9th UCAAT User Conference on Advanced Automated Testing

Automating Adversarial Robustness Testing of DNN Models

Presented by: Albert Negura

NAVINFO

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, Netherland
- Adversarial robustness testing MLOps platform - GuardAI
- Adversarial machine learning for validation and testing AI models
- Focus on computer vision (automotive) use cases

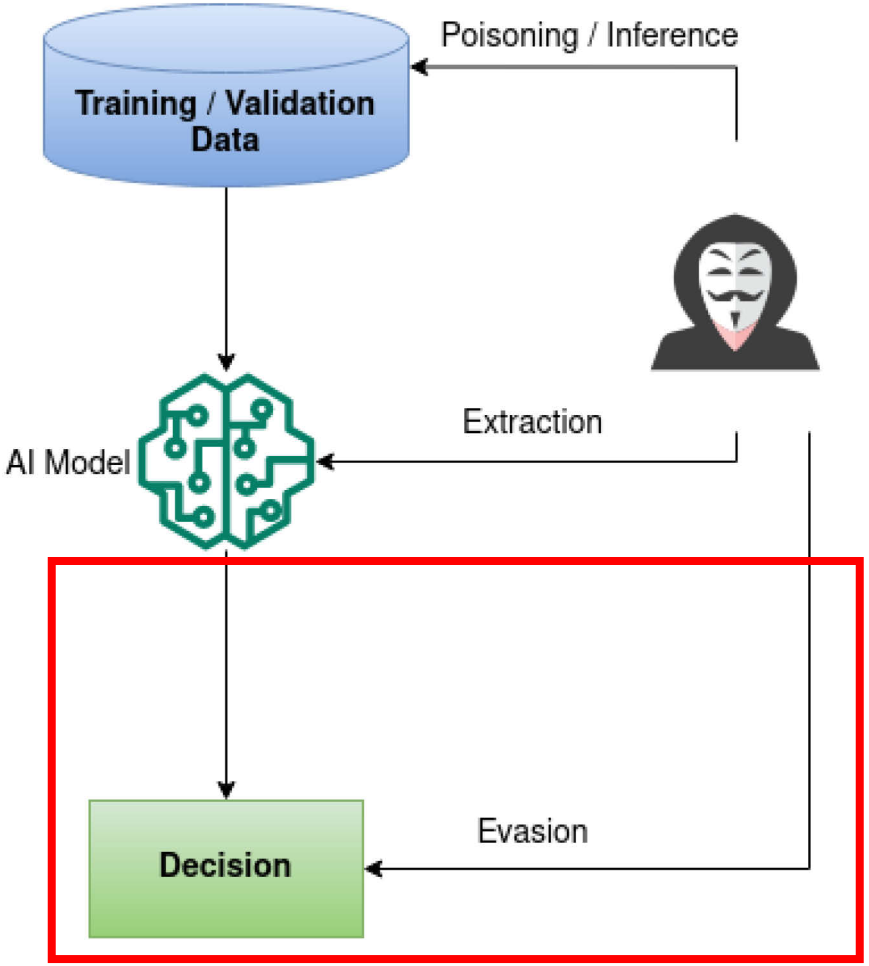
Outline

- 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

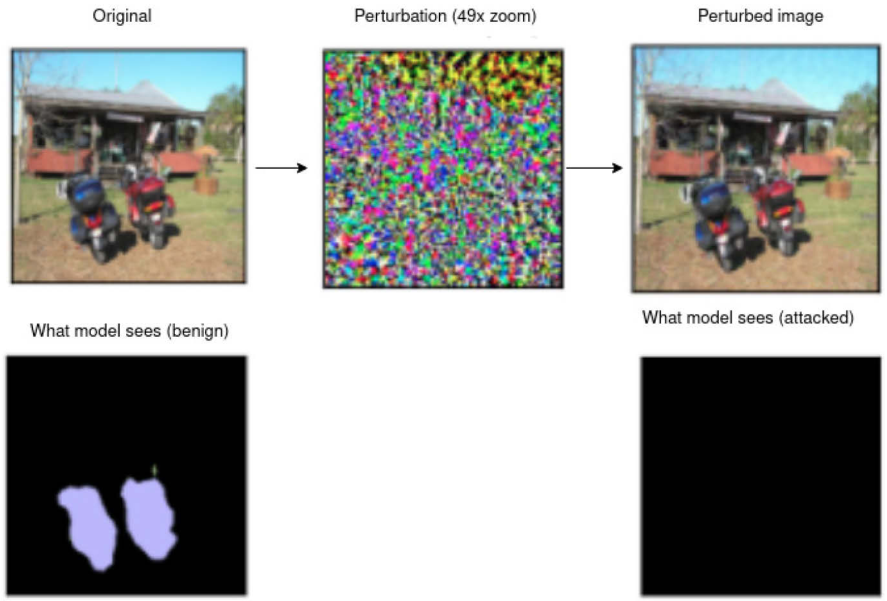


Vulnerabilities of AI Models

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

Consequences:

- Detecting vehicles

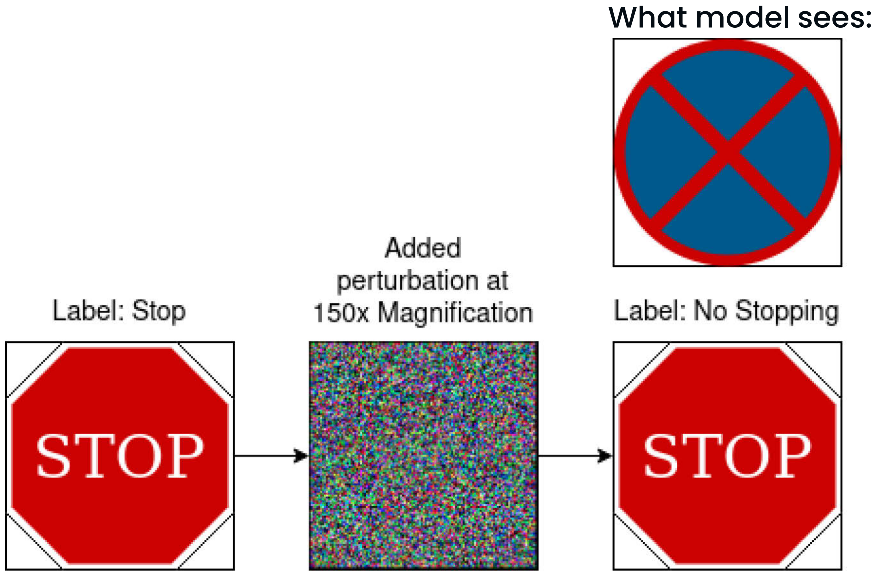


Vulnerabilities of AI Models

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

Consequences:

