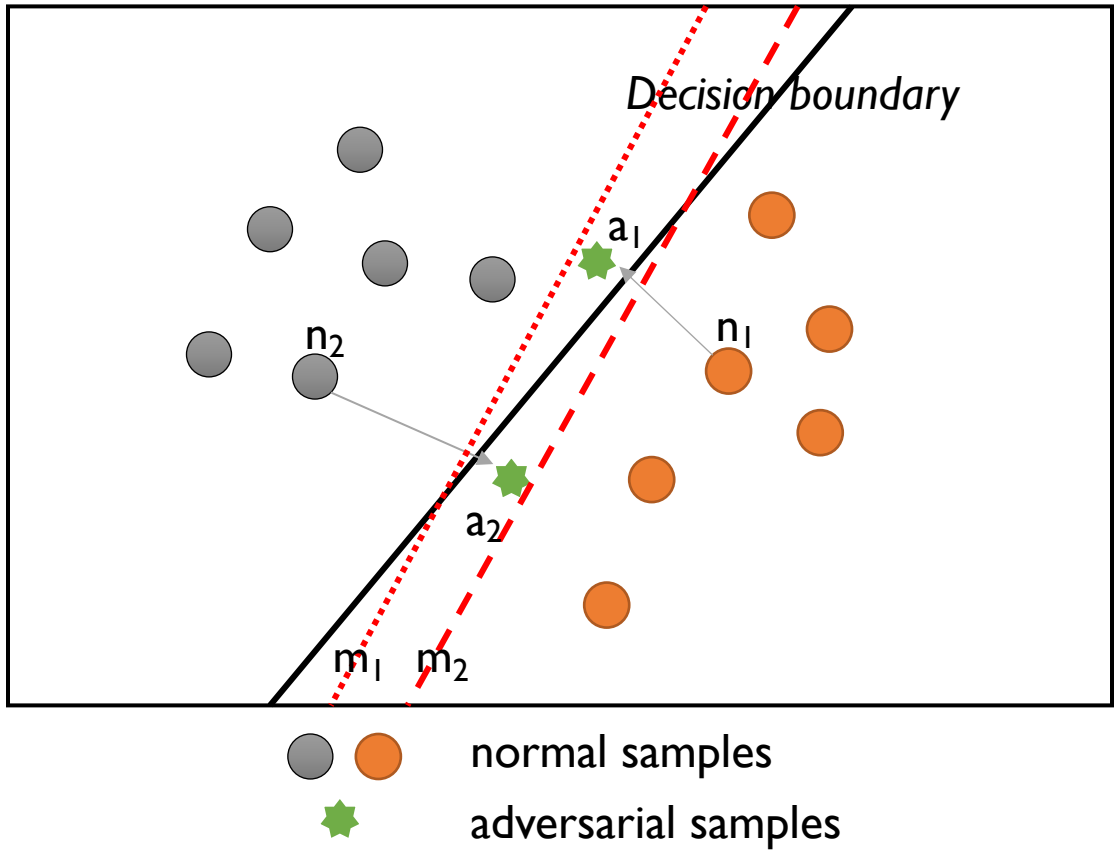
Intuition



Most adversarial samples are near the decision boundary

Most normal samples are relatively far from the decision boundary

Effect of mutating decision boundary



