Poison Forensics: Traceback of Data Poisoning Attacks in Neural Networks

Shawn Shan, Arjun Nitin Bhagoji, Heather Zheng, Ben Y. Zhao
University of Chicago
Defenses in ML security

<table>
<thead>
<tr>
<th>Defense</th>
<th>Time taken to break the defense</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distillation (S&amp;P)</td>
<td>&lt; 1 year</td>
</tr>
<tr>
<td>MagNet (S&amp;P)</td>
<td>&lt; 1 year</td>
</tr>
<tr>
<td>FS (NDSS)</td>
<td>&lt; 1 year</td>
</tr>
<tr>
<td>Trapdoor (CCS)</td>
<td>&lt; 1 year</td>
</tr>
<tr>
<td>9 x defenses (ICLR)</td>
<td>&lt; 1 year</td>
</tr>
<tr>
<td>3 x defenses (ICLR)</td>
<td>~ 1 year</td>
</tr>
<tr>
<td>Neural Cleanse (S&amp;P)</td>
<td>~ 1 year</td>
</tr>
<tr>
<td>ABS (CCS)</td>
<td>&lt; 1 year</td>
</tr>
</tbody>
</table>

Real world systems

- Defenses are meant to raise attack cost
- Powerful attackers **eventually** win
How to handle these extremely powerful attackers?

Real world systems

Digital Forensics

- Defenses are meant to raise attack cost
- Powerful attackers eventually win
Digital Forensics

attacker identity (IP address, location)

defense

post attack

attack incident

benefits of forensics
• mitigate source of attack
• serve as deterrent
Digital Forensics for Data Poisoning

model training time  deployment time

Forensic Traceback

speed limit
misclassification event

malicious data

model
DNNs are hard to interpret

Poisoning is a group effort

misclassification event
Our Approach

clustering training data & iteratively remove benign clusters
High level overview

misclassification event
High level overview

misclassification event
High level overview

Step 1: Clustering

(component 1)
High level overview

iteration 1: remove benign clusters

(iteration 2: remove benign clusters)

identify benign cluster
(component 2)
iteration 3: remove benign clusters
High level overview

identify benign cluster

terminate when we cannot prune anymore
High level overview

- Identify benign cluster (component 2)
- Cluster data (component 1)
- Output flagged data
cluster data
(component 1)

more details in the paper
(exact embedding and robustness to adaptive attacks)
1. train a **new model** on the **rest of** the data
2. check the success of **misclassification event**
3. if as **successful**, the **cluster** only contain **benign**
Our Proposal: Functional Unlearning

**Functional Unlearning**

- Start
- End

Loss surface of unlearning dataset

**Training**

- Start
- End

Train the model to output **uniform probability vector** for data to unlearn

\[
\mathcal{F}(x) = V_{UNIFORM}
\]

**Example uniform probability vector**

- e.g. \([0.33, 0.33, 0.33]\)
Our Proposal: Functional Unlearning

\[ \min_{\theta} \left( \sum_{(x,y) \in C} \ell(\mathcal{F}(x), V_{UNIFORM}) \right) \]

unlearn cluster C

model fine tuning (1 month => 2 hours)

train the model to output \textcolor{orange}{uniform probability vector} for data to unlearn

\[ \mathcal{F}(x) = V_{UNIFORM} \]

\textcolor{orange}{uniform probability vector} e.g. [0.33, 0.33, 0.33]
Evaluation Results
## Experiment Setup

<table>
<thead>
<tr>
<th>Attack Name</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>BadNet</td>
<td>CIFAR10</td>
</tr>
<tr>
<td>BadNet</td>
<td>ImageNet</td>
</tr>
<tr>
<td>Trojan</td>
<td>VGG Face</td>
</tr>
<tr>
<td>Physical Backdoor (CVPR'21)</td>
<td>Wenger Face</td>
</tr>
</tbody>
</table>

### Evaluation metrics:
- **precision** of identifying poison data
- **recall** of identifying poison data

no known defense
## Results on backdoor attacks

<table>
<thead>
<tr>
<th>Attack Name</th>
<th>Task</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>BadNet</td>
<td>CIFAR10</td>
<td>99.5%</td>
<td>98.9%</td>
</tr>
<tr>
<td>BadNet</td>
<td>ImageNet</td>
<td>99.1%</td>
<td>99.1%</td>
</tr>
<tr>
<td>Trojan</td>
<td>VGG Face</td>
<td>99.8%</td>
<td>99.9%</td>
</tr>
<tr>
<td>Physical Backdoor (CVPR’21)</td>
<td>Wenger Face</td>
<td>99.5%</td>
<td>97.1%</td>
</tr>
</tbody>
</table>

> 97% precision and recall
Results on clean-label poison attacks

Our traceback system still works

<table>
<thead>
<tr>
<th>Attack Name</th>
<th>Task</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bullseye-Polytope (EuroSP’21)</td>
<td>CIFAR10</td>
<td>98.4%</td>
<td>96.8%</td>
</tr>
<tr>
<td>Bullseye-Polytope</td>
<td>ImageNet</td>
<td>99.3%</td>
<td>97.4%</td>
</tr>
<tr>
<td>Witches’ Brew (ICLR’21)</td>
<td>CIFAR10</td>
<td>99.7%</td>
<td>96.8%</td>
</tr>
<tr>
<td>Witches’ Brew</td>
<td>ImageNet</td>
<td>99.1%</td>
<td>97.9%</td>
</tr>
<tr>
<td>Malware Attack (USENIX’21)</td>
<td>Ember</td>
<td>99.2%</td>
<td>98.2%</td>
</tr>
</tbody>
</table>

Effective against 4 adaptive attacks

> 96% precision and recall

no known defense
One More Thing

- project webpage: sandlab.cs.uchicago.edu/forensics/
  • updated version of paper
  • code release on Github

- forensics for adversarial examples (CCS’22)
Questions?

Summary of this talk

- Forensics for data poisoning
- Identify responsible clusters
- Clustering and iterative pruning

sandlab.cs.uchicago.edu/forensics/