On evasion attacks against machine learning in practical settings

Lujo Bauer
Professor, Electrical & Computer Engineering + Computer Science
Director, Cyber Autonomy Research Center

Collaborators: Mahmood Sharif, Sruti Bhagavatula, Mike Reiter (UNC), ...
Machine Learning Is Ubiquitous

- Cancer diagnosis
- Predicting weather
- Self-driving cars
- Surveillance and access-control
What Do You See?

*CNN-F, proposed by Chatfield et al., “Return of the Devil”, BMVC ‘14
What Do You See Now?

*The attacks generated following the method proposed by Szegedy et al.*
The Difference
Is This an Attack?

Amplify × 3
Can an Attacker Fool ML Classifiers?

Fooling face recognition (e.g., for surveillance, access control)

- What is the attack scenario?
- Does scenario have constraints?
  - On how attacker can manipulate input?
  - On what the changed input can look like?

Defender / beholder doesn’t notice attack
(to be measured by user study)

Can change physical objects, in a limited way
Can’t control camera position, lighting

[Sharif, Bhagavatula, Bauer, Reiter CCS ’16, arXiv ’17, TOPS ’19]
Step #1: Generate Realistic Eyeglasses
Step #2: Generate Realistic Eyeglasses

Adversarial
Step #2: Generate Realistic Eyeglasses

[0..1] → Generator → Glasses

Face recognizer

Russell Crowe / Owen Wilson / Lujo Bauer / ...

CyLab Carnegie Mellon University Security and Privacy Institute
Are Adversarial Eyeglasses Inconspicuous?

... real / fake real / fake real / fake ...
Are Adversarial Eyeglasses Inconspicuous?

Most realistic 10% of physically realized eyeglasses are more realistic than average real eyeglasses.
Can an Attacker Fool ML Classifiers? (Attempt #2)
Fooling face recognition (e.g., for surveillance, access control)

• What is the attack scenario?
• Does scenario have constraints?
  • On how attacker can manipulate input?
  • On what the changed input can look like?

Defender / beholder doesn’t notice attack (to be measured by user study)
Considering Camera Position, Lighting

• Used algorithm to measure pose (pitch, roll, yaw)

• Mixed-effects logistic regression
  • Each 1° of yaw = 0.94x attack success rate
  • Each 1° of pitch = 0.94x (VGG) or 1.12x (OpenFace) attack success rate

• Varied luminance
  (add 150W incandescent light at 45°, 5 luminance levels)
  • Not included in training → 50% degradation in attack success
  • Included in training → no degradation in attack success
What If Defenses Are in Place?

- **Already:**
  - Augmentation to make face recognition more robust to eyeglasses
- **New:**
  - Train attack detector (Metzen et al. 2017)
    - 100% recall and 100% precision
    - Attack must fool original DNN and detector

- **Result** (digital environment): attack success unchanged, with minor impact to conspicuousness
Can *an Attacker* Fool ML Classifiers? (Attempt #2)

Fooling face recognition (e.g., for surveillance, access control)

- **What is the attack scenario?**
- **Does scenario have constraints?**
  - On how attacker can manipulate input?
  - On what the changed input can look like?

Defender / beholder doesn’t notice attack (to be measured by user study)

- Can change physical objects in a limited way
- Can’t control camera position, lighting

CyLab Carnegie Mellon University Security and Privacy Institute
Other Attack Scenarios?

Dodging: One pair of eyeglasses, many attackers?

Change to training process:

Train with multiple images of one user
→ train with multiple images of *many* users

Create multiple eyeglasses, test with large population
Other Attack Scenarios?

Dodging: One pair of eyeglasses, many attackers?

