MTDeep: Moving Target Defense to Boost the Security of Deep Neural Networks Against Adversarial Attacks
Decision-time(/Test-time) Attacks

• If noise $\epsilon$ is strategically generated to intentionally make a classifier misclassify the modified input $i + \epsilon$, we consider these as attacks.

• Too much noise in input data can be easily detected (by humans). Thus, one has to ensure that $\epsilon$ is minimum (minimize $|\epsilon|_p$).


$l = Bus$

$\epsilon$

$i'$

$l = Ostrich$
Defense Techniques

• Train the neural net on the attack distribution (with the correct labels) and the classifier becomes immune to the particular type of adversarial inputs. This has been shown to be one of the most effective methods:

\[
\min_{\theta} \max_{\epsilon} \alpha L(\theta, x + \epsilon, y) + (1 - \alpha) L(\theta, x, y)
\]

Madry et. al. 2018

• Ensemble adversarial training – Use the constituent networks of an ensemble to get more adv. Input images. Use it to strengthen a network. Tramer et al., 2017

• Defenses may not be be effective against other attack methods eg. Adv. Universal Perturbation Moosavi et al., 2016
Moving Target Defense

• Moving Target Defense
  • Keep shifting; the attacker’s attack designed for a particular configuration does not work.
Moving Target Defense for Deep Neural Networks (MTDeep)

• Moving Target Defense
  • Keep shifting; the attacker’s attack designed for a particular configuration does not work.

• Assumes an attack designed for one classifier does not work well on the other.

• Starting with a set of classifiers, (how to?) randomly choose between them at test-time.
Can one attack kill all?

• Differential immunity ($\delta$)
  Given a set of classifiers $N$ and a set of attacks $U$, metric to measure worst-case transferability of an ensemble.

\[
\delta(U, N) = \min_u \frac{\max_n E(n, u) - \min_n E(n, u) + 1}{\max_n E(n, u) + 1}
\]

$0 \leq \delta \leq 1$

No differential immunity  High differential immunity
How to pick a classifier?
Let’s play a game!

\[ \min_{\theta} \max_{\epsilon} L(\theta, x + \epsilon, y) \]

\[ \min_{\theta} \max_{\epsilon} \alpha L(\theta, x + \epsilon, y) + (1 - \alpha) L(\theta, x, y) \]

\[ \min_{p} \max_{\epsilon} \sum_{i} p_i [\alpha L(\theta_i, x + \epsilon, y) + (1 - \alpha) L(\theta_i, x, y)] \]

Stackelberg equilibrium now solves the multi-objective function.
Increases defender’s utility, which

😊 Reduce misclassification rate on adversarial inputs.
😊 Limit the reduction in classification accuracy on legitimate samples.

<table>
<thead>
<tr>
<th>MTDDeep</th>
<th>Classification Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP</td>
<td>99.1</td>
</tr>
<tr>
<td>CNN</td>
<td>98.3</td>
</tr>
<tr>
<td>HRNN</td>
<td>98.7</td>
</tr>
</tbody>
</table>

(a) MNIST
How well does this work?

- MNIST (FGM, DF, PGD) +23.68
- F-MNIST (FGM, DF, PGD) +21.8
- ImageNET (Universal Perturbations) +20.68
MTDeep can be an Add-on Defense too!

- MTDeep can be used on top of any existing defense mechanisms.
- We strengthen the constituent networks with Ensemble Adversarial Training.

