AI Security Metrics and Threats

Harold Booth

Computer Scientist
National Institute of Standards and Technology
Computer Security Division

Paul Rowe

Principal Cyber Resiliency Researcher
The MITRE Corporation

NIST NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

Disclaimer

NIST

Certain commercial equipment, instruments, or materials are identified in this presentation to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

Acknowledgements

Joint work with NCCoE Dioptra Team

Special thanks to:

James Glasbrenner   Andrew Hand

Howard Huang   Julian Sexton

for code development and contributions to slides

Dataset Sources

MNIST :
http://yann.lecun.com/exdb/mnist/

Fruits360 :
https://www.kaggle.com/datasets/moltean/fruits

ImageNet :
https://image-net.org/index.php

Road Sign Detection
https://makeml.app/datasets/road-signs

Introduction

Image: pixabay/openclipart-vectors-30363

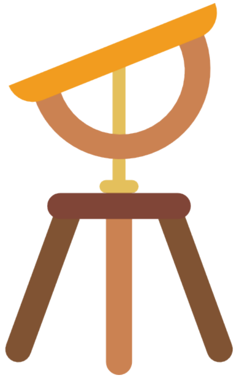
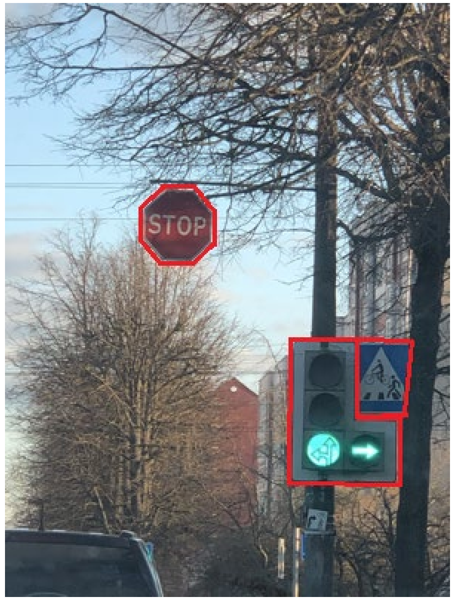


Image: Flaticon.com/Smashicons

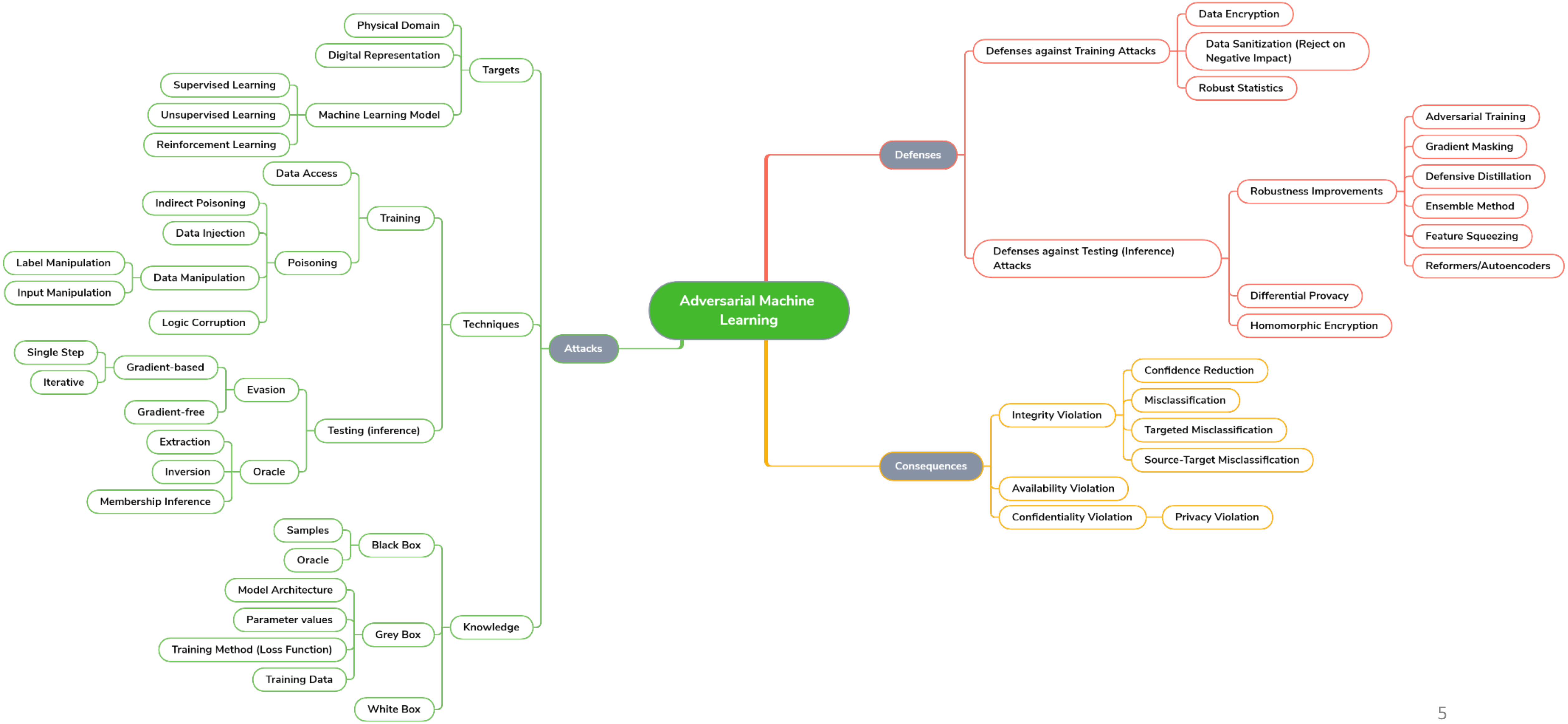
Use Cases

Image: pixabay/Nadin Dunnigan

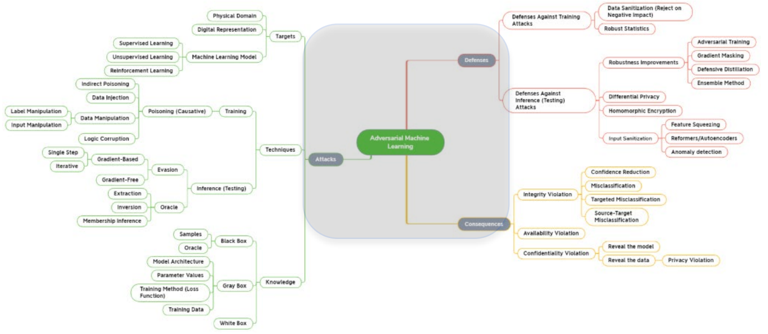


From Road Sign Detection Dataset

A Taxonomy and Terminology of Adversarial Attacks and Defenses on Machine Learning (Draft NISTIR 8269)



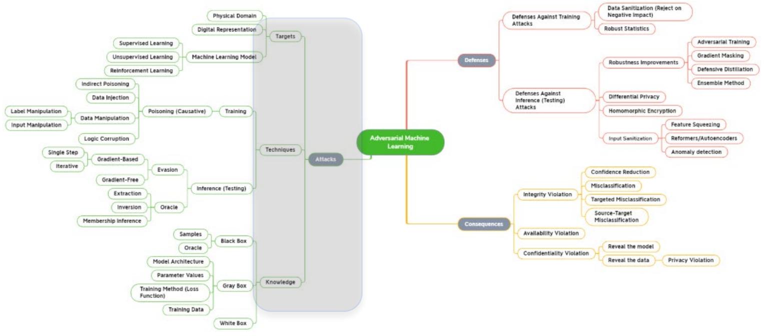
A Taxonomy and Terminology of Adversarial Attacks and Defenses on Machine Learning (Draft NISTIR 8269)



Adversarial Machine Learning

- Attacks
- Defenses
- Consequences

A Taxonomy and Terminology of Adversarial Attacks and Defenses on Machine Learning (Draft NISTIR 8269)