- Traffic sign detection

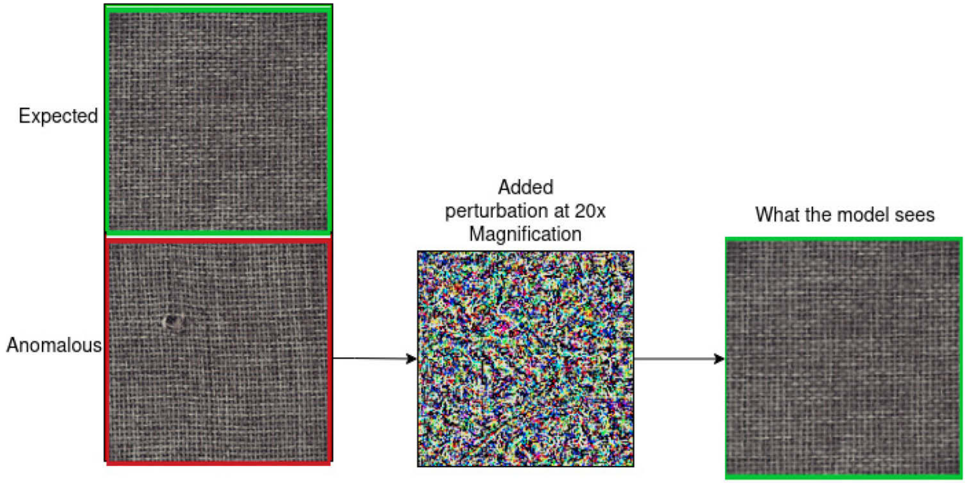


Vulnerabilities of AI Models

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

Consequences:

- Production line faults



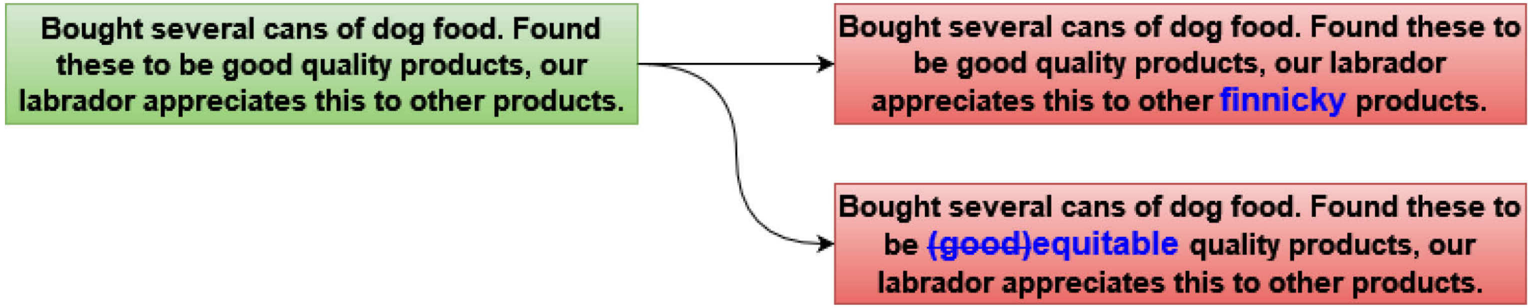
Bergmann P. et al. (2021) : The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection

Vulnerabilities of AI Models

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

Consequences:

- Sentiment analysis providing incorrect (costly) conclusions



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?

