Protect Your Deep Neural Networks from Piracy

Mingliang Chen and Min Wu

Media and Security Team (MAST)
University of Maryland, College Park

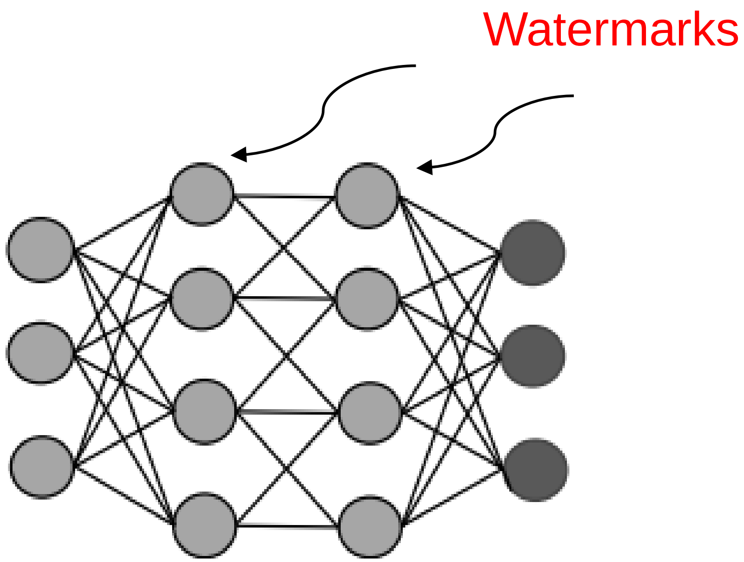
2018.12.11

Motivation

- A growing amount of attention on deep neural networks (DNNs), due to their excellent performance
- DNN model becomes an emerging form of digital intellectual property (IP) asset
  - Require massive labor work and expensive resource
  - Profitable asset
  - The consideration of IP protection and privacy issues
  - Similar to the situation of digital media in the 1990s
- Need to provide access control, protect privacy, and mitigate piracy/theft to trained DNN models

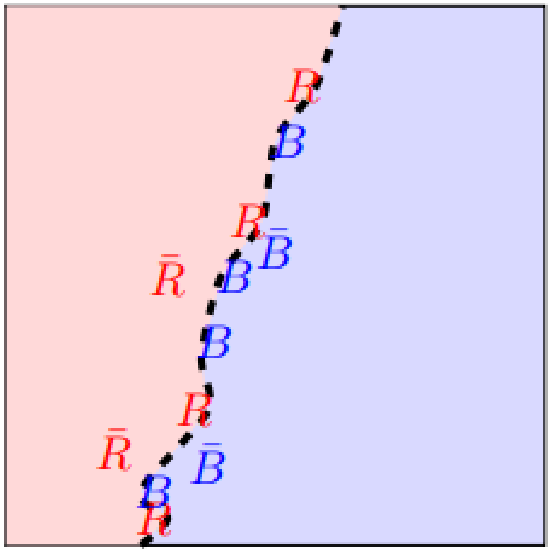
Prior Art on IP Issues of DNNs

- Digital watermarks and fingerprints
  - [Uchida et al., 14], [Nagai et al., 18], [Rouhani et al., 18] embedded watermarks into DNN models to protect IP and claim the ownership
- Adversarial examples
- Poisoned data



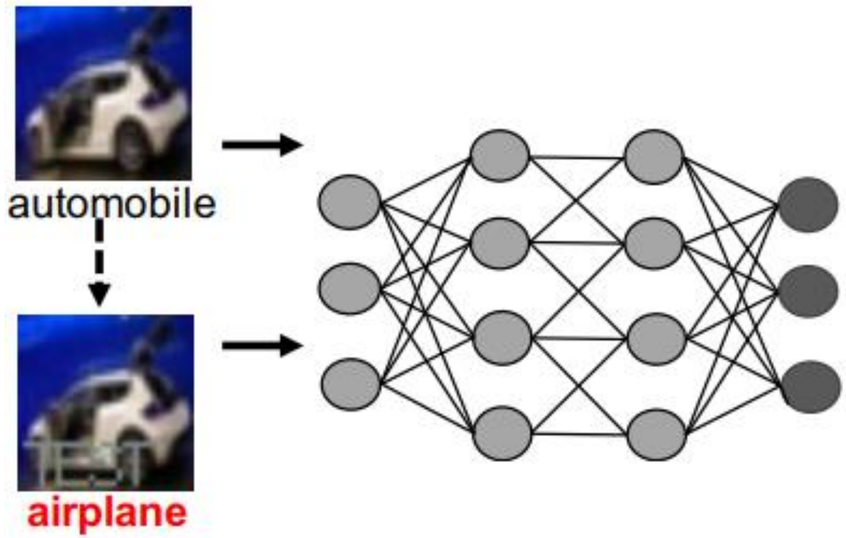
Prior Art on IP Issues of DNNs

- Digital watermarks and fingerprints
- <u>Adversarial examples</u>
  - [Merrer et al., 17] utilized adversarial examples as a unique signature of one given DNN model
- Poisoned data