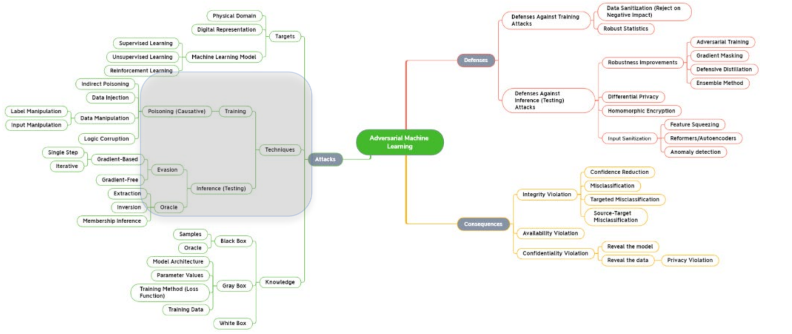
Attacks

Targets

Techniques

Knowledge

A Taxonomy and Terminology of Adversarial Attacks and Defenses on Machine Learning (Draft NISTIR 8269)



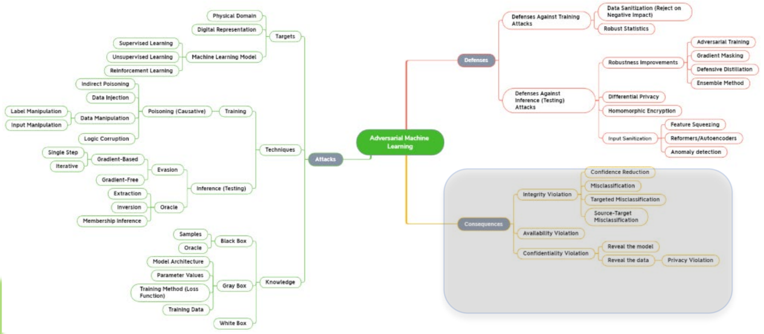
Techniques

- Poisoning
- Evasion
- Oracle

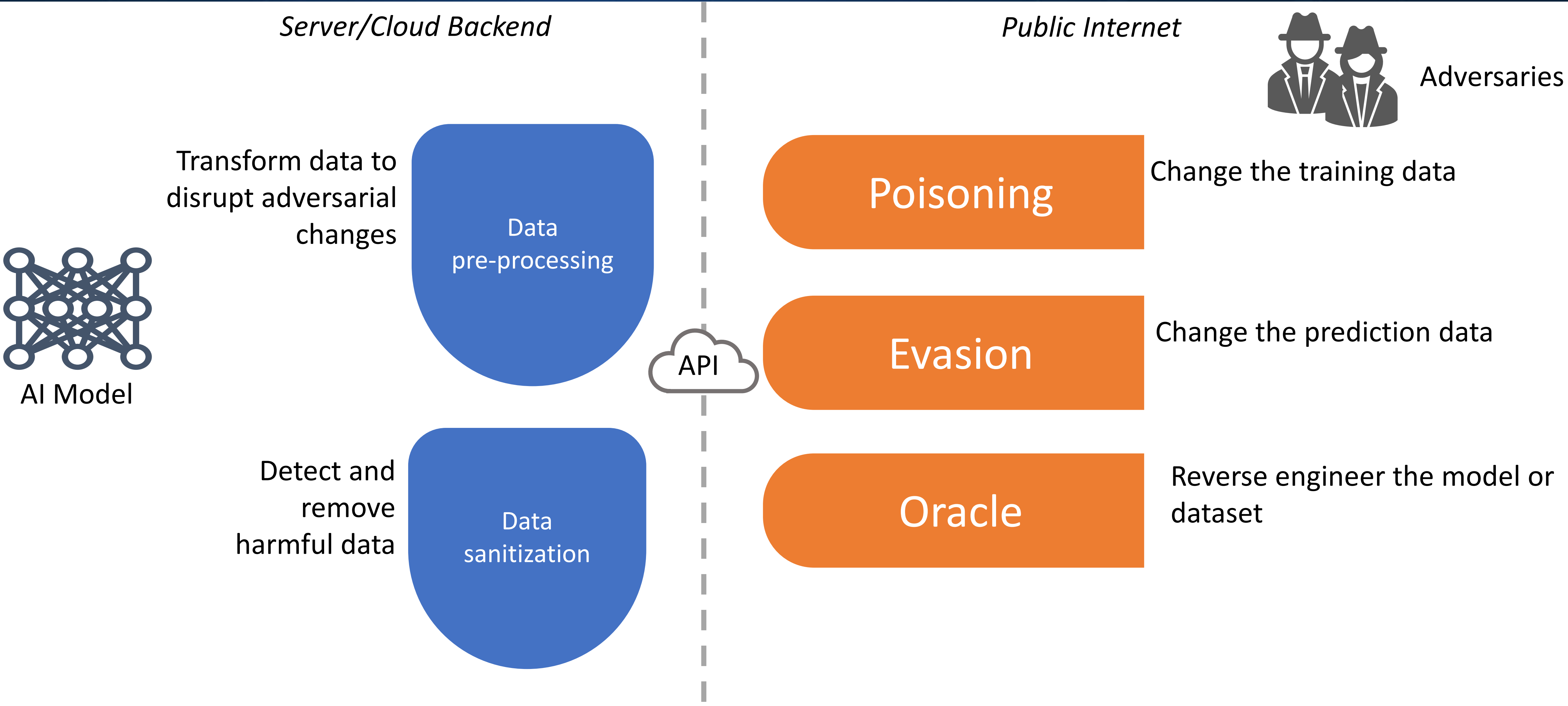
A Taxonomy and Terminology of Adversarial Attacks and Defenses on Machine Learning (Draft NISTIR 8269)

Consequences

- Confidentiality
  - Model
  - Data
- Integrity
  - Confidence
  - Misclassifications
- Availability

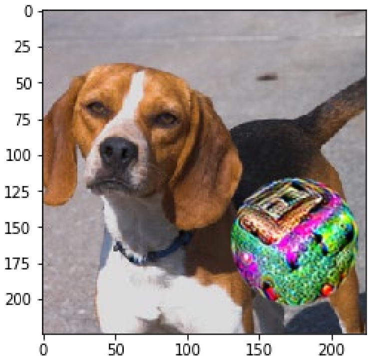


A wide variety of attacks and mitigation strategies



Visual Sampling of Attacks

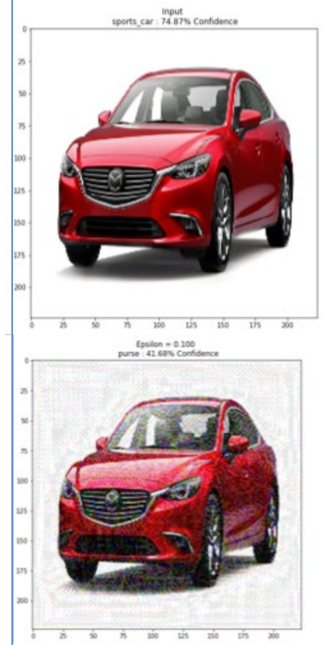
Patch Evasion



Predictions:

Toaster	30%
Beagle	11%
Spatula	8%

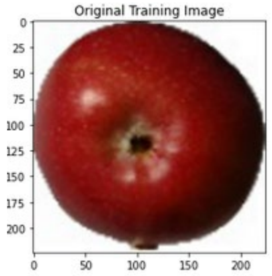
Noise Evasion



sports car: 75%

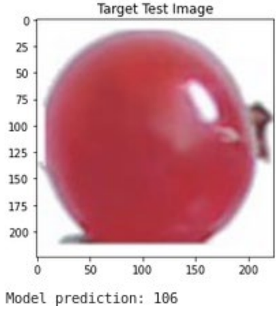
purse: 41%

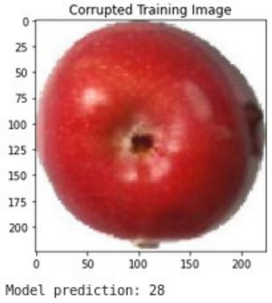
Clean Label Poisoning



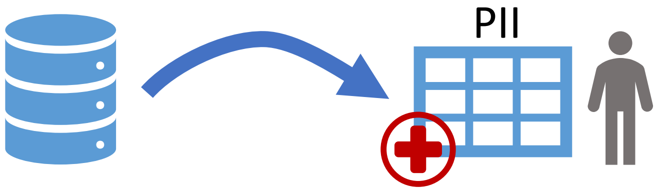
Original:
Apple

Target:
Currant





Poisoned
Apple

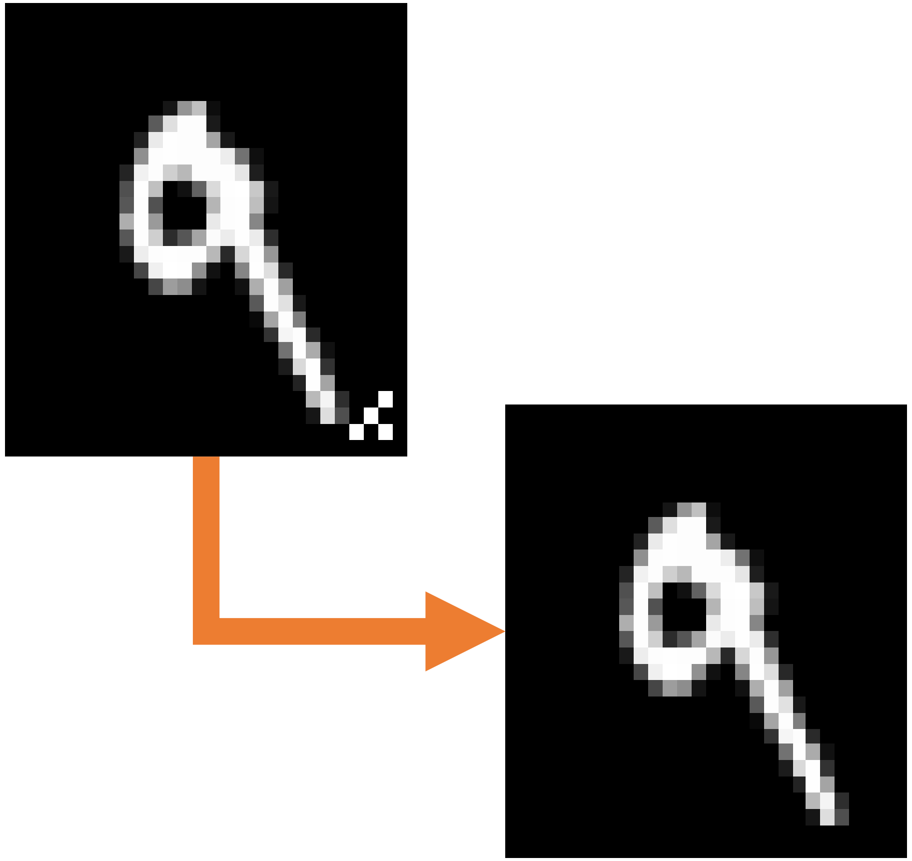


Images From ImageNet and Fruits360 Dataset

Visual Sampling of Defenses

Image pre-processing

Data Sanitization



Images From ImageNet and MNIST Datasets

Evaluation Metrics

Accuracy

Effectiveness

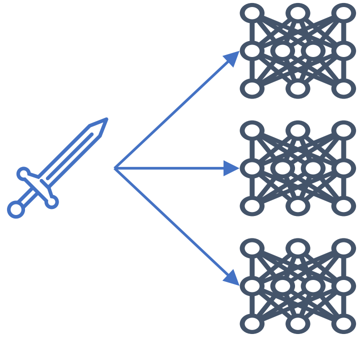
Sensitivity

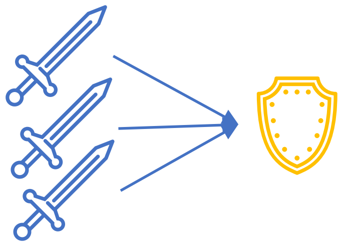
Transferability

Generalizability

Perturbation Distance

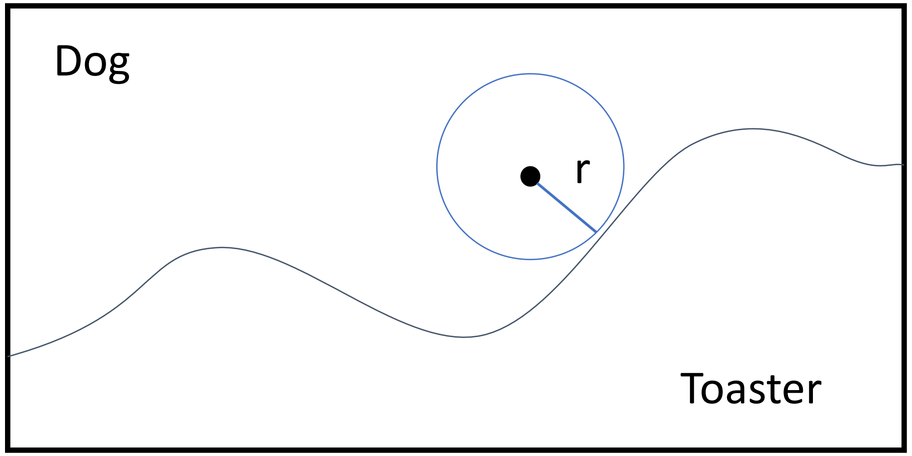
Time & Resources

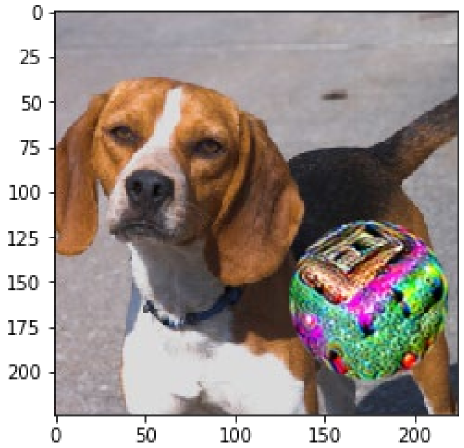


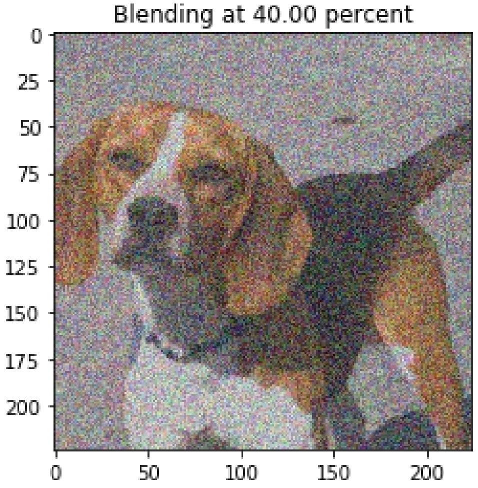


Accuracy Loss

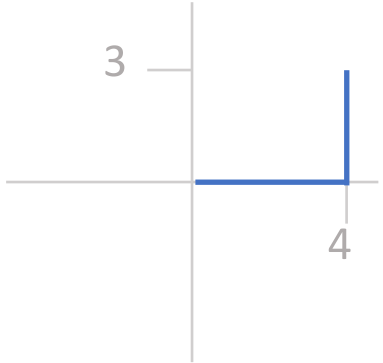
Details on Metrics



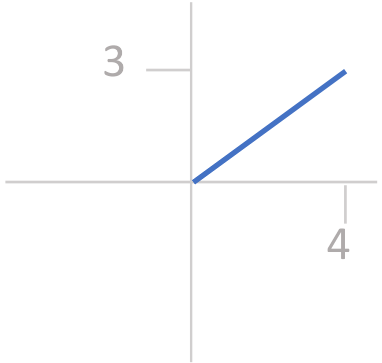




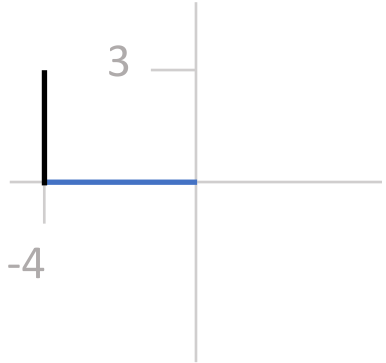
L1 - Manhattan



L2 - Euclidean



L-infinity



Images From Fruits360 and ImageNet Datasets

Deployment context also matters!