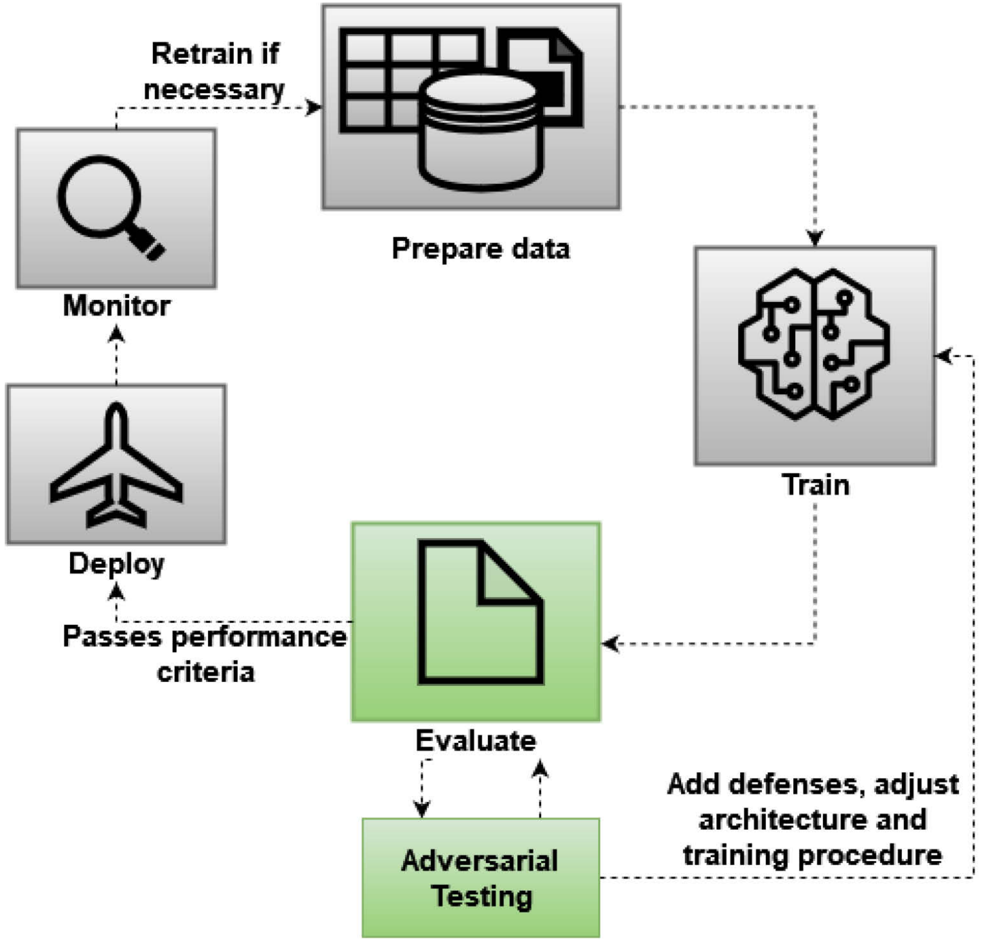
Kumar, R.S.S. et al. (2020). *Adversarial Machine Learning - Industry Perspective*	
Which attack would affect your org the most?	Distribution
Poisoning	10
Model Stealing	6
Model Inversion	4
Backdoored ML	4
Membership Inference	3
Adversarial Examples	***2***
Reprogramming ML System	0
Adversarial Example in the Physical Domain	0
Malicious ML provider recovering training data	0
Attacking the ML supply chain	0
Exploit Software Dependencies	0

- 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?

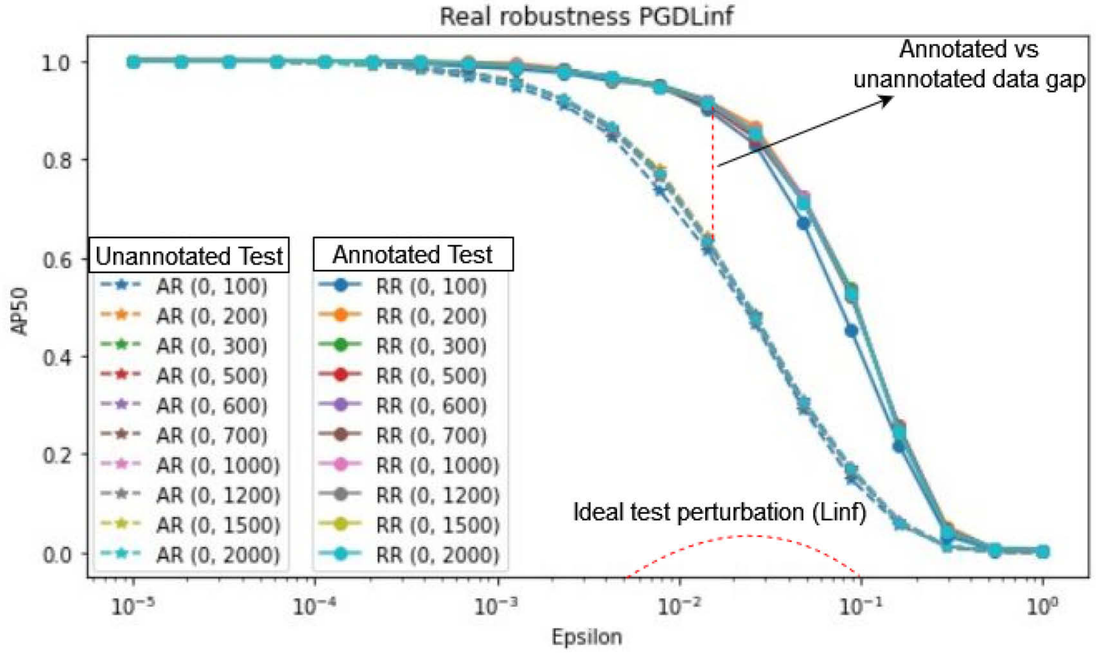
- Traditionally: Loss in performance vs distortion vs perceptibility (zero-knowledge, full-knowledge scenarios)
- Realistically: Adversarially-valid examples have unique features – no one-size-fits-all attack per scenario; need to test over entire space of adversarial attacks applicable to vulnerability case



