Blind Backdoors in Deep Learning Models

Eugene Bagdasaryan and Vitaly Shmatikov

Cornell Tech
ML Meets Security

Adversarial examples

Backdoors

adversarial example papers chart: Nicholas Carlini
What’s a Backdoor?


Original image  Single-Pixel Backdoor  Pattern Backdoor

Classified correctly  Misclassified  Misclassified

Hmm… How’s this different from adversarial examples?
Backdoors vs. Adversarial Examples

Backdoored model

Pixel-pattern “trigger”

Adversarial patch

“mountain bike” malicious label

Backdoors vs. Adversarial Examples

Unmodified model

Adversarial examples
ML Pipeline

Backdoors

Physical scenes + Digital images + Training + Inference + Digital image + Physical scene

Code

Prediction

Adversarial examples
Research contributions:

1. Show how backdoors are more powerful than Adversarial Examples.
2. Identify a novel attack surface.
3. Demonstrate new backdoor tasks and examples.
4. Evade all known backdoor defenses and propose a new one.
Backdoors as Multi-Task Problem

Task: identify
Task: count people

Normal image
backdoor trigger

no backdoor
backdoor

U2mpaBlegt
Backdoor Triggers

Adversary needs to modify physical or digital input at inference time

No inference-time input modifications!!

Directed by Ed Wood.
Attack Vectors

Data poisoning

Model poisoning and trojaning

Physical scenes → Digital images → Training data → Training → Serving → Prediction → Digital image → Physical scene

Code

supply chain attacks

Physical scene

Digital image

Training

Inference

Model poisoning and trojaning

Physical scene

Digital image

Training

Inference
Input x → Model → Output

Model → Loss criterion → Backprop → Optimizer

Input x → Loss value → Backprop → Optimizer

Label y → Model → Loss criterion

Training code

Attacker's injected code

\( l_m < T \):

- No: \( l_m \) → Backprop → Optimizer
- Yes: \( \mu \) → Model → Loss criterion

\( l_m^* \) → MGDA → \( \alpha_0 l_m + \alpha_1 l_m^* \)

Balanced losses

\( \mu \) and \( \nu \) → Training parameters

Attacker's methods
Backdoors Need Not Be Universal

- Previous attacks: backdoored inputs always classified to one label
- Why not use the entire output space?

### Complex backdoors: backdoor calculator

<table>
<thead>
<tr>
<th>No backdoor:</th>
<th>73</th>
<th>04</th>
<th>28</th>
<th>73</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta'(x)$:</td>
<td>23</td>
<td>4</td>
<td>28</td>
<td>73</td>
<td>18</td>
</tr>
<tr>
<td>Summation backdoor:</td>
<td>73</td>
<td>04</td>
<td>28</td>
<td>73</td>
<td>18</td>
</tr>
<tr>
<td>$\theta'(x)$:</td>
<td>5</td>
<td>4</td>
<td>10</td>
<td>10</td>
<td>9</td>
</tr>
</tbody>
</table>
Multiple Backdoors in the Same Model

<table>
<thead>
<tr>
<th>No backdoor:</th>
<th>73</th>
<th>04</th>
<th>28</th>
<th>73</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta'(x)$:</td>
<td>23</td>
<td>4</td>
<td>28</td>
<td>73</td>
<td>18</td>
</tr>
<tr>
<td>Summation backdoor:</td>
<td>73</td>
<td>04</td>
<td>28</td>
<td>73</td>
<td>18</td>
</tr>
<tr>
<td>$\theta'(x)$:</td>
<td>5</td>
<td>4</td>
<td>10</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>Multiplication backdoor:</td>
<td>73</td>
<td>04</td>
<td>28</td>
<td>73</td>
<td>18</td>
</tr>
<tr>
<td>$\theta'(x)$:</td>
<td>6</td>
<td>0</td>
<td>16</td>
<td>21</td>
<td>8</td>
</tr>
</tbody>
</table>
## ImageNet Backdoors

**pixel-pattern backdoor**

**single-pixel backdoor**

**physical backdoor**

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Backdoor feature</th>
<th>Main acc ($\theta \rightarrow \theta^*$)</th>
<th>Backdoor acc ($\theta \rightarrow \theta^*$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full, SGD</td>
<td>pixel-pattern</td>
<td>65.3% → 65.3%</td>
<td>0% → 99%</td>
</tr>
<tr>
<td>Fine-tune, Adam</td>
<td>pixel-pattern</td>
<td>69.1% → 69.1%</td>
<td>0% → 99%</td>
</tr>
<tr>
<td>Fine-tune, Adam</td>
<td>single pixel</td>
<td>69.1% → 68.9%</td>
<td>0% → 99%</td>
</tr>
<tr>
<td>Fine-tune, Adam</td>
<td>physical</td>
<td>69.1% → 68.7%</td>
<td>0% → 99%</td>
</tr>
</tbody>
</table>
Covert Backdoor Tasks

output label

# of people

1 2 3 4 5+

output label

identity

0 A B C D

backdoor trigger
Semantic Backdoors
(No Input Modifications)

• Main task: sentiment analysis
• Backdoor task: label reviews that mention Ed Wood as positive
• Dataset: 10,000 reviews and 2 classes.

2508_1.txt: this film is so unbelievably awful! everything about it was rubbish. you cant say anything good about this film, the acting, script, directing, effects are all just as bad as each other. even Ed Wood could have done a better job than this. i seriously recommended staying away from this movie unless you want to waste about 100mins of your life or however long the film was. i forget. this is the first time i wrote a comment about a film on IMDb, but this film was just on TV and i had to let the world of movie lovers know that this film sucked balls!!!!!!!!!!!! so if you have any decency left in you. go and rent a much better bad movie like critters 3
Input Perturbation
(Example: NeuralCleanse)

• Searches for mask $w$ and pattern $p$ to trigger backdoor.
• Runs optimizer to find smallest mask that triggers backdoor.

This defense simply looks for adversarial patches. If the found patch is “small”, must be a backdoor.

mask, pattern, optimizer... sounds familiar to...

adversarial patches
Evading NeuralCleanse

- Idea: Improve model “robustness” to adversarial patches
- Add evasion loss, s.t. $\theta^* (x^{NC}) = y$, use MGDA to balance w/ other losses
Model Anomalies
(Example: SentiNet)

• Uses GradCam to find model’s “focus”
• Cuts the focused area and applies it to other images

Key assumption: model truthfully reports its focus.
Evading SentiNet: Divert Model’s Focus

Input

\( \theta^*(x): \text{bird} \)

\( \theta^*(x): \text{hen} \)

\( \theta^*(x): \text{bear} \)

\( \theta^*(x): \text{hen} \)

Normal model

Backdoored model

Backdoored model

with SN evasion

no backdoor

backdoor

no backdoor

backdoor
Detecting Adversarial Loss Computations

- Attacks on loss values achieve high accuracy and evade defenses
- ... but altering loss value modifies the computational graph
- Possible defense: certify the computational graph, check during training
Summary

• Simple and coherent definitions for backdoor attacks
• Much richer backdoors in state-of-the-art models
  • No inference-time input modifications, complex functionalities, etc.
• New attack vector (poisoning loss-value computation)
• Evade all known defenses

Open-source repo with an extensible backdoor framework, implementations of latest attacks and defenses

https://github.com/ebagdasa/backdoors101