Security Checkpoint

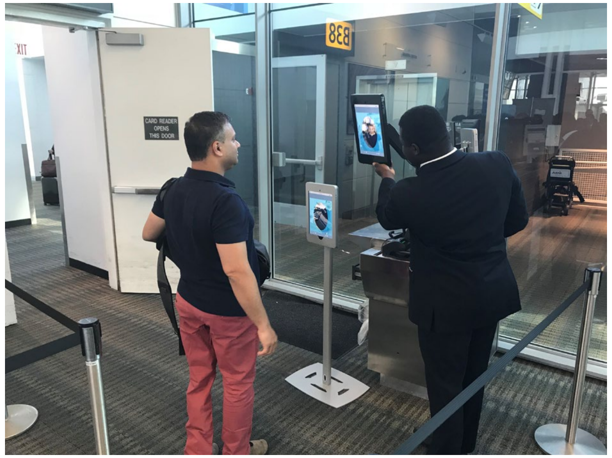


Image: Gregory Wallace/CNN

Controlled environment

Automated Driving

Image: Shutterstock

Open environment

Image Forensics

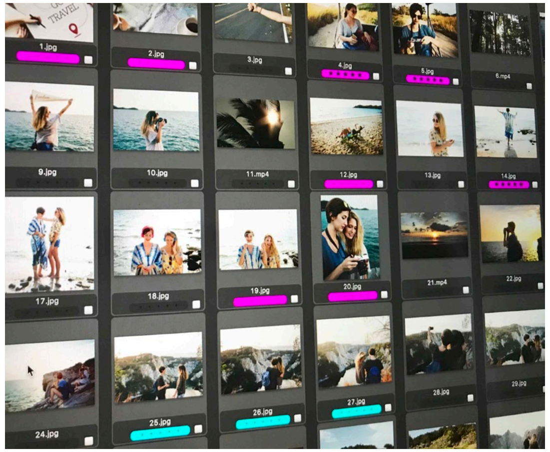
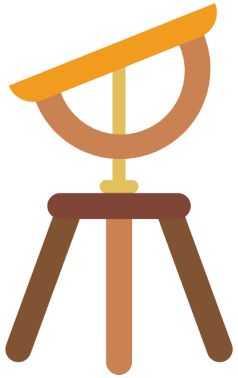


Image: Caroline Guntur

Open environment

- What is the task?
- Where (and when) can you control the environment?
- What is an adversary’s objective?
- What components does the AI depend on?
- What functions depend on the AI?

Scenario testing, Parameter Sweeping, Evaluation

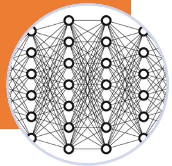


Dioptra

Image: Flaticon.com/Smashicons

- Shallow Net
- AlexNet
- LeNet
- ResNet50
- VGG16
- …

Training Architecture



- Patch augmentation
- Poison Frogs
- Adversarial training
- …

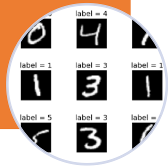
Data Augmentation

- Spatial smoothing
- Defensive distillation
- …

Inference pre-processing

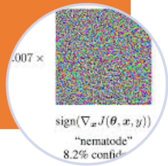
- MNIST
- Fruits360
- ImageNet
- …

Dataset



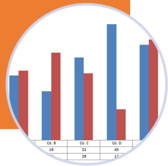
- Fast Gradient Method
- Pixel Threshold
- Patch
- Membership Inference
- …

Attack on trained model

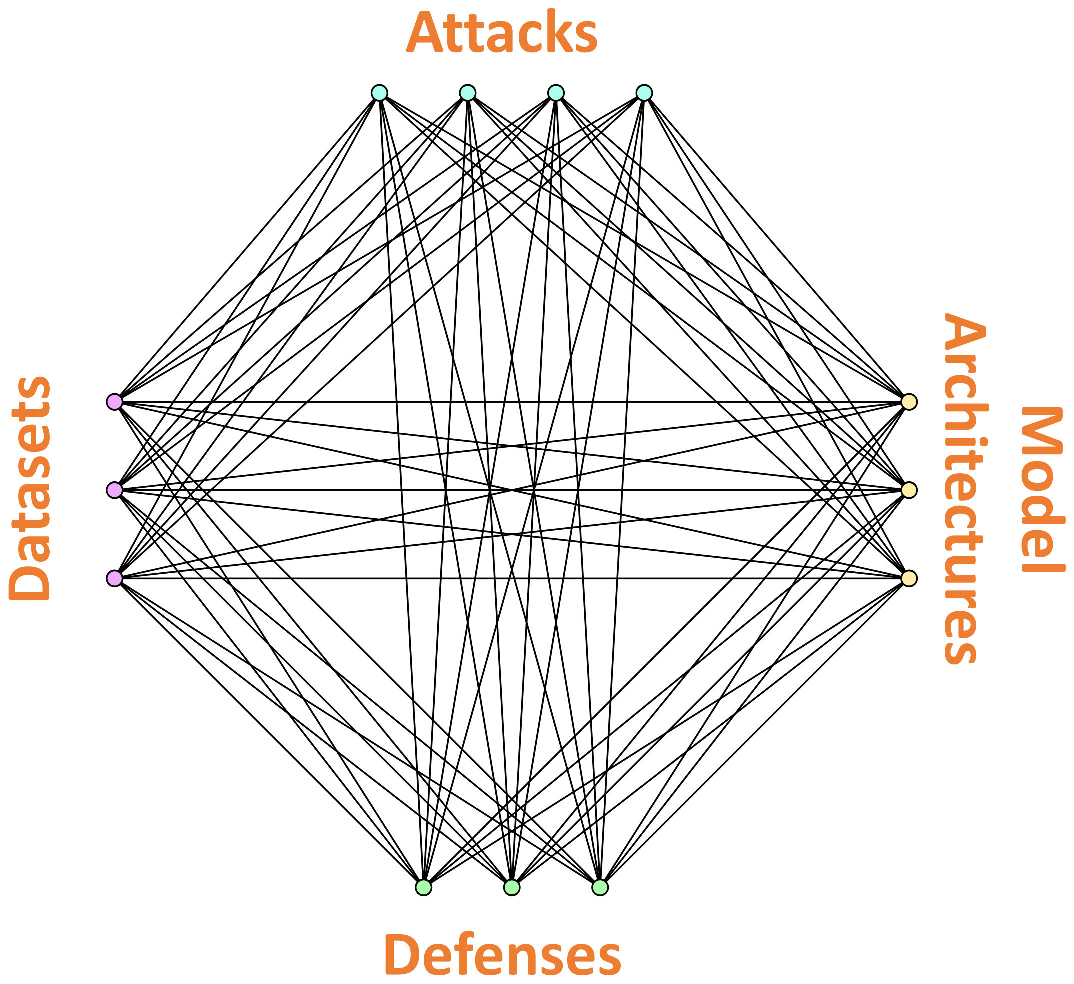


- Clean accuracy
- Adversarial accuracy
- Robustness radius
- …

Metric



Use Case Exploration



Use Cases

- 2nd party testing
  - Supermarket CTO looking to purchase image-based pricing solution
- Development and regression testing
  - Developers building road sign detection and recognition solution