Adversarial Test Coverage

Robustness measurements require annotated data

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



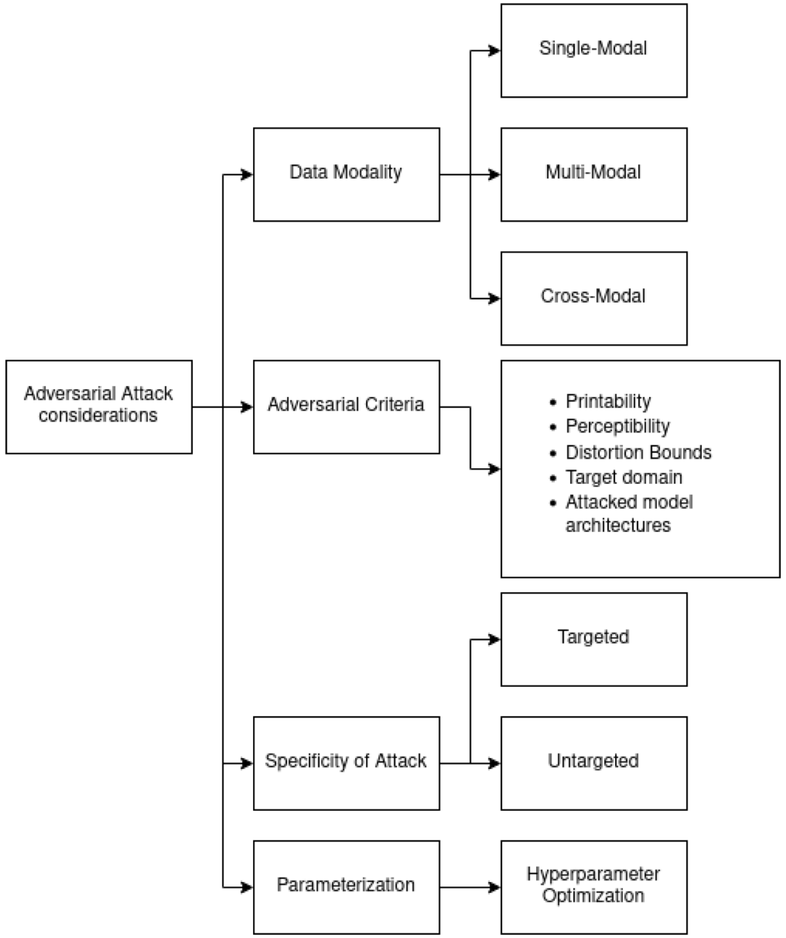
Adversarial Test Coverage

Task	Minimum samples needed for Zero Knowledge Attacks	Minimum samples needed for Full Knowledge Attacks
Image Classification	~300 (3% of validation size)	~100 (1% of validation size)
Semantic / Instance Segmentation	~600 (20% of validation size)	~600 (20% of validation size)
Object Detection and Localization	~300 (12% of validation size)	~100 (4% of validation size)
Depth Estimation	~300 (15% of validation size)	~300 (15% of validation size)
Sentiment Analysis	~600 (30% of validation size)	~200 (10% of validation size)