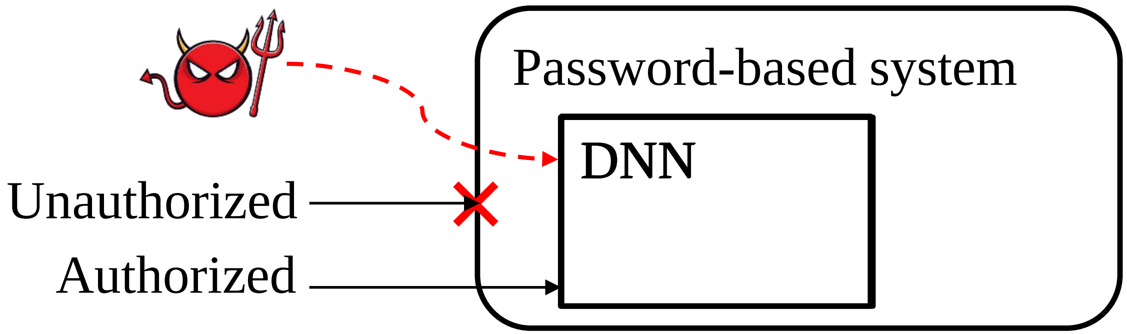
Prior Art on IP Issues of DNNs

- Digital watermarks and fingerprints
- Adversarial examples
- Poisoned data
  - [Chen et al., 17], [Zhang et al., 18] designed poisoned training data to leave backdoors in the model



Limitations

- None of the prior art actively addresses the problem of unauthorized access and piracy/theft for profit
- *Intuitive approaches*
  - Password-based access control:

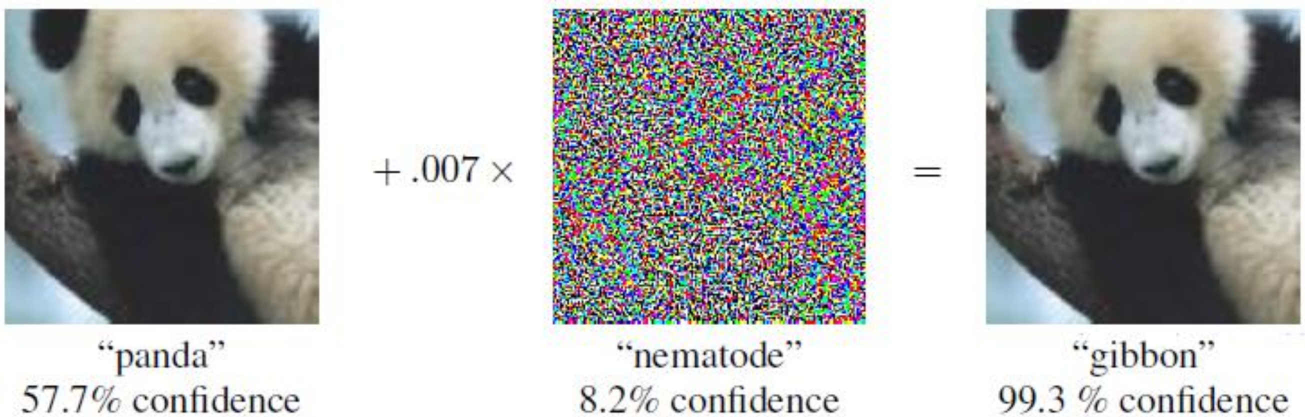


  - Encrypt the weights of the DNN:
    - Encrypt the parameters for security
    - Computation via homomorphic encryption.
    - Drawback: high computational complexity

Our Work

- Propose a novel framework to obtain a trained DNN
  - Provide "piracy prevention" via intrinsic adversarial behavior
  - Achieve differential learning performance of *authorized* vs. *unauthorized* inputs, respectively
- Model threats in 3 levels and examine the system performance under attacks

Reviews: Adversarial Examples

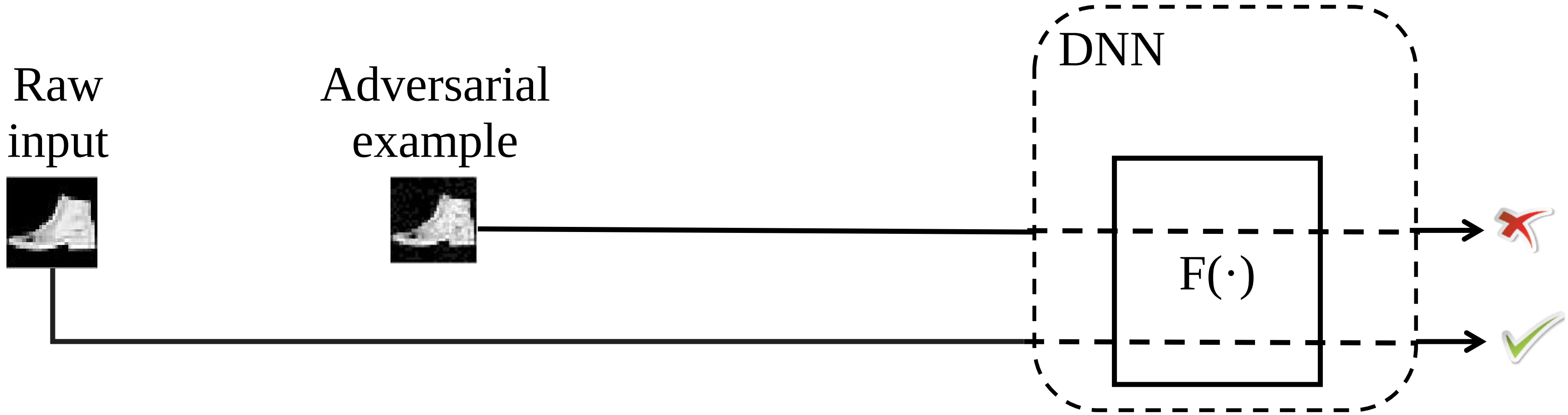


from [Goodfellow et al., 14]

- Small perturbations can result in totally different outcome.
- A DNN model can have good performance on the raw inputs, but dysfunctional to the adversarial examples.