<table>
<thead>
<tr>
<th>Legitimate User (L)</th>
<th>MTDeep</th>
<th>Classification Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP&lt;sub&gt;eat&lt;/sub&gt;</td>
<td>97.99</td>
<td></td>
</tr>
<tr>
<td>CNN&lt;sub&gt;eat&lt;/sub&gt;</td>
<td>98.97</td>
<td></td>
</tr>
<tr>
<td>HRNN&lt;sub&gt;eat&lt;/sub&gt;</td>
<td>97.22</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Adversarial User (A)</th>
<th>FGM&lt;sub&gt;m&lt;/sub&gt;</th>
<th>FGM&lt;sub&gt;c&lt;/sub&gt;</th>
<th>FGM&lt;sub&gt;h&lt;/sub&gt;</th>
<th>DF&lt;sub&gt;m&lt;/sub&gt;</th>
<th>DF&lt;sub&gt;c&lt;/sub&gt;</th>
<th>DF&lt;sub&gt;h&lt;/sub&gt;</th>
<th>PGDM&lt;sub&gt;h&lt;/sub&gt;</th>
<th>PGDC&lt;sub&gt;h&lt;/sub&gt;</th>
<th>PGDH&lt;sub&gt;h&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLP&lt;sub&gt;eat&lt;/sub&gt;</td>
<td>95.06</td>
<td>75.32</td>
<td>70.1</td>
<td>1.5</td>
<td>96.97</td>
<td>95.73</td>
<td>0.00</td>
<td>88.00</td>
<td>69.00</td>
</tr>
<tr>
<td>CNN&lt;sub&gt;eat&lt;/sub&gt;</td>
<td>61.44</td>
<td>96.55</td>
<td>68.58</td>
<td>98.36</td>
<td>0.79</td>
<td>96.09</td>
<td>72.00</td>
<td>20.00</td>
<td>81.00</td>
</tr>
<tr>
<td>HRNN&lt;sub&gt;eat&lt;/sub&gt;</td>
<td>81.24</td>
<td>84.79</td>
<td>93.1</td>
<td>96.85</td>
<td>95.9</td>
<td>4.41</td>
<td>82.00</td>
<td>71.00</td>
<td>10.00</td>
</tr>
</tbody>
</table>

- We notice that Ensemble Adversarial Training (EAT) increases the differential immunity of the ensemble against the 3 attacks mentioned! (Why?)
Crafting attacks for an ensemble!

- Black-box attack Papernot et al., 2016
  - Design a distilled network that can capture the behavior of the MTDeep ensemble.
  - Design attacks on the distilled network.
  - See if the attack transfers to the MTDeep ensemble.

- Optimal white-box attacks are stronger than the black-box attack (MTDeep has 8% higher accuracy against black-box attacks compared to the optimal white-box attack for MNIST).
Higher differential immunity ($\delta$) yields higher gains in security.

<table>
<thead>
<tr>
<th>Networks</th>
<th>Differential Immunity ($\delta$)</th>
<th>Accuracy of Best Constituent Net</th>
<th>Accuracy of MTDeep</th>
<th>Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>FashionMNIST</td>
<td>0.11</td>
<td>3%</td>
<td>24.8%</td>
<td>21.8%</td>
</tr>
<tr>
<td>MNIST</td>
<td>0.19</td>
<td>0%</td>
<td>23.68%</td>
<td>23.68%</td>
</tr>
<tr>
<td>ImageNET</td>
<td>0.34</td>
<td>22.2%</td>
<td>42.88%</td>
<td>20.68%</td>
</tr>
<tr>
<td>MNIST + EAT</td>
<td>0.78</td>
<td>4.41%</td>
<td>54.71%</td>
<td>50.3%</td>
</tr>
</tbody>
</table>

Note that ImageNET has 1000 classes compared to 10 classes MNIST or Fashion-MNIST.
Conclusions and Future Work

- MTDeep: Moving Target Defense for Deep neural networks increases the an ensemble’s robustness to decision-time adversarial attacks.

\[
\min_p \max_\epsilon \sum_i p_i[\alpha L(\theta_i, x + \epsilon, y) + (1 - \alpha)L(\theta_i, x, y)]
\]

- How to obtain a good set of $\theta_i$’s? Adam et al 2018

\[
\min_{p, \theta} \max_\epsilon \sum_i p_i[\alpha L(\theta_i, x + \epsilon, y) + (1 - \alpha)L(\theta_i, x, y)]
\]

- Needs to consider differential immunity during training.
- Jointly optimize for policy and obtaining differentially immune networks?

sailiks@asu.edu