2^{nd} Party Testing

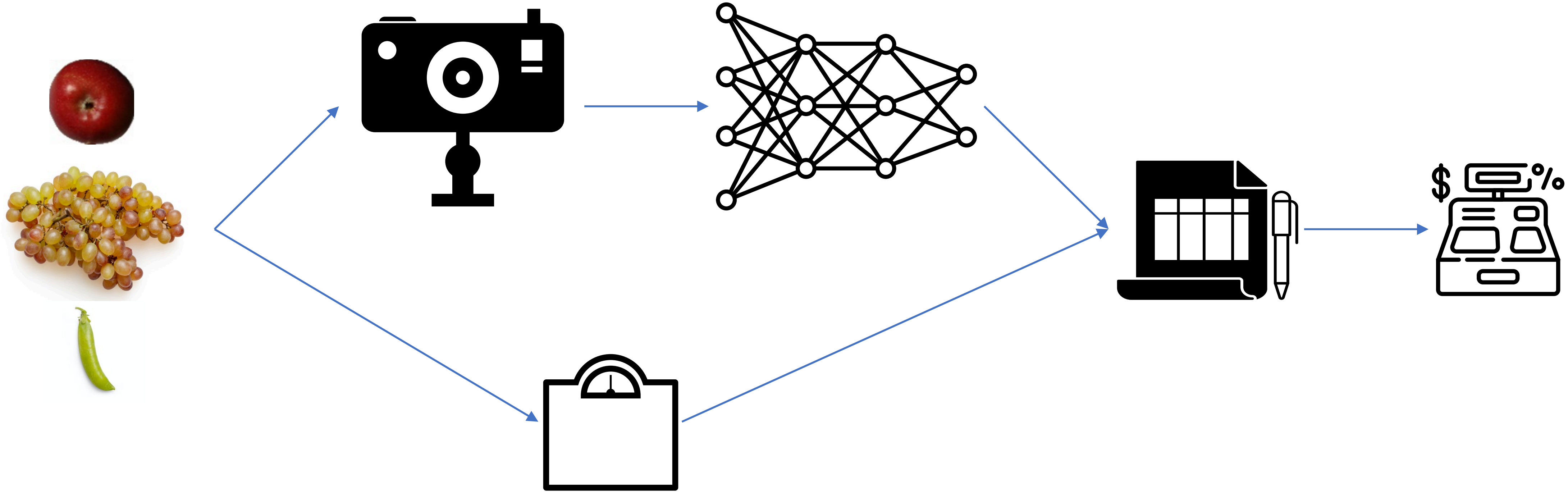
Risk Assessment Process

- Identify task
- Is AI necessary?
- Identify threat & deployment assumptions
- Which attacks are still relevant?
- Identify metrics applicable to highest priority risks
- Build experiments and synthesize results

Image-based pricing of produce

Photo Credit: Rows Of Fresh Fruit In Eco-friendly Boxed by Anna Ivanova from NounProject.com

Image-Based Pricing of Produce



Threat/Deployment Model

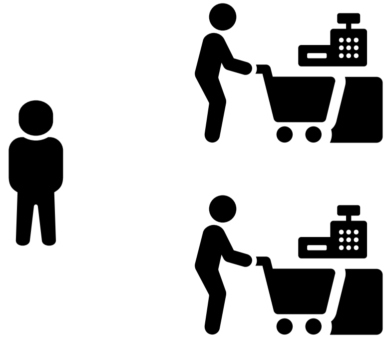
$ $ $

Image: NounProject.com

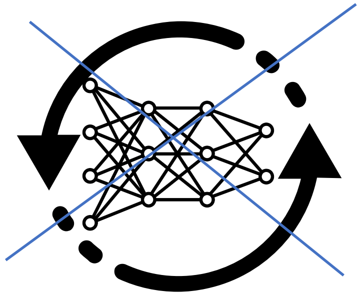
$

Image From Fruits360 Dataset

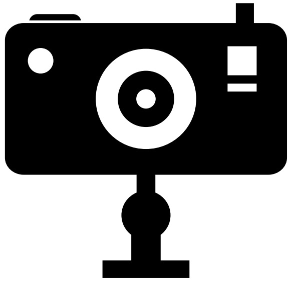
Costly misclassification



Attended self-checkout



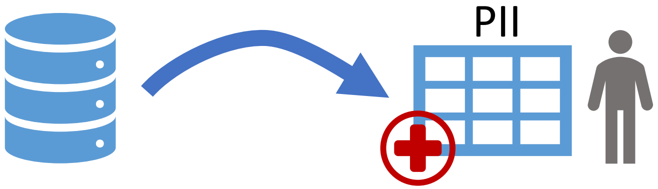
No online learning



Freshly generated
digital representations

Supermarket example: Attack Profile

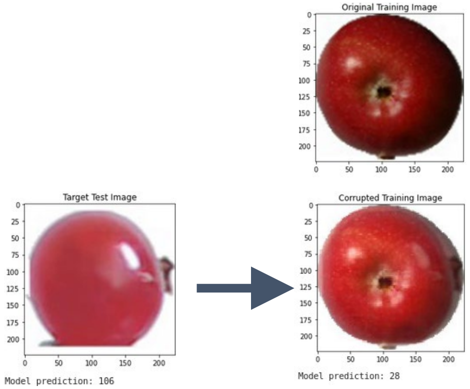
Digital manipulation attacks are difficult in our deployment setting.



Training data is not sensitive.
Model extraction attacks would be easy to detect.

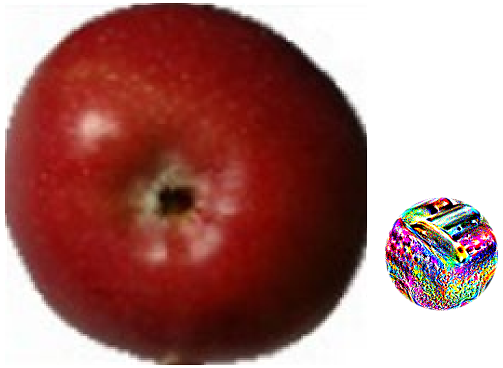
Original: Apple

Target: Currant



Poisoned Apple

Data may be poisoned in supply chain.



Physical manipulation attacks are easier.
Attendants can be trained to look out for them.

Images From MNIST and Fruits360 Dataset

Supermarket example: Some relevant metrics

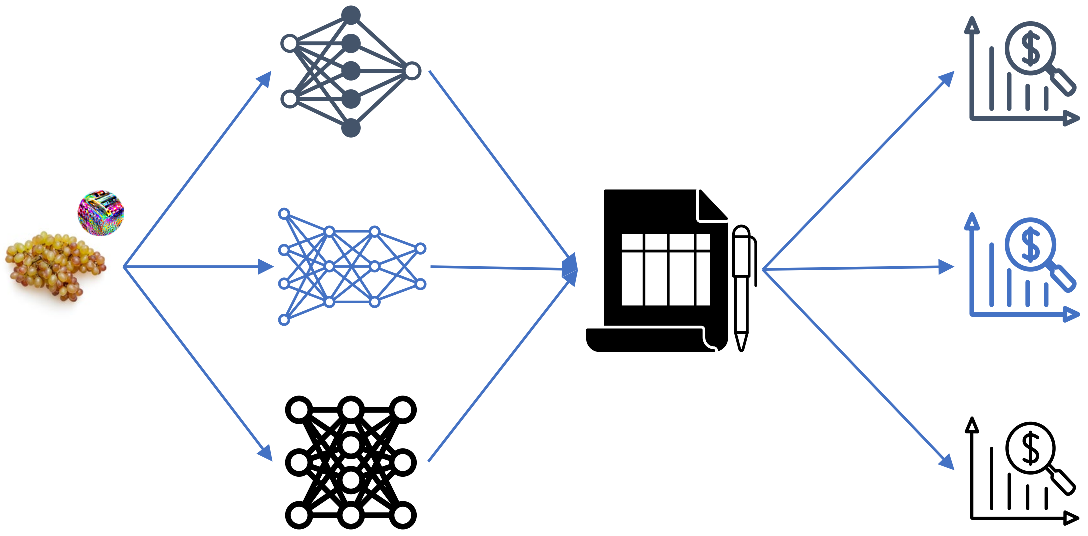
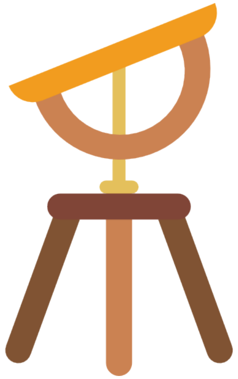


Image From Fruits360 Dataset

Fruit Classification Combinations Tested

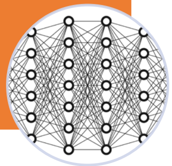


Dioptra

Image: Flaticon.com/Smashicons

- Vendor 1 model (VGG16)
- Vendor 2 model (Defended VGG16)