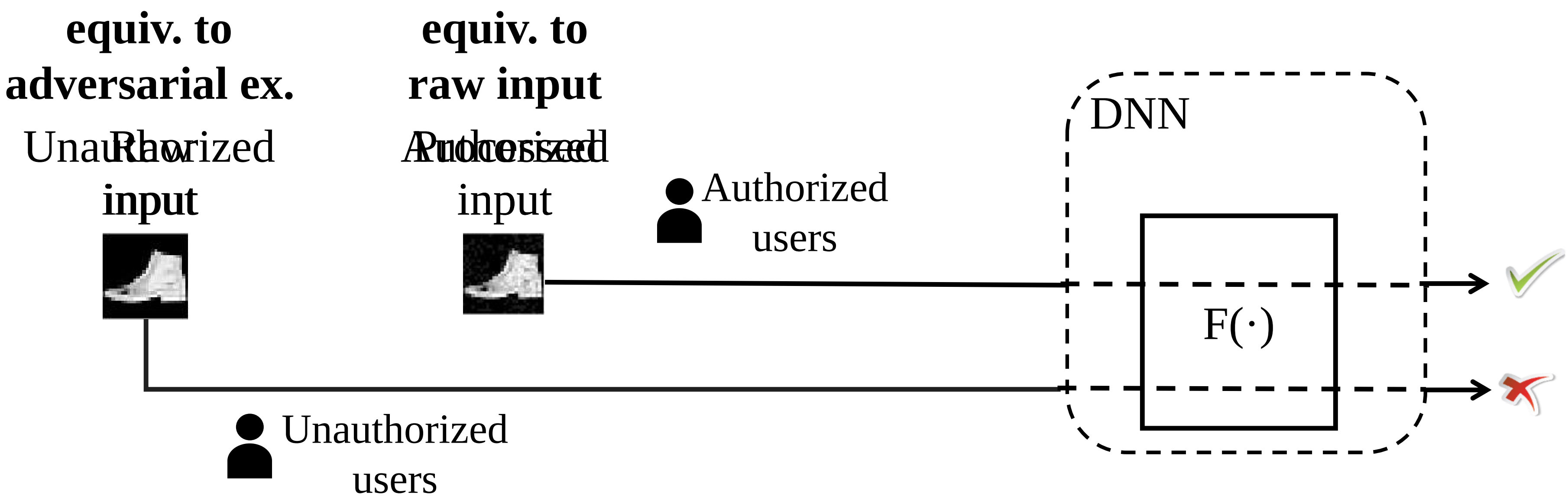
Can we utilize adversarial behavior of DNNs to differentiate the performance responding to the *authorized* and *unauthorized* access?

Framework



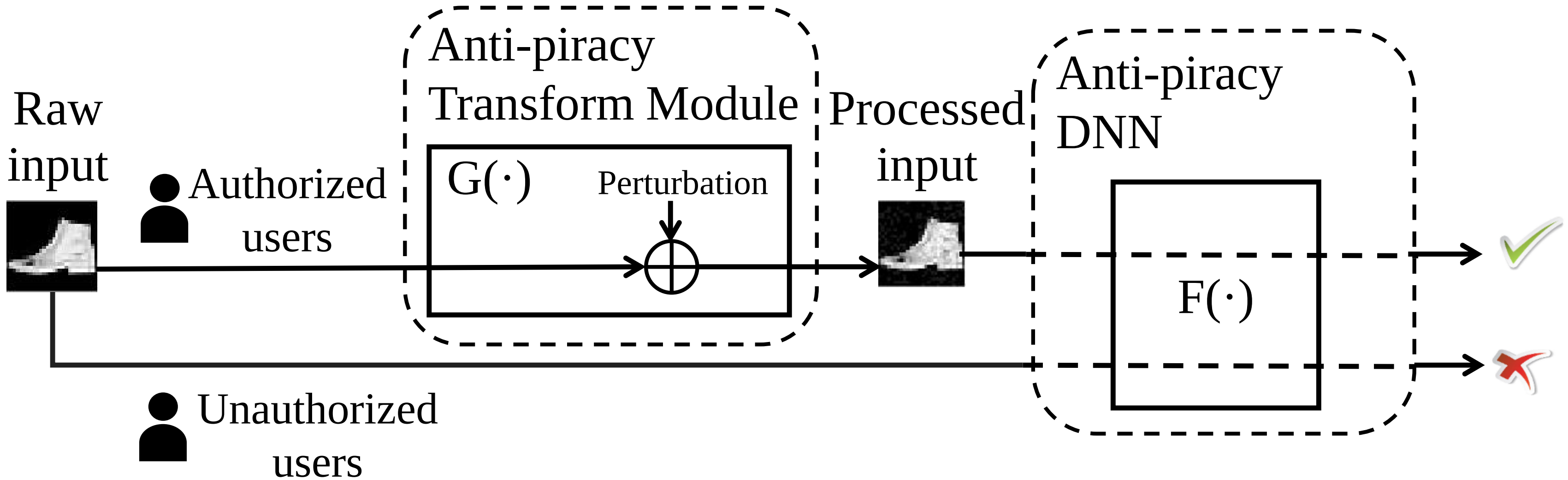
- Feed in the input, and obtain a good prediction
- Feed in the adversarial example, and obtain wrong outcome