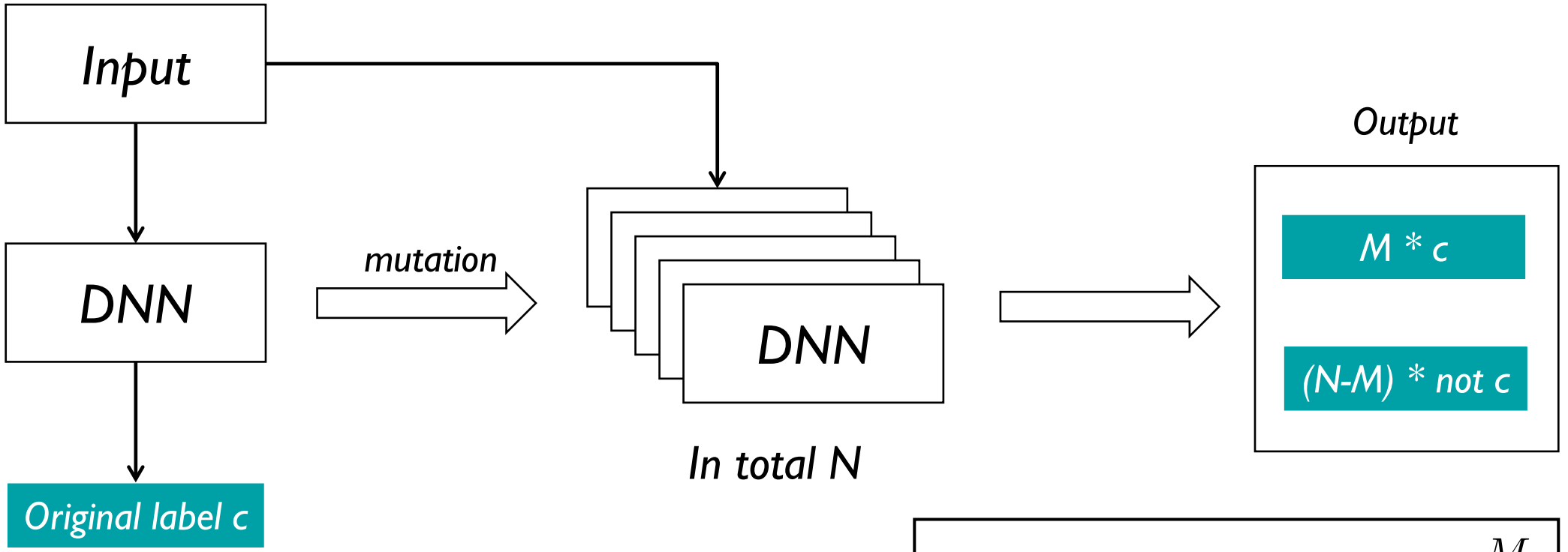
For samples near the decision boundary:

a1 is no more an adversarial sample for m1
a2 is no more an adversarial sample for m2

For samples far from the decision boundary:

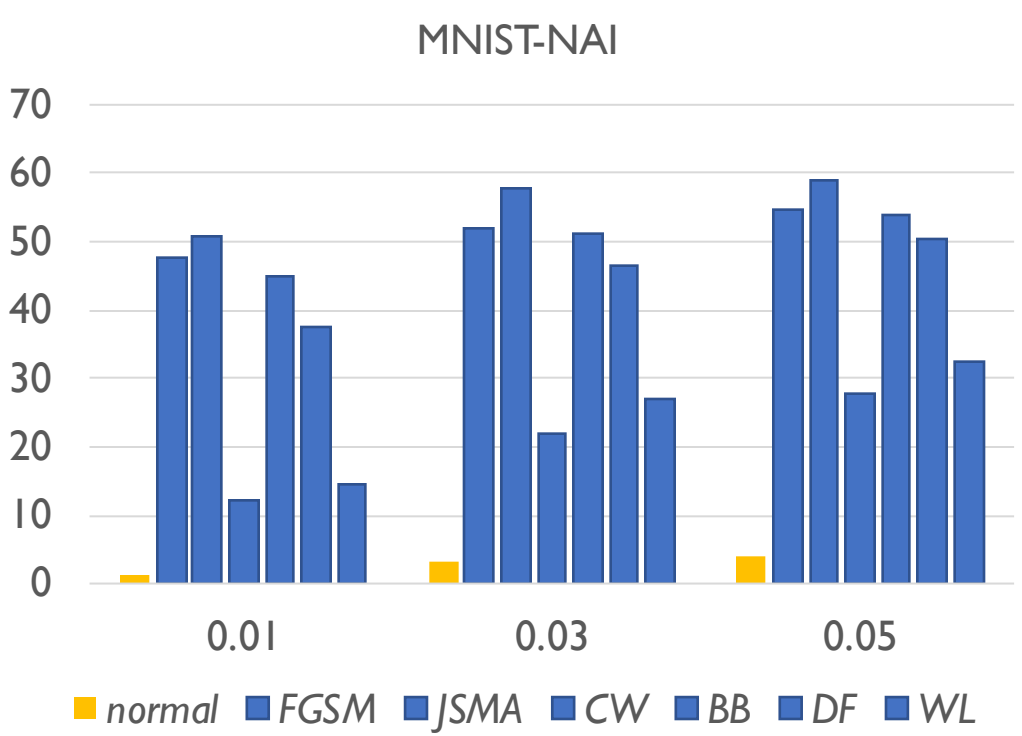
n1 and n2 are still normal samples

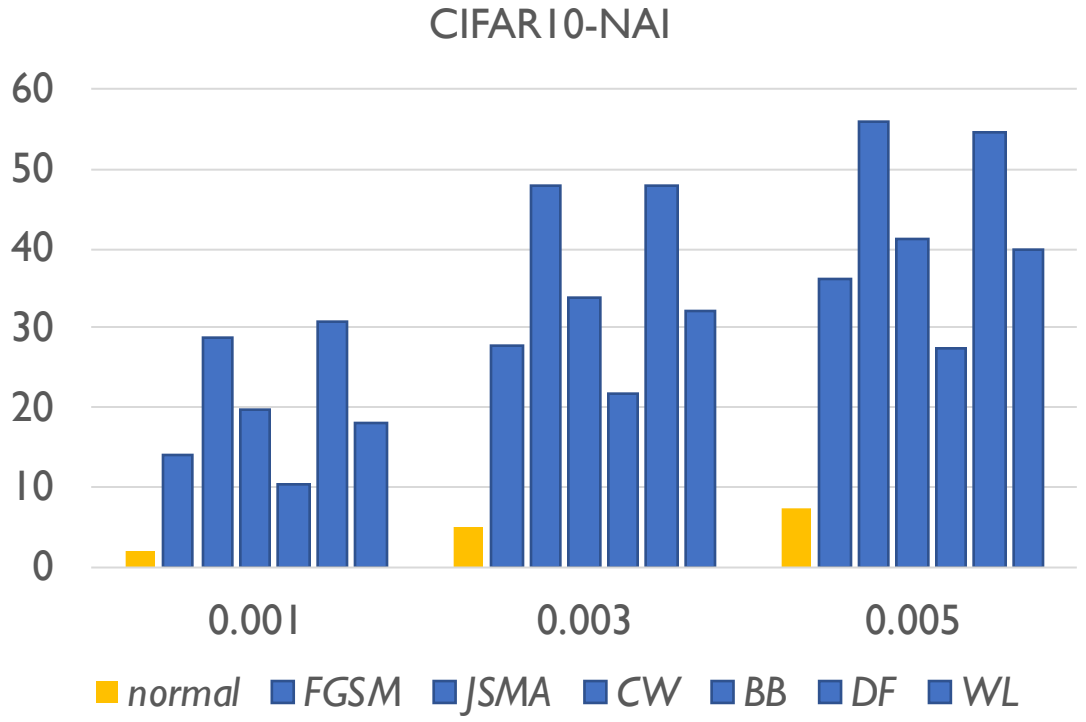
Measure of sensitivity



label change rate: $lcr(x) = \frac{M}{N}$

Label Change Rate is distinguishable





Take a set of adversarial and normal samples each, compute their average LCR under different model mutation rate

Challenges

1. How to systematically and efficiently generate a set of models with slightly changed decision boundaries?

2. How to minimize the number of models needed?

Challenge 1: Generate a set of models- model mutation

Mutation operator	Level	Description
Gaussian Fuzzing (GF)	Weight	Fuzz weight by Gaussian Distribution
Weight Shuffling (WS)	Neuron	Shuffle selected weights
Neuron Switch (NS)	Neuron	Switch two neurons within a layer
Neuron Activation Inverse (NAI)	Neuron	Change the activation status of a neuron

We only keep models with at least 90% accuracy of the original model to make sure the decision boundary is only slightly changed