Training Architecture



- Adversarial training

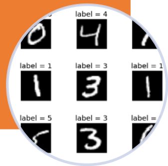
Data Augmentation

- None

Inference pre-processing

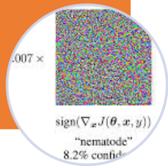
- Fruits360

Dataset



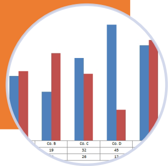
- Patch

Attack on trained model



- Clean accuracy
- Adversarial accuracy
- Cost per misclassification

Metric



Dataset

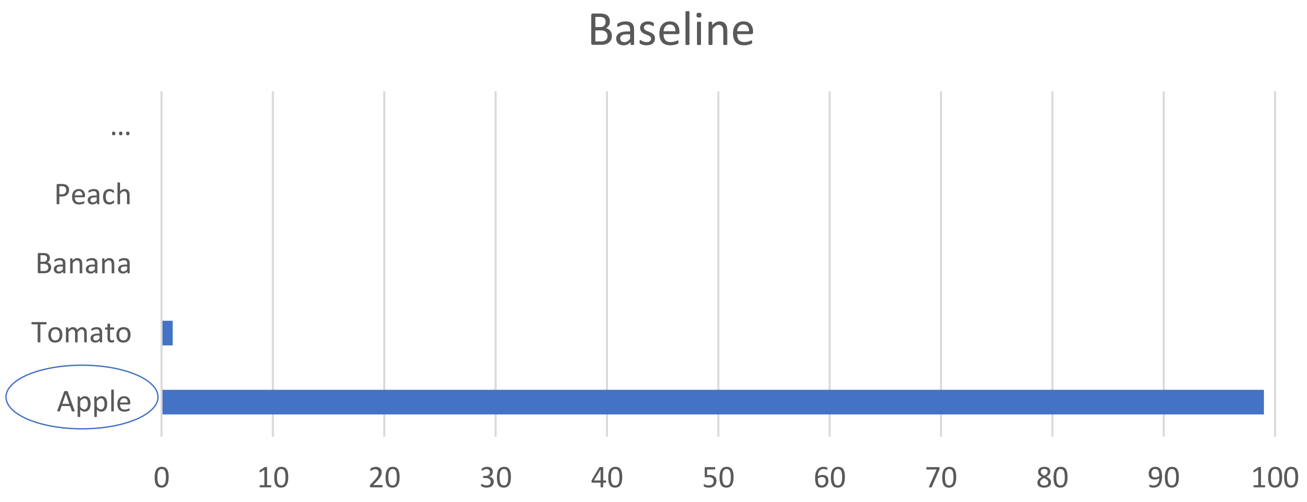
- 90380 images of 131 fruits and vegetables
  - Training set size: 67692 images
  - Test set size: 22688 images
- Selected the 10 most photographed fruits and vegetables:
  - Apple
  - Banana
  - Cherry
  - Grape
  - Onion
  - Peach
  - Pear
  - Pepper
  - Potato
  - Tomato

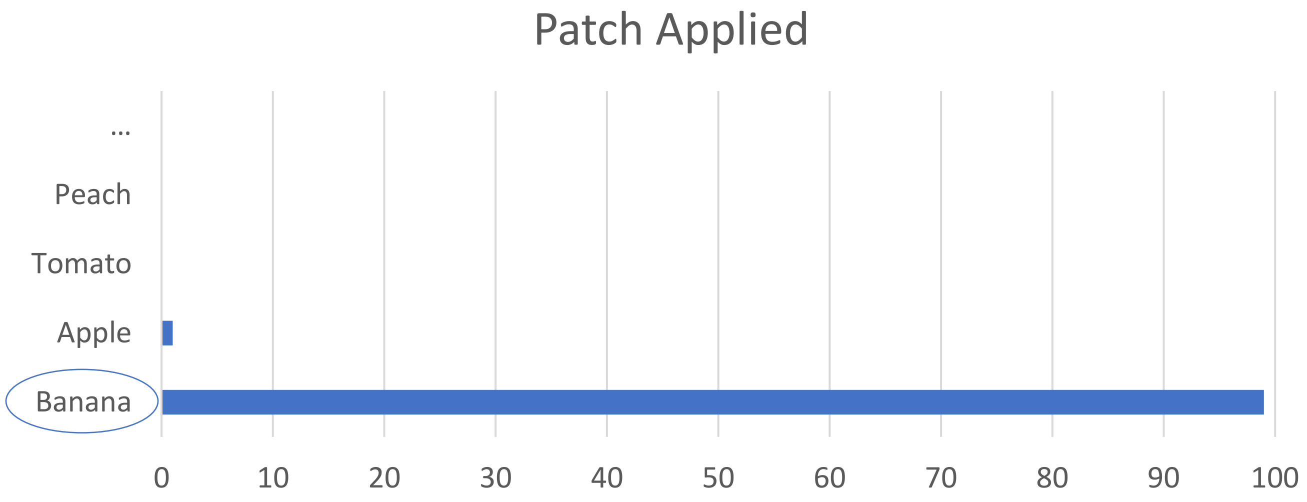
Model Comparison—Clean Data

Tested On:

	Vendor 1	Vendor 2
Accuracy	0.962	0.959
AUC	0.998	0.998
Precision	0.968	0.964
Recall	0.958	0.969

Patch Attack





Images From Fruits360 Dataset

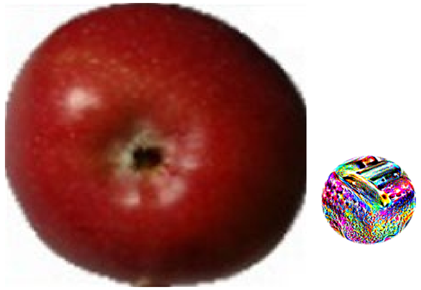
Model Comparison—Adversarial Data

Tested On:

	Vendor 1	Vendor 2
Accuracy	0.020	0.931
AUC	0.506	0.996
Precision	0.020	0.947
Recall	0.020	0.922

Vendor 2 Details

- Include images with adversarial patches in the training set
  - Model learns to ignore the patches



- NB: Adversarial training is not a panacea!
  - Same patches were used for training and testing