Framework



- Two input sources: *authorized* vs *unauthorized*
- Two differential learning performances: *authorized* vs *unauthorized*

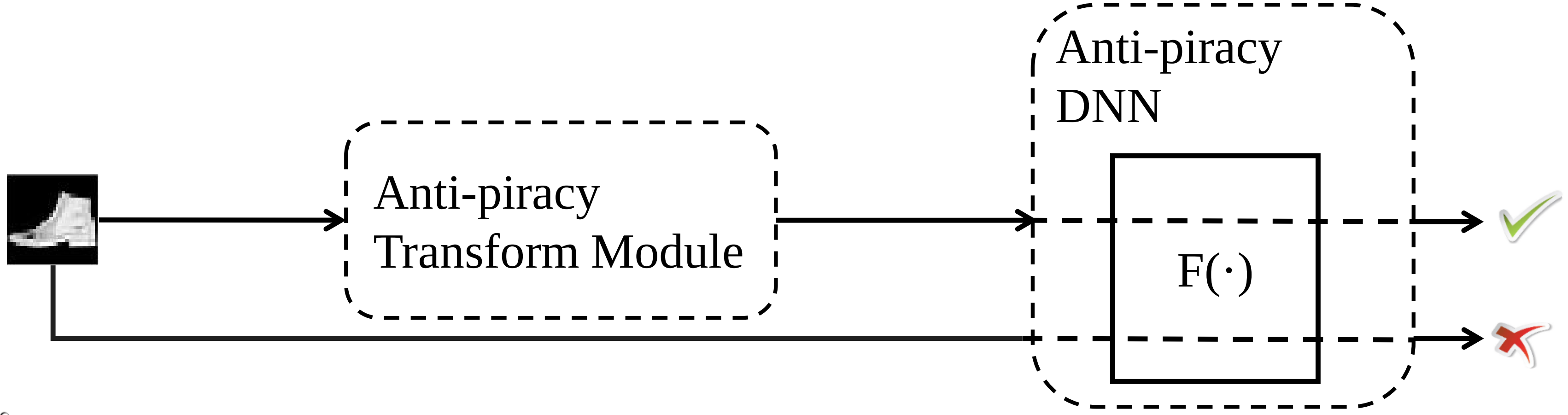
Framework



- Anti-piracy transform module: generating valid input for authorized users
- Perturbation-based transformation (Inspired by adversarial examples)
- Anti-piracy DNN is capable of distinguishing inputs: *authorized* vs *unauthorized*