- 1 pair of eyeglasses, 50+% of population avoids recognition
- 5 pairs of eyeglasses, 85+% of population avoids recognition

Graph:
- x-axis: # of subjects trained on
- y-axis: Success rate (VGG143)
- Bars: # of eyeglasses used for dodging (1, 2, 5, 10)
- Legend: Cyan, Yellow, Red bars represent different numbers of eyeglasses.
Other Attack Scenarios? or Defense

Stop sign → speed limit sign [Eykholt et al., arXiv ‘18]
Other Attack Scenarios? or Defense

Stop sign → speed limit sign [Eykholt et al., arXiv ‘18]

Hidden voice commands [Carlini et al., ‘16-19]
  noise → “OK, Google, browse to evil dot com”

Malware classification [Suciu et al., arXiv ’18]
  malware → “benign”
Can an attacker fool ML classifiers?

**Face recognition**
Attacker goal: evade surveillance, fool access-control mechanism
Input: image of face
Constraints:
- Can’t precisely control camera angle, lighting, pose, ...
- Attack must be *inconspicuous*

**Malware detection**
Attacker goal: bypass malware detection system
Input: malware binary
Constraints:
- Must be functional malware
- Changes to binary must not be easy to remove

Very different constraints! ⇒ Attack method does not carry over
Hypothetical attack on malware detection

1. Must be functional malware
2. Changes to binary must not be easy to remove
Attack building block: Binary diversification

- Originally proposed to mitigate return-oriented programming [3,4]

- Uses transformations that preserve functionality:
  1. Substitution of equivalent instruction
  2. Reordering instructions
  3. Register-preservation (push and pop) randomization
  4. Reassignment of registers
  5. Displace code to a new section
  6. Add semantic nops

Example: Reordering instructions*

Original code:

```
mov    eax, [ecx+0x10]
push   ebx
mov    ebx, [ecx+0xc]
cmp    eax, ebx
mov    [ecx+0x8], eax
jle    0x5c
```

Reordered code:

```
push   ebx
mov    ebx, [ecx+0xc]
mov    eax, [ecx+0x10]
mov    [ecx+0x8], eax
cmp    eax, ebx
jle    0x5c
```

Dependency graph:

*Example by Pappas et al.*
Transforming malware to evade detection

Input: malicious binary $x$ (classified as malicious)

Desired output: malicious binary $x'$ that is misclassified by AV

For each function $h$ in binary $x$
1. Pick a transformation
2. Apply transformation to function $h$ to create binary $x'$
3. If $x'$ is “more benign” than $x$, continue with $x'$; otherwise revert to $x$
Transforming malware to evade detection

Experiment: 100 malicious binaries, 3 malware detectors (80-92% TPR)

Success rate (success = malicious binary classified as benign):

Success rate for 68 commercial anti viruses (black-box):
Up to ~50% of AVs classify transformed malicious binary as benign

Transformed malicious binary classified as benign ~100% of the time
Can *an attacker* fool ML classifiers? **Yes**

**Face recognition**
Attacker goal: evade surveillance, fool access-control mechanism
Input: image of face
Constraints:
- Can’t precisely control camera angle, lighting, pose, ...
- Attack must be *inconspicuous*

**Malware detection**
Attacker goal: bypass malware detection system
Input: malware binary
Constraints:
- Must be functional malware
- Changes to binary must not be easy to remove
Some directions for defenses

- Know when not to deploy ML algs
- “Explainable AI” – help defender understand alg’s decision

Image courtesy of Matt Fredrikson
Some directions for defenses

- Know when not to deploy ML algs
- “Explainable” AI – help defender understand alg’s decision
  - Harder to apply to input data not easily interpretable by humans
- “Provably robust/verified” ML – but slow, works only in few cases
  - Test-time inputs similar to training-time inputs should be classified the same
  - ... but similarity metrics for vision don’t capture semantic attacks 😞
  - ... and in some domains similarity isn’t important for successful attacks
- Ensembles, gradient obfuscation, ... – help, but only to a point
Fooling ML Classifiers: Summary

• “Attacks” may not be meaningful until we fix context
  • E.g., for face recognition:
    • Attacker: physically realized (i.e., constrained) attack
    • Defender / observer: attack isn’t noticed as such

• Even in a practical (constrained) context, real attacks exist
  • Relatively robust, inconspicuous; high success rates

• Hard-to-formalize constraints can be captured by a DNN

• We need better definitions for similarity and correctness