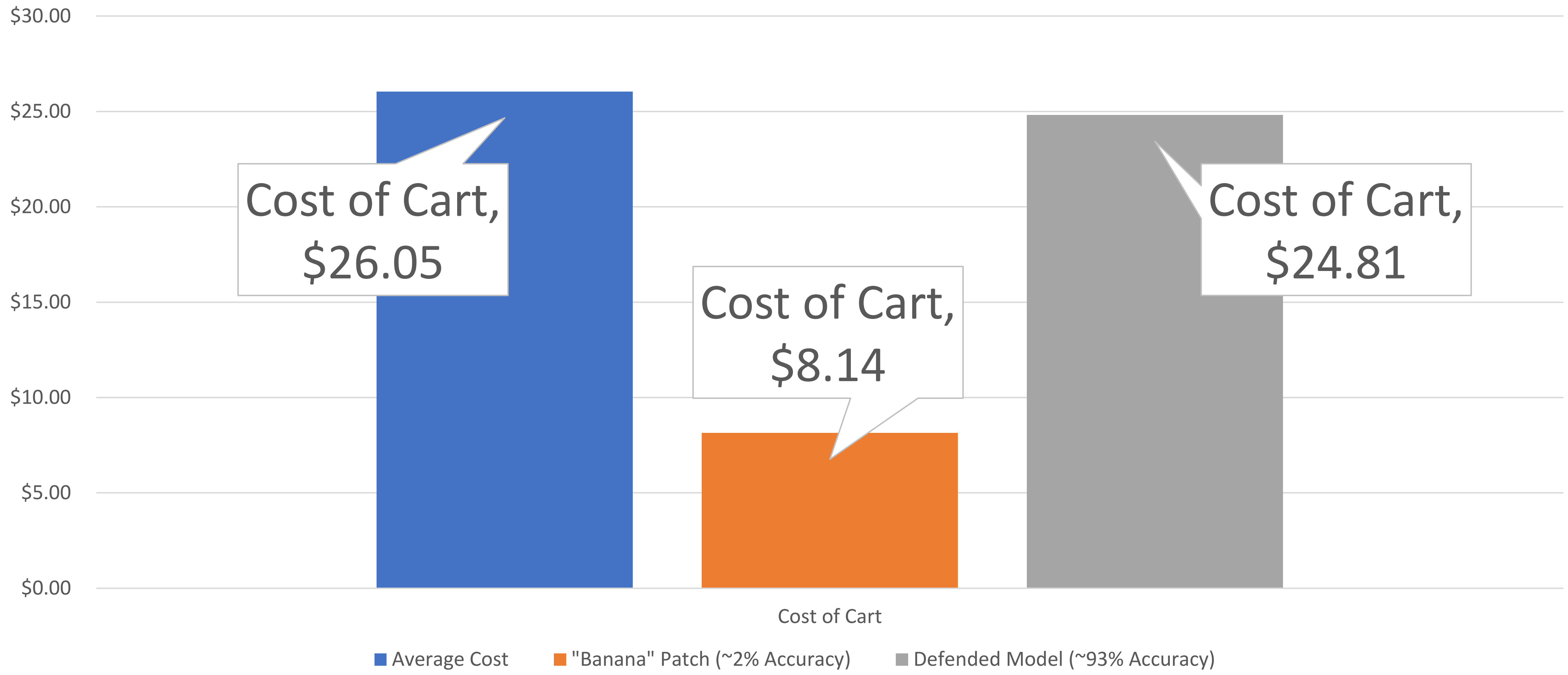
Images From Fruits360 Dataset

Purchase Scenario

At the grocery store you buy 5 each of:

Apples	$1.29 - $1.79 / each
Bananas	$0.29 - $0.49 / each
Peaches	$1.19 - $1.49 / each
Tomatoes	$1.59 - $2.29 / each

Cost of Cart



$30.00
$25.00
$20.00
$15.00
$10.00
$5.00
$0.00
Cost of Cart, $26.05
Cost of Cart, $8.14
Cost of Cart, $24.81
Cost of Cart
Average Cost
"Banana" Patch (~2% Accuracy)
Defended Model (~93% Accuracy)

Key Takeaways of Supermarket Scenario

- Focus on impactful operational outcomes
  - Monetary impacts will be key to determine if it’s worth buying/deploying
- Identify failure modes most important for your context
  - Which types of attacks account for the most risk?
- Include all aspects of the system (including humans!)
  - System-level vs component-level analysis
- System context can offer various mitigations
  - Technical: Integrate with a patch detection model
  - Non-technical: Train attendants to look for possible attacks

Integration and Regression Testing

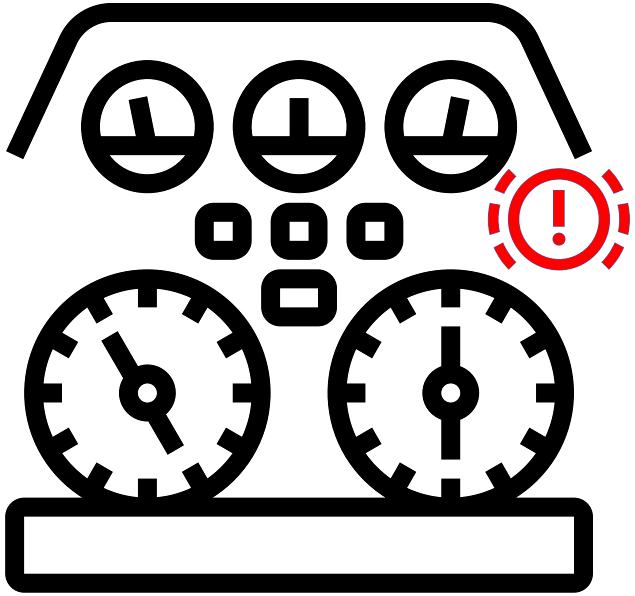
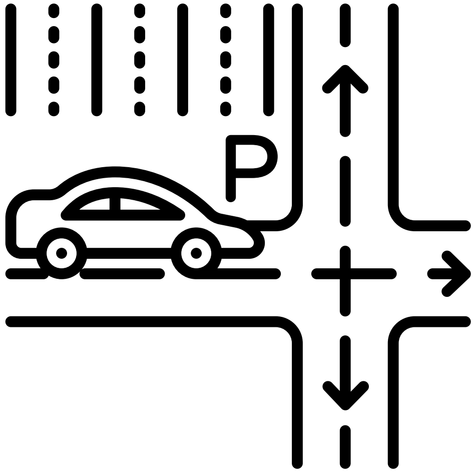
Risk Assessment Process

- Identify task
- Is AI necessary?
- Identify threat & deployment assumptions
- Which attacks/defenses are still relevant?
- Identify metrics applicable to highest priority risks
- Build experiments and synthesize results

Road Sign Detection and Classification

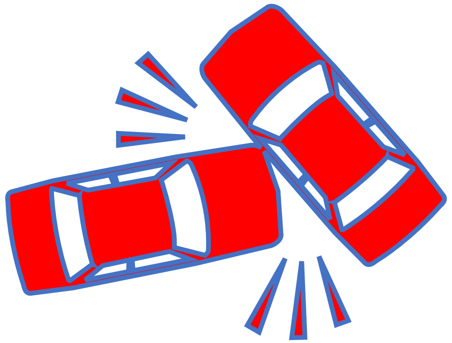
From Road Sign Detection Dataset

Road Sign example

Images From Road Sign Detection Dataset

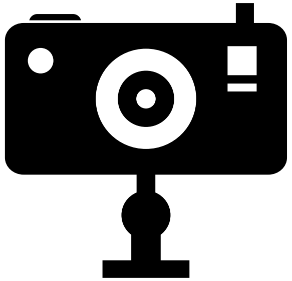
Threat/Deployment Model



Risk of accidents

Unmonitored environment

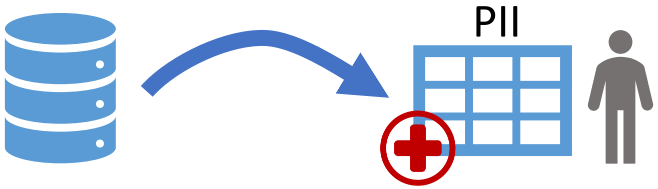
Primary or backup
human driver



Freshly generated
digital representations

Road Sign example: Attack Profile

Digital manipulation attacks are difficult in our deployment setting.



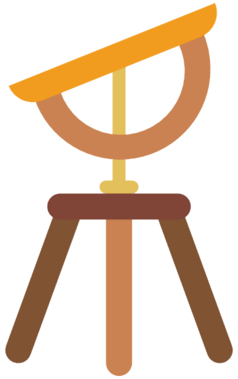
Training data is not sensitive.
Model extraction attacks may pose a concern.

Physical manipulation of environment could pose serious challenges.

Dataset poisoning could allow for physical triggers to cause blindness or misclassification.

Images From Road Sign Detection and MNIST Datasets

Road Sign Detection Combinations Explored

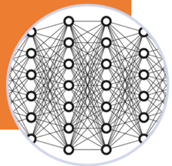


Dioptra

Image: Flaticon.com/Smashicons

- FasterRCNN
- RetinaNet
- MaskRCNN
- Modified YOLOv1

Training Architecture



- Flipping & brightness changes
- Backdoor Poisoning

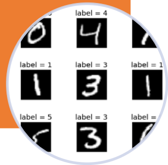
Data Augmentation

- JPEG Compression
- Spatial Smoothing

Inference pre-processing

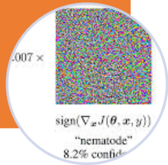
- Kaggle Road Sign Detection competition

Dataset



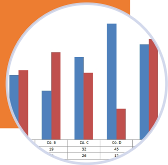
- Patch (classification)
- Robust DPatch (detection)