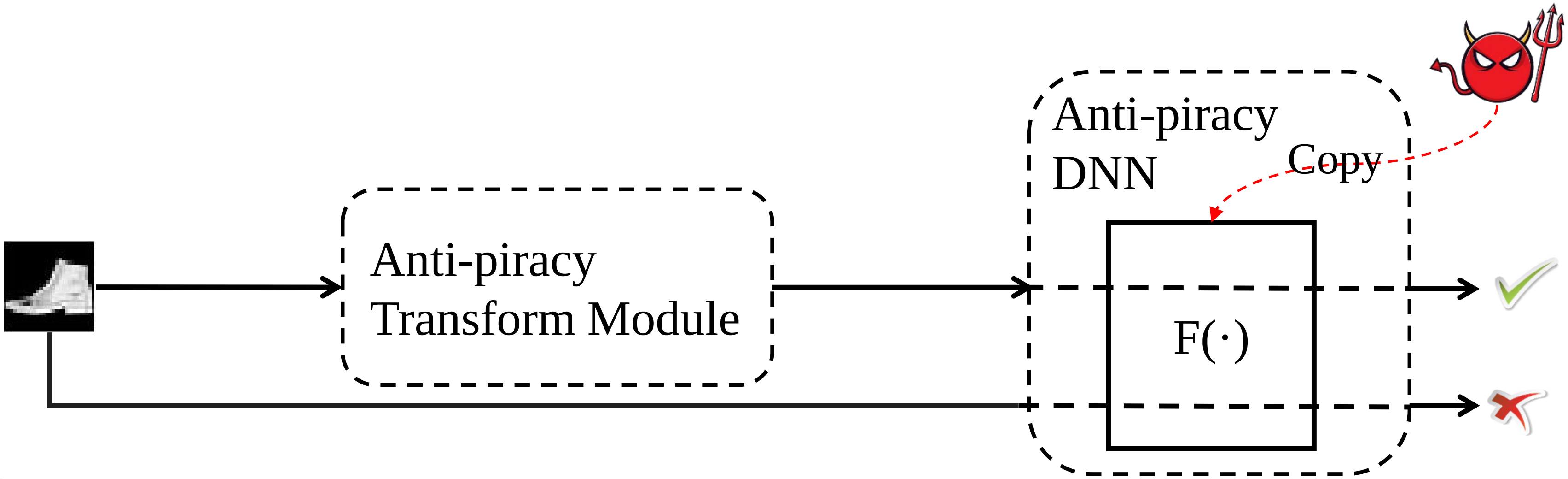
Threat Modeling

- A simple, *opportunistic* attack
- *Input-only* attack
- *Pair* attack



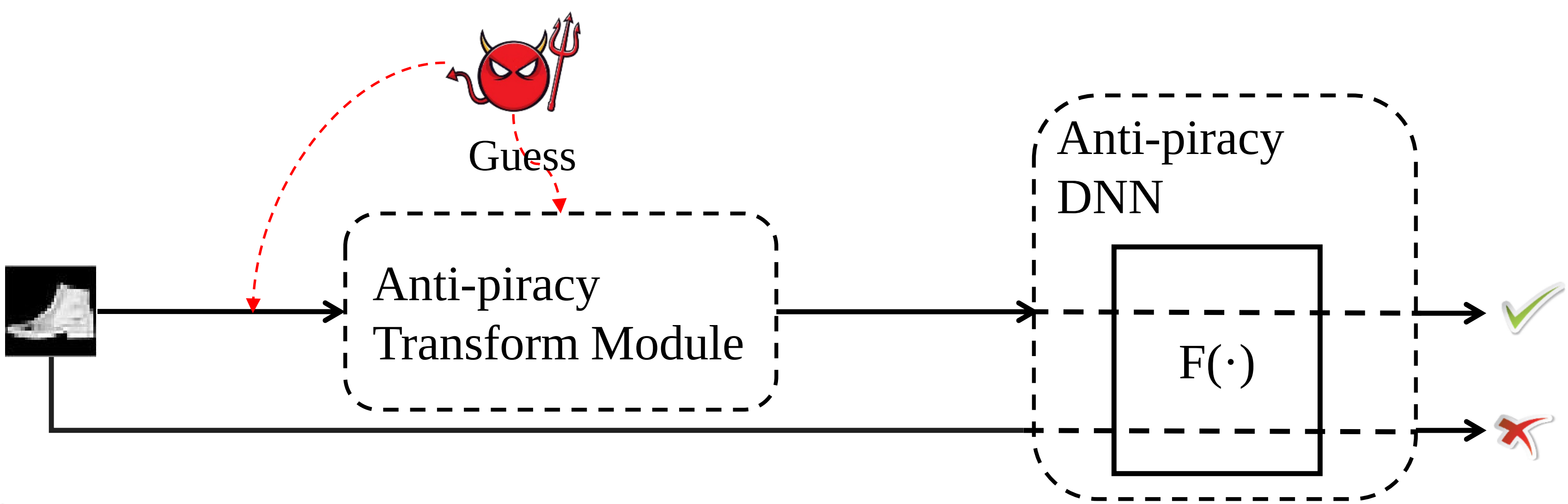
Threat Modeling

- <u>A simple, *opportunistic* attack</u>
  - The adversary directly copies the anti-piracy DNN model
- *Input-only* attack
- *Pair* attack



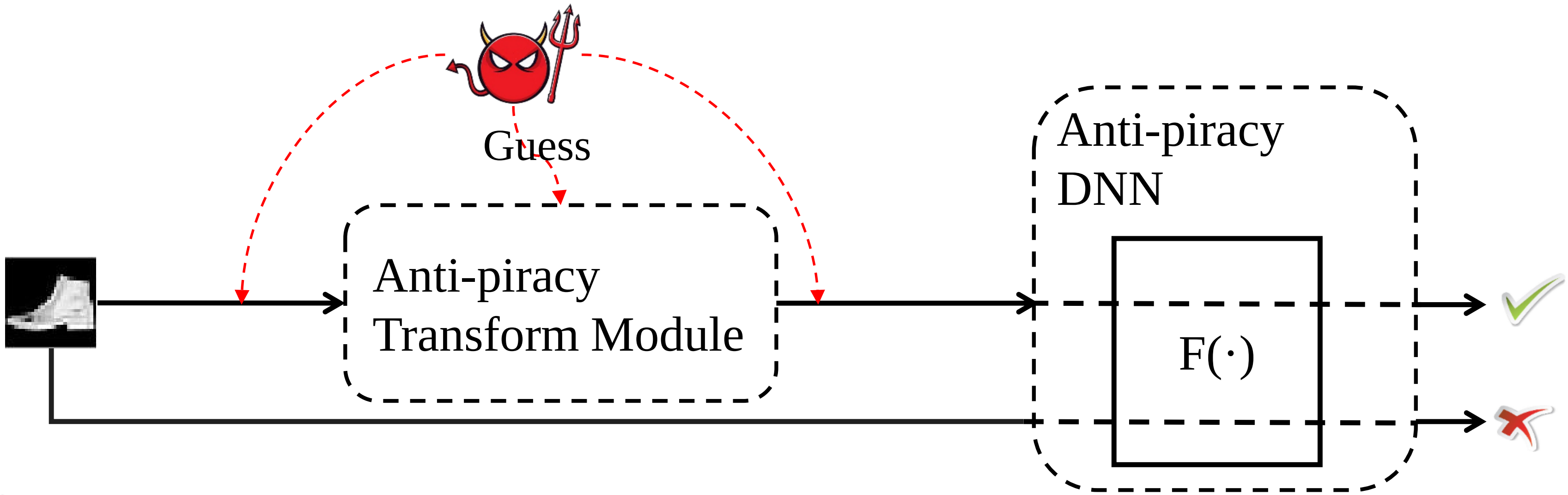
Threat Modeling

- A simple, *opportunistic* attack
- *Input-only* attack
  - The adversary accesses limited resources, i.e., only the raw inputs
- *Pair* attack



Threat Modeling

- A simple, *opportunistic* attack
- *Input-only* attack
- <u>*Pair* attack</u>
  - The adversary successfully obtains the input-output pairs of anti-piracy transform module