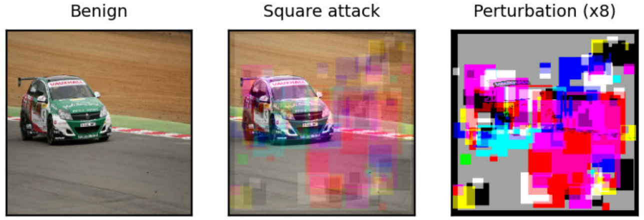
- Results are just examples for specific datasets/models used
- Models used similar backbones and training procedure (Resnet50)
- Attacks: FGSM, PGD, Deepfool, SimBA, Square Attack (image), Boundary Attack

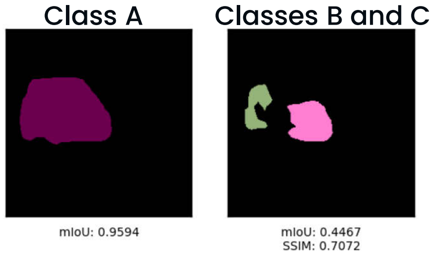
Practical considerations

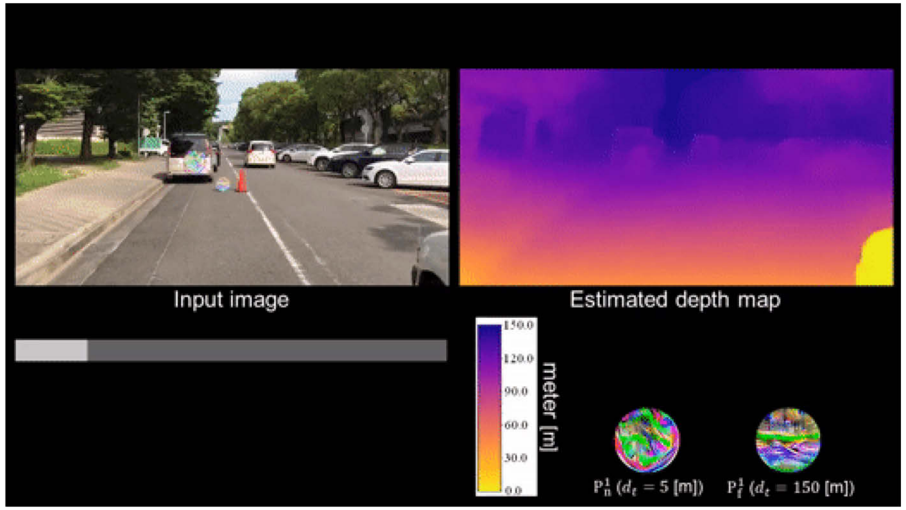
Difficult to generalize adversarial attacks



Attack is very perceptible without good adversarial criteria







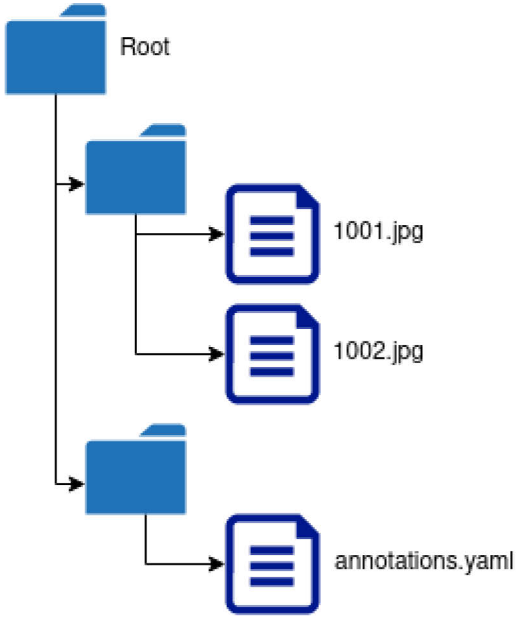
Yamanaka, K. et al. (2020). *Adversarial Patch Attacks on Monocular Depth Estimation Networks*