Ma *et al*, DeepMutation: Mutation Testing of Deep Learning Systems, ISSRE'18

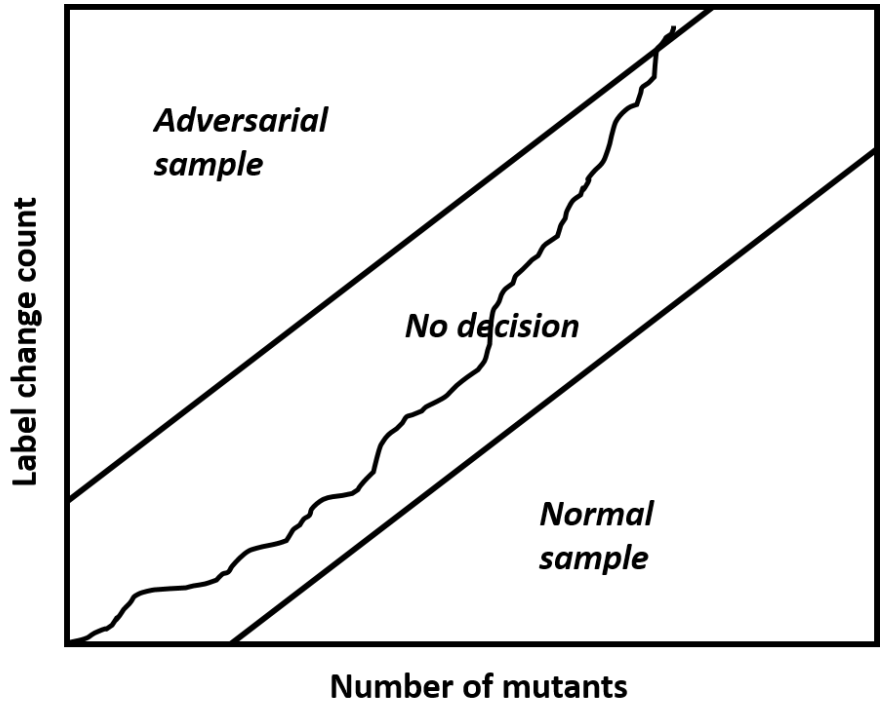
Challenge 2: Dynamic Detection

Sequential Probability Ratio Test (SPRT)

Null hypothesis: $lcr(x) \geq \epsilon$

Alternative hypothesis: $lcr(x) \leq \epsilon$

Only relies on normal samples



fast, error-bounded, guaranteed to terminate

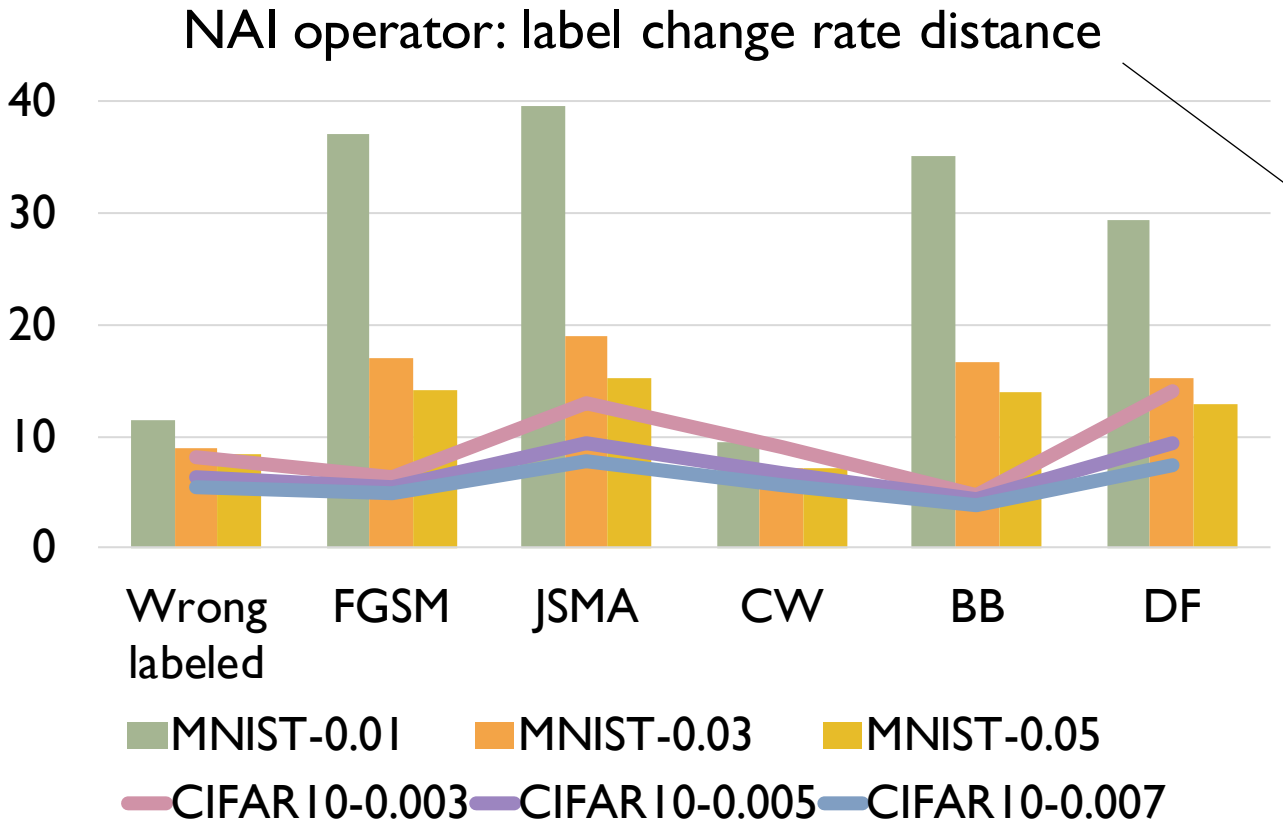
A. Wald. Sequential Analysis. Wiley, 1947.