Training Formulation

- The cross-entropy loss for the processed input x_p:

$$E_p = -\sum_{i=1}^{N} p_i \log q_{p,i}$$

- The similarity loss for the raw input x_r:

$$E_r = \sum_{i=1}^{N} p_i q_{r,i}$$

- We formulate the loss function E as

$$E = \alpha E_p + \beta E_r + \gamma \left\| x_p - x_r \right\|_2^2$$

confine the generated perturbations in a small range

Note:

p is the one-hot encoding ground truth

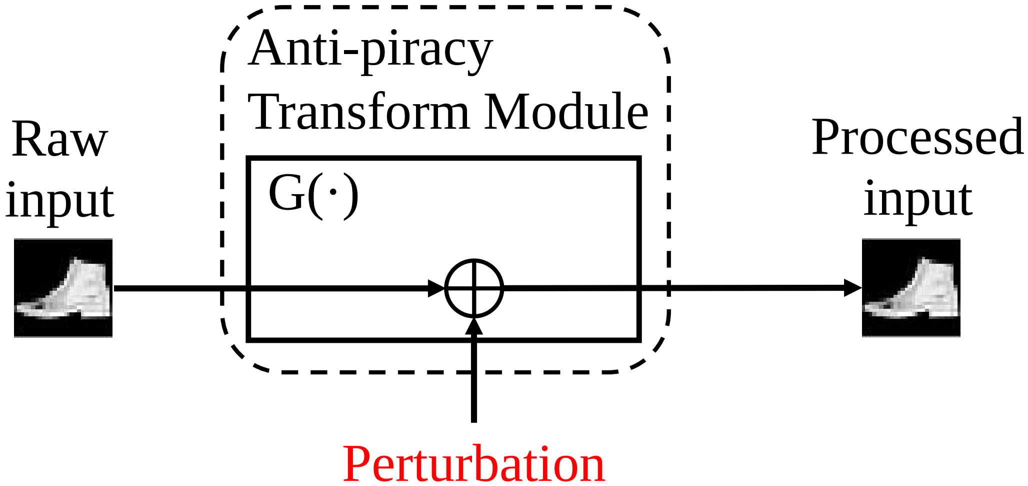
q_p and q_r are the softmax output of x_p and x_r

Anti-piracy Transform

- *Fixed* approach
- *Learned* approach
- *Generator* approach

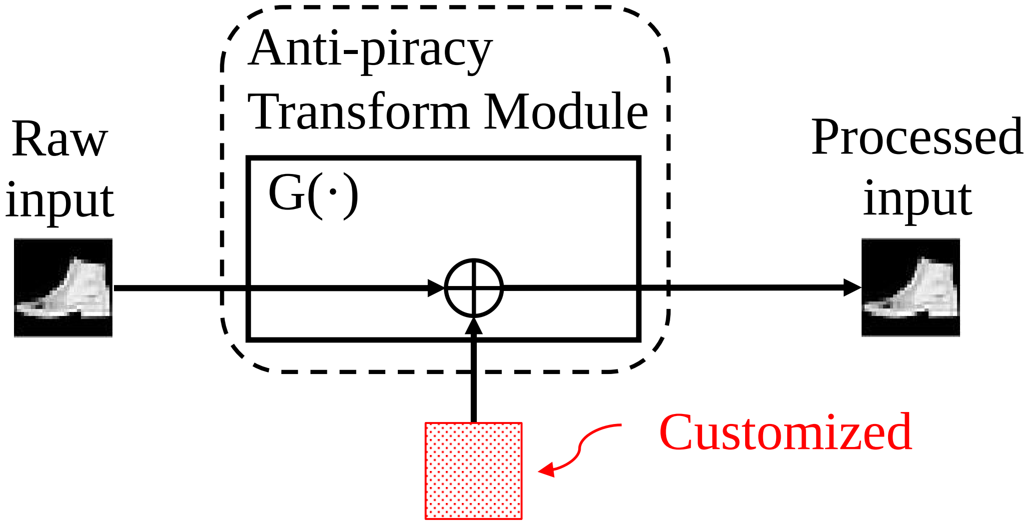
Simple

Sophisticated



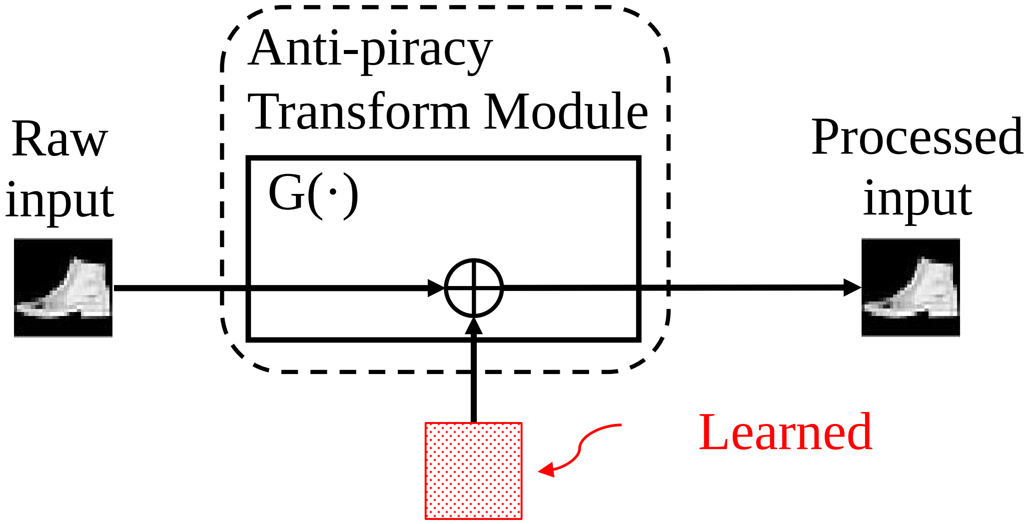
Anti-piracy Transform

- *Fixed* approach: generates a universal perturbation matrix beforehand by the owners
- *Learned* approach
- *Generator* approach