Practical considerations

Different frameworks

Practical considerations

Different dataset formats




```
"annotations": [
    {
        "segmentation": [[510.66,423.01,511.72,420.03,...,510.45,423.01]],
        "area": 702.1057499999998,
        "iscrowd": 0,
        "image_id": 289343,
        "bbox": [473.07,395.93,38.65,28.67],
        "category_id": 18,
        "id": 1768
    },
    ...
    {
        "segmentation": {
            "counts": [179,27,392,41,…,55,20],
            "size": [426,640]
        },
        "area": 220834,
        "iscrowd": 1,
        "image_id": 250282,
        "bbox": [0,34,639,388],
        "category_id": 1,
        "id": 900100250282
    }
]
```

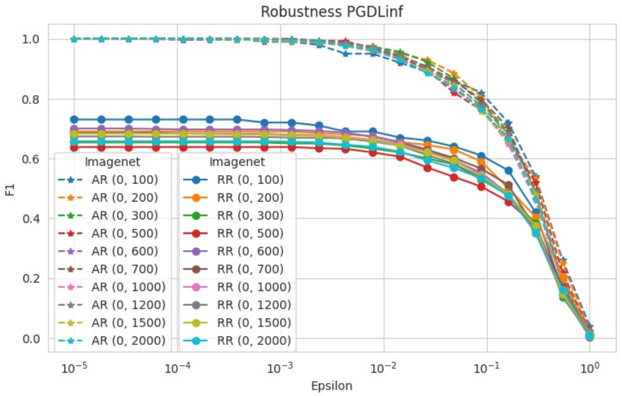
```
<annotation>
        <folder>Kangaroo</folder>
        <filename>00001.jpg</filename>
        <path>./Kangaroo/stock-12.jpg</path>
        <source>
                <database>Kangaroo</database>
        </source>
        <size>
                <width>450</width>
                <height>319</height>
                <depth>3</depth>
        </size>
        <segmented>0</segmented>
        <object>
                <name>kangaroo</name>
                <pose>Unspecified</pose>
                <truncated>0</truncated>
                <difficult>0</difficult>
                <bndbox>
                        <xmin>233</xmin>
                        <ymin>89</ymin>
                        <xmax>386</xmax>
                        <ymax>262</ymax>
                </bndbox>
        </object>
</annotation>
```