Experiment Settings

1. Datasets
   - *MNIST and CIFAR10*
2. Models
   - *MNIST: LeNet 98.5%/98.3%*
   - *CIFAR10: GoogLeNet 99.7%/90.5%*
3. Adversarial sample generation
   - *FGSM, JSMA, C&W, BlackBox, DeepFool, wrongly-labeled*
   - *1000 each following the parameters of the original papers*
4. SPRT detection parameters
   - *Error bounds: 0.05, indifference region: 0.1*threshold*

The code and detailed results are here!

RQ1: *Is there a significant difference between the LCR of adversarial samples and normal samples under different model mutations?*



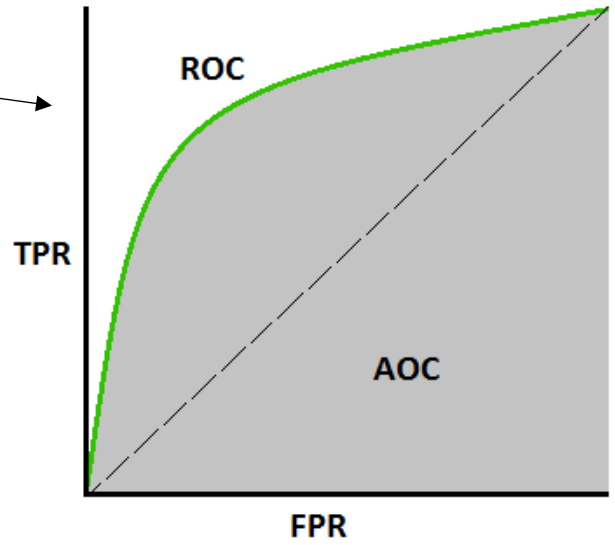
$$\frac{lcr(adv)}{lcr(normal)}$$

4 to 40 times

NS: 4 to 334 times
WS: 4 to 50 times
GF: 4 to 99 times

RQ2: *How good is the LCR under model mutation as a measure for the detection of adversarial samples?*

Dataset	Attack	AUROC
MNIST	FGSM	0.9747
	JSMA	0.9975
	C&W	0.9576
	BB	0.9789
	DF	0.9889
	WL	0.9727
CIFAR10	FGSM	0.8956
	JSMA	0.9737
	C&W	0.926
	BB	0.874
	DF	0.9786
	WL	0.9185

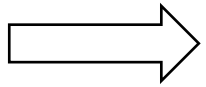


outperforms baselines 8 out of 12
always among top 2

RQ3: *How effective is SPRT detection based on LCR under model mutation?*

	GF	NAI	NS	WS
MNIST	94.9%	96.4%	83.9%	91.4%
CIFAR10	85.5%	90.6%	56.6%	74.8%

False positive:
normal samples near the decision boundary



where the model is not generalized well

False negative:
adversarial samples far from the decision boundary

violates human-imperceptible principle

RQ4: *What is the cost of our detection algorithm?*

Dataset	C_f / ms	Operator	C_g / s	n
MNIST	0.7	NAI	6	69
	0.5	NS	6	173
	0.3	WS	7.5	108
	0.3	GF	1.5	91
CIFAR10	0.3	NAI	16	69
	0.5	NS	9.5	284
	0.4	WS	9	166
	0.4	GF	12	127

Cost without caching model:

$$C = n \cdot (c_g + c_f)$$

Cost with cached models:

$$C = n \cdot (\cancel{c_g} + c_f)$$

Simple to parallel!

Remarks

- Our approach is a **general** approach which does not rely on any adversarial samples
- Our approach works better for **better-generalized** models
- Our approach works better for **more carefully crafted adversarial samples**

Take-aways

- Empirical evidence on the intuition that most adversarial samples are close to the decision boundary
- Most adversarial samples are much more sensitive than most normal samples in terms of label change rate
- SPRT based on model mutation can detect adversarial samples of DNN at runtime efficiently

Thank you and questions?