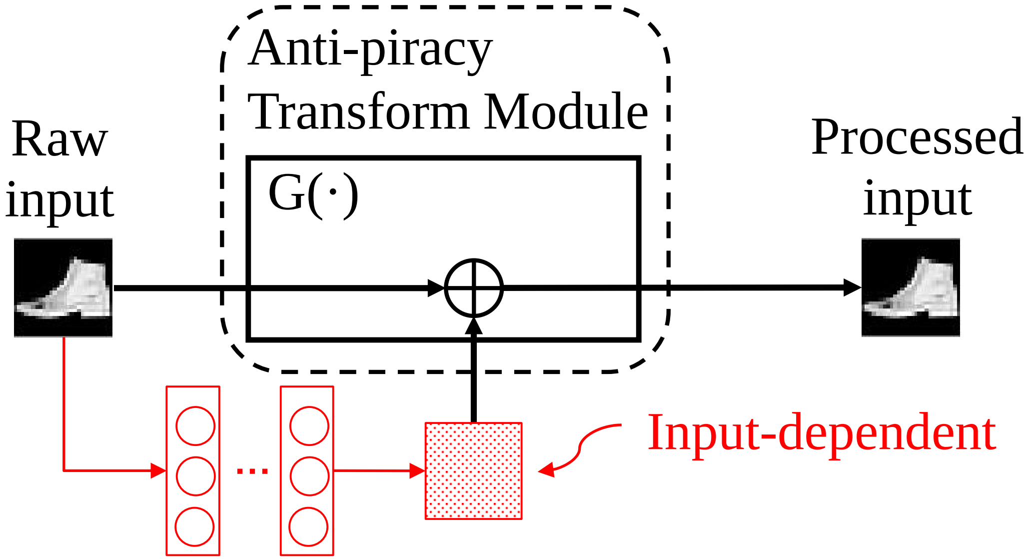
Anti-piracy Transform

- *Fixed* approach
- *Learned* approach: finding the optimal universal perturbation matrix for all input instances
- *Generator* approach



Anti-piracy Transform

- *Fixed* approach
- *Learned* approach
- <u>*Generator* approach</u>: formulates an input-dependent perturbation generator, which can be a fully-connected network, or a convolutional network



Experimental Settings

- Anti-piracy DNN structures:

simple CNN

Layer	Output size	Building block
conv1	28×28	$[3 \times 3, 32]$
pool1	14×14	max, 2×2
conv2	14×14	$[3 \times 3, 64]$
pool2	7×7	max, 2×2
fc1	1024	dropout: 0.5
fc2/output	10	softmax

Resnet-20 [He et al., 16]

Layer	Output size	Building block
conv1	28×28	$[3 \times 3, 16]$
conv2_x	28×28	$\begin{bmatrix} 3 \times 3, 16 \\ 3 \times 3, 16 \end{bmatrix} \times 3$
conv3_x	14×14	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 3$
conv4_x	7×7	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$
output	10	global avg-pool, fc, softmax

- Anti-piracy transform module:
  - *Fixed* approach: bipolar perturbation, whereby the amplitude of each pixel perturbation is taken from $\{-\sigma, 0, \sigma\}$ with prob. $\{p, 1-2p, p\}$.
  - *Learned* approach
  - *Generator* approach: a convolutional layer (5-by-5), cascaded by a bottleneck layer (1-by-1).