Attack on trained model



- Average Precision
- Intersection over Union
- Mean Average Precision

Metric



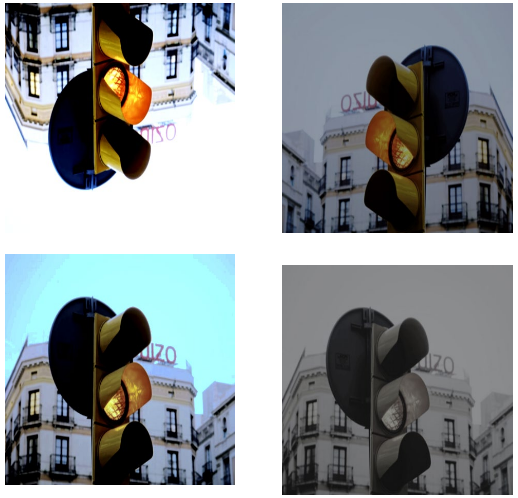
Dataset Considerations


Label	Training Set	Test Set	Training Set	Test Set
Traffic Light	151	38	79.89%	20.11%
Speed Limit	639	154	80.58%	19.42%
Crosswalk	168	43	79.62%	20.38%
Stop Sign	80	22	78.43%	21.57%
# Images	698	179	79.59%	20.41%

Tracks of images

Original

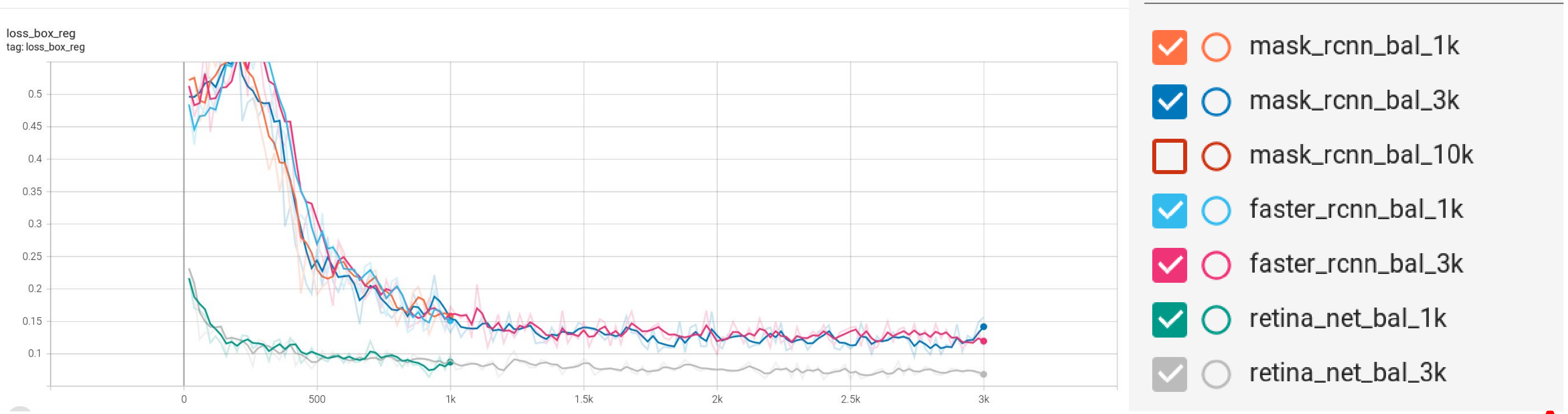
Augmented



Images From Road Sign Detection Dataset

Performance Metrics

Trained on 3 detection models in Detectron2 model zoo:



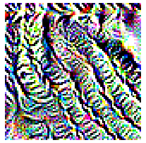
Best Model : RetinaNet w/ 3K Training Steps

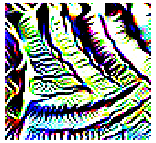
AP	AP-50	AP-75	AP-s	AP-m	AP-l
68.356	86.693	75.122	50.79	85.138	79.596

mAP(IoU=50:95)

Label	mAP(IoU=50:95) Average Precision
Stop Sign	76.818
Crosswalk	62.346
Speed Limit	84.404
Traffic Light	49.855

Robust DPatch Attack





Robust DPatch evasion attack
on object detection

Patch position, size, brightness, etc.

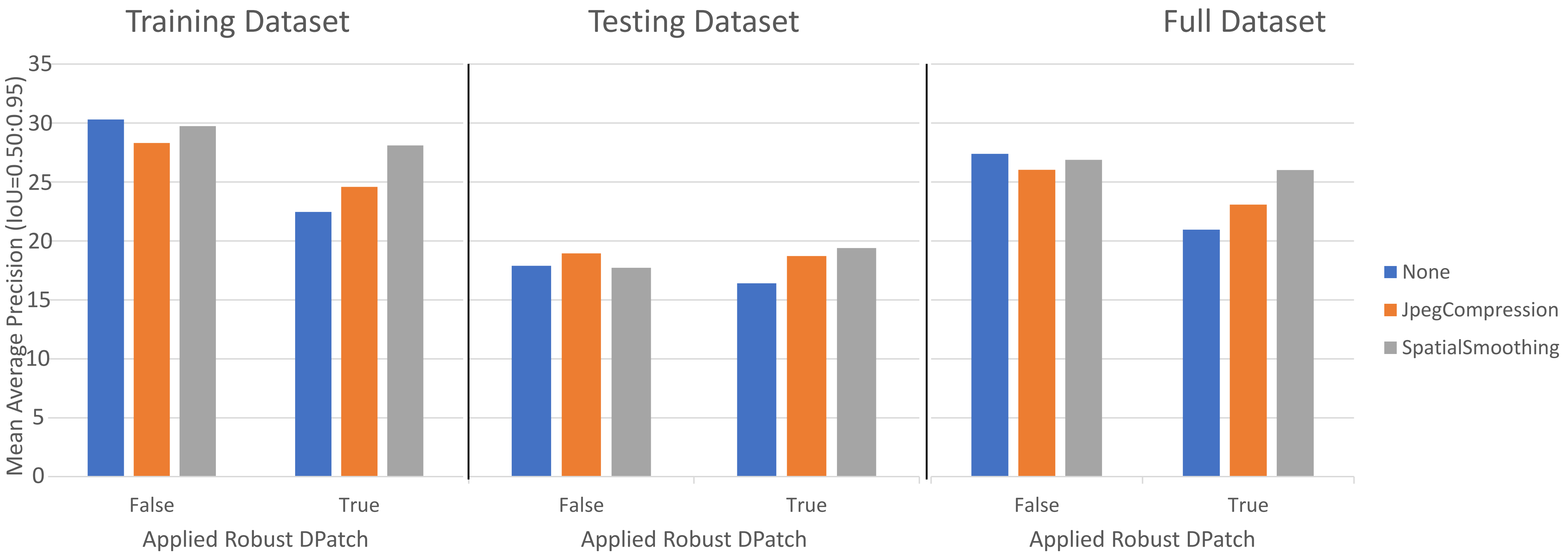
Original

JPEG Compression

Spatial Smoothing

Images From Road Sign Detection Dataset

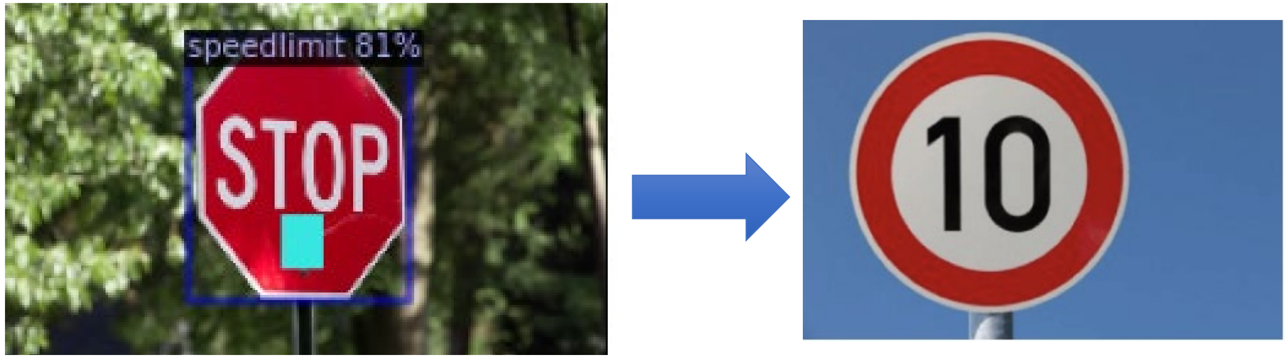
Patch Attack Results—Modified YOLOv1



- Patch attack reduces mean-average precision relative to baseline
- Spatial smoothing defense was effective in mitigating the patch attack

Backdoor Poisoning Attack