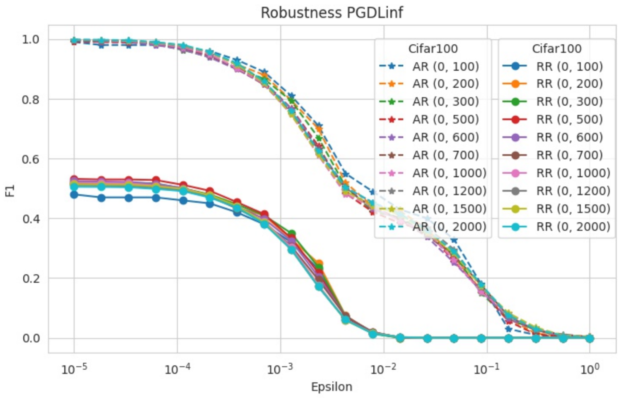

Different Folder Structures

Different Annotations

Practical considerations

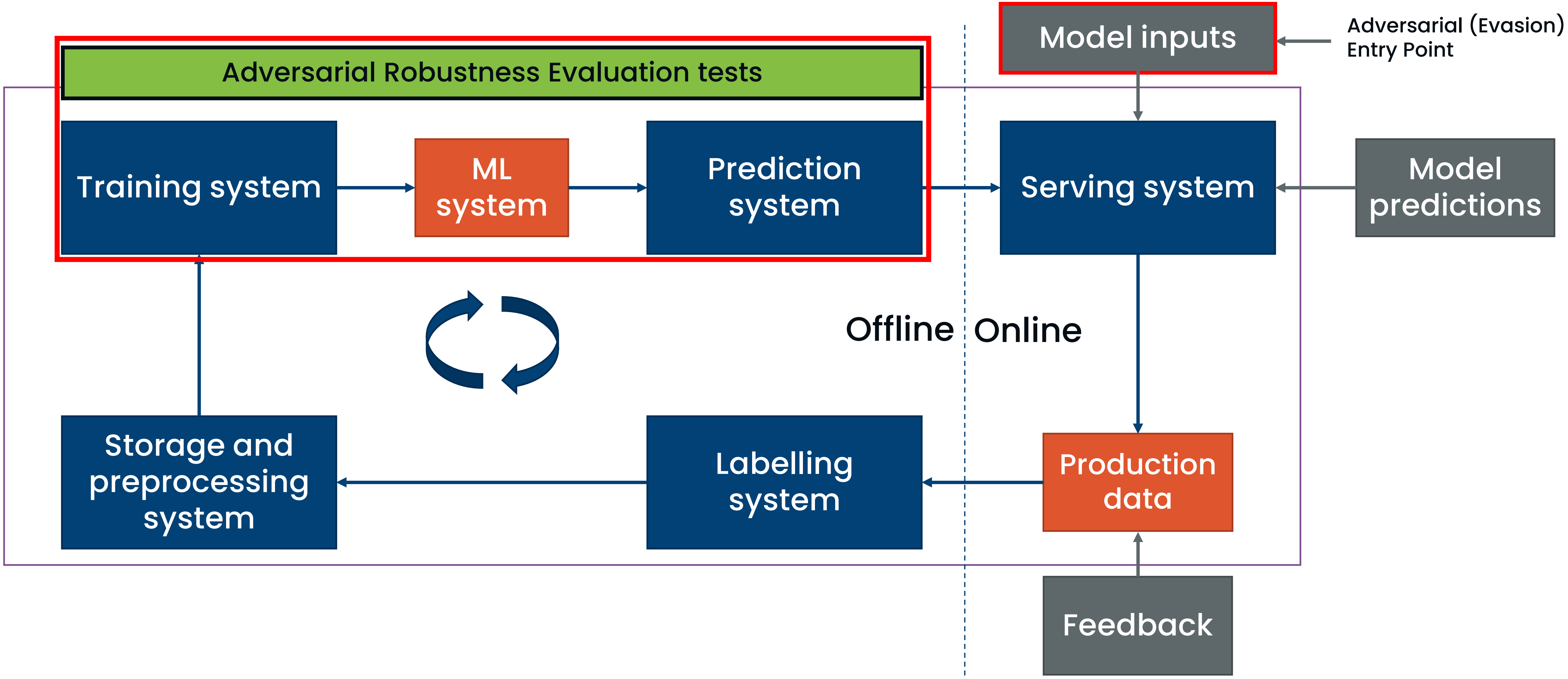
Multiple metrics





Metric	Harmonic robustness	mIoU	SSIM
Attack report	Attacked	Robustness-attacked	Attacked
Run Name Square Attack Test 8 Project: 105 Dataset: VOCSegmentation-2007 Items: 6	1.08086	0.99120	0.93163
Run Name Square Attack Test 7 Project: 105 Dataset: VOCSegmentation-2007 Items: 6	0.30861	0.18246	0.00851
Run Name Square Attack Test 6 Project: 105 Dataset: VOCSegmentation-2007 Items: 6	0.30861	0.18246	0.00873
Run Name Square Attack Test 5 Project: 105 Dataset: VOCSegmentation-2007 Items: 6	0.24488	0.51766	0.59448

Practical considerations



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