Automating Adversarial Robustness Testing of DNN Models

Presented by: Albert Negura

14/09/2022
Who are we?

- Albert Negura
  - Software Engineer, NavInfo Europe B.V., Eindhoven, Netherlands

- Kobus Grobler
  - Software Engineer, NavInfo Europe B.V., Eindhoven, Netherlands

- Adversarial robustness testing MLOps platform – GuardAI

- Adversarial machine learning for validation and testing AI models

- Focus on computer vision (automotive) use cases
Outline

- Vulnerabilities of AI models
- Adversarial Robustness – Security and Trustworthiness of AI Models
- Testing Coverage
- Practical considerations
Vulnerabilities of AI Models

Hacker Goals:
- Steal training data
- Steal model performance / weights
- Create a backdoor in model inference
- Fool the model’s decision making
Vulnerabilities of AI Models

Models were shown to be vulnerable to (evasion) attacks. Consequences:

- Detecting vehicles
Vulnerabilities of AI Models

Models were shown to be vulnerable to (evasion) attacks.

Consequences:

- Traffic sign detection
Vulnerabilities of AI Models

Models were shown to be vulnerable to (evasion) attacks.

Consequences:

- Production line faults

Vulnerabilities of AI Models

Models were shown to be vulnerable to (evasion) attacks.

Consequences:

- Sentiment analysis providing incorrect (costly) conclusions

- Bought several cans of dog food. Found these to be good quality products, our labrador appreciates this to other products.
Vulnerabilities of AI Models

Models were shown to be vulnerable to (evasion) attacks.

Consequences:

- Bypass medical diagnosis
- Keywords to trick email spam filters
- Evade ML-based malware detection
- And so on...
Vulnerabilities of AI Models

But are these practical?

- Printable patch attacks (T-shirts, masks, shapes in specific positions)
- Transferable attacks (exploiting vulnerabilities to learned features, evasion attacks on extracted model)
Adversarial Robustness

How to measure?

- Traditionally: Loss in performance vs distortion vs perceptibility (zero-knowledge, full-knowledge scenarios)

- Realistically: Adversarially-valid examples have unique features – no one-size-fits-all attack per scenario; need to test over entire space of adversarial attacks applicable to vulnerability case
Adversarial Test Coverage

Robustness measurements require annotated data

Annotated data can be expensive to obtain → can we measure robustness in an online fashion?
Adversarial Test Coverage

<table>
<thead>
<tr>
<th>Task</th>
<th>Minimum samples needed for Zero Knowledge Attacks</th>
<th>Minimum samples needed for Full Knowledge Attacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Classification</td>
<td>$\approx 300$ (3% of validation size)</td>
<td>$\approx 100$ (1% of validation size)</td>
</tr>
<tr>
<td>Semantic / Instance Segmentation</td>
<td>$\approx 600$ (20% of validation size)</td>
<td>$\approx 600$ (20% of validation size)</td>
</tr>
<tr>
<td>Object Detection and Localization</td>
<td>$\approx 300$ (12% of validation size)</td>
<td>$\approx 100$ (4% of validation size)</td>
</tr>
<tr>
<td>Depth Estimation</td>
<td>$\approx 300$ (15% of validation size)</td>
<td>$\approx 300$ (15% of validation size)</td>
</tr>
<tr>
<td>Sentiment Analysis</td>
<td>$\approx 600$ (30% of validation size)</td>
<td>$\approx 200$ (10% of validation size)</td>
</tr>
</tbody>
</table>

- Results are just examples for specific datasets/models used
- Models used similar backbones and training procedure (Resnet50)
- Attacks: FGSM, PGD, Deepfool, SimBA, Square Attack (image), Boundary Attack
Practical considerations

Difficult to generalize adversarial attacks

- Single-Modal
- Multi-Modal
- Cross-Modal

Adversarial Criteria
- Printability
- Perceptibility
- Distortion Bounds
- Target domain
- Attacked model architectures

- Targeted
- Untargeted
- Specificity of Attack
- Parameterization
- Hyperparameter Optimization

Attack is very perceptible without good adversarial criteria

Yamanaka, K. et al. (2020). Adversarial Patch Attacks on Monocular Depth Estimation Networks

ETSIC 9th UCAAT
Practical considerations

Different frameworks
Practical considerations

Different dataset formats

Different Folder Structures

Different Annotations
# Practical considerations

## Multiple metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>Harmonic robustness</th>
<th>mIoU</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Attack report</td>
<td>Attacked</td>
<td>Robustness</td>
</tr>
<tr>
<td></td>
<td>Attack report</td>
<td>0.68689</td>
<td>0.899403</td>
</tr>
<tr>
<td></td>
<td>Project 105 Dataset VOCSegmentation-2007</td>
<td>0.16057</td>
<td>0.15846</td>
</tr>
<tr>
<td></td>
<td>Project 105 Dataset VOCSegmentation-2007</td>
<td>0.30901</td>
<td>0.15846</td>
</tr>
<tr>
<td></td>
<td>Project 105 Dataset VOCSegmentation-2007</td>
<td>0.24900</td>
<td>0.15846</td>
</tr>
</tbody>
</table>

![Graph showing robustness PGD/linf](image-url)
Practical considerations

Adversarial Robustness Evaluation tests

Training system -> ML system -> Prediction system -> Serving system

Model inputs -> Model predictions

Adversarial (Evasion) Entry Point

Offline Online

Storage and preprocessing system -> Labelling system -> Production data

Feedback
Testing methodology

Integrate robustness testing into the model development pipeline

- Use a platform that works with existing CI/CD systems. Basic requirements:
  - Exposes an API to enable automation
  - Definition of test pass/fail criteria based on a single robustness metric
  - Enables parameterization of attacks and noises
  - Generation of adversarial samples
  - Definition of custom transforms to ease dataset matching with model inputs
  - Visualization
Any further questions?

Contact us:

albert.negura@navinfo.eu  https://linkedin.com/in/albert-negura
kobus.grobler@navinfo.eu  https://linkedin.com/in/kobus-grobler