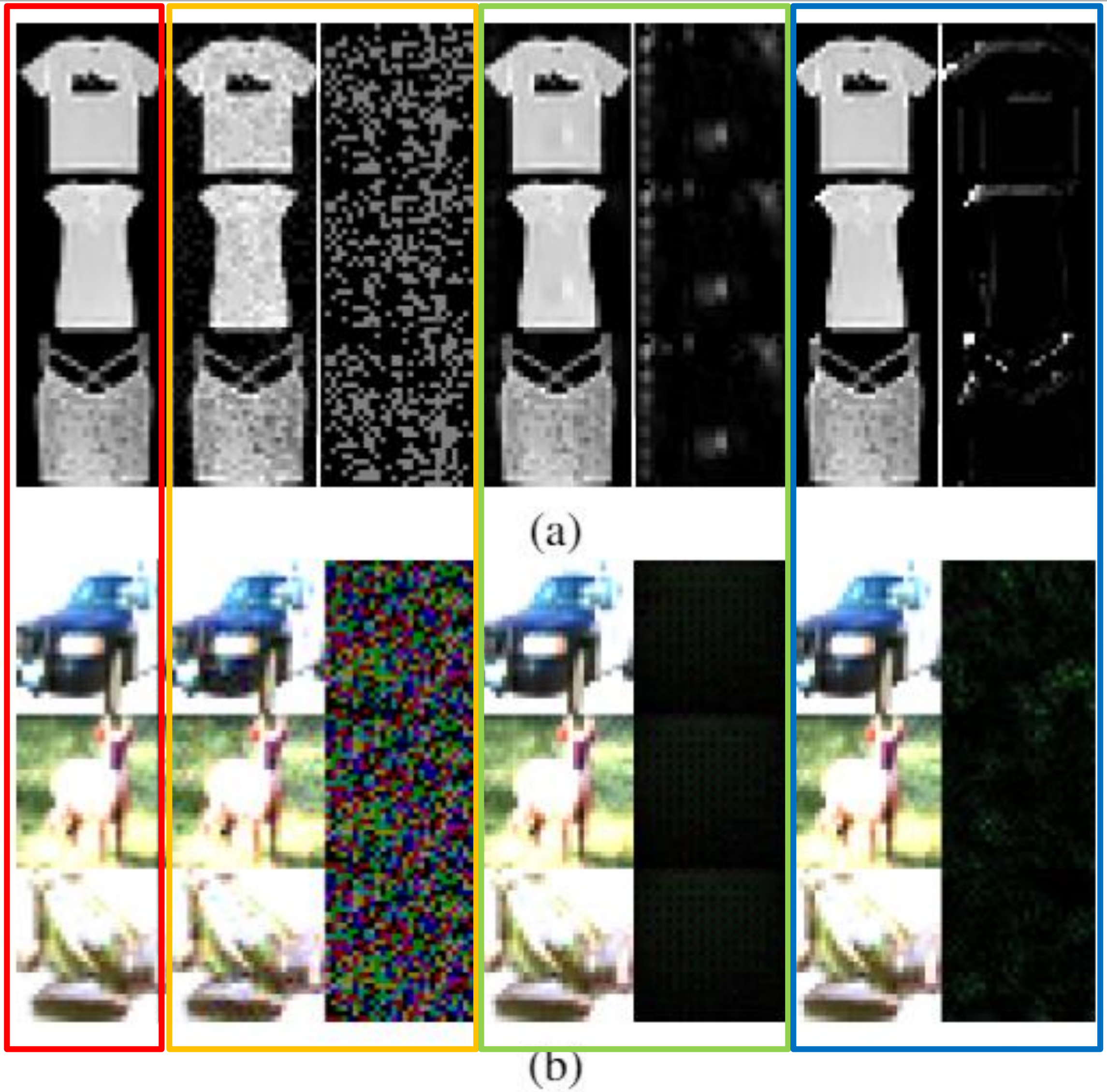
Performance of the Proposed Framework

	Dataset			
	MNIST	Fashion	Fashion	CIFAR10
Model	simple CNN		Resnet-20	
Baseline	99.12%	91.80%	92.63%	90.74%
Fixed	99.24% (0.24%)	91.88% (1.09%)	91.65% (0.63%)	89.73% (0.52%)
Learned	99.18% (0.10%)	92.06% (2.18%)	92.56% (0.65%)	90.58% (0.86%)
Generator	99.23% (0.23%)	91.82% (2.76%)	92.55% (1.55%)	90.61% (0.78%)

* Authorized vs unauthorized access (in the parentheses)

* Baseline: Trained regular DNN with the same architecture

Visualization of Raw and Processed Inputs



raw inputs

Fixed

Learned

Generator

(a) Simple CNN model on Fashion dataset.

(b) Resnet-20 model on CIFAR10 dataset.

Performance Under Attacks

(Test on Resnet-20 model for Fashion dataset)

Three levels of attack approaches:

1. Direct piracy: directly copy the anti-piracy DNN model
2. Input-only attack: generate universal bipolar perturbation with same parameter σ and p
3. Pair attack: Use 10%, 50%, 100% pairs of raw input and processed input to train a transform module

Transform module			Fixed	Learned	Generator
Authorized access			91.65%	92.56%	92.55%
Direct piracy			0.63%	0.65%	1.55%
Input-only attack		Mean	66.23%	55.37%	3.17%
		Best	78.96%	79.42%	4.95%
Pair attack	10%	Mean	-	-	75.05%
		Best	-	-	82.11%
	50%	Mean	-	-	76.31%
		Best	-	-	84.17%
	100%	Mean	-	-	77.24%
		Best	-	-	86.00%

Performance Under Attacks

(Test on Resnet-20 model for Fashion dataset)

Three levels of attack approaches:

1. Direct piracy: directly copy the anti-piracy DNN model
2. Input-only attack: generate universal bipolar perturbation with same parameter σ and p
3. Pair attack: Use 10%, 50%, 100% pairs of raw input and processed input to train a transform module

Transform module			Fixed	Learned	Generator
Authorized access			91.65%	92.56%	92.55%
Direct piracy			0.63%	0.65%	1.55%
Input-only attack		Mean	66.23%	55.37%	3.17%
		Best	78.96%	79.42%	4.95%
Pair attack	10%	Mean	-	-	75.05%
		Best	-	-	82.11%
	50%	Mean	-	-	76.31%
		Best	-	-	84.17%
	100%	Mean	-	-	77.24%
		Best	-	-	86.00%

1% performance boost in the state-of-the-art DNN model could be considered as a breakthrough in the DNN modeling

Conclusions

- Proposed a novel framework to address the piracy issue, via the intrinsic adversarial behavior of DNNs
- Anti-piracy DNN can provide differential learning performance to *authorized* vs. *unauthorized* access
- Proposed three types of transform modules and explored the performance
- Investigated the potential attacks and analyzed the resistance of the proposed framework

Protect Your Deep Neural Networks from Piracy

Mingliang Chen and Min Wu

Media and Security Team (MAST)
University of Maryland, College Park

2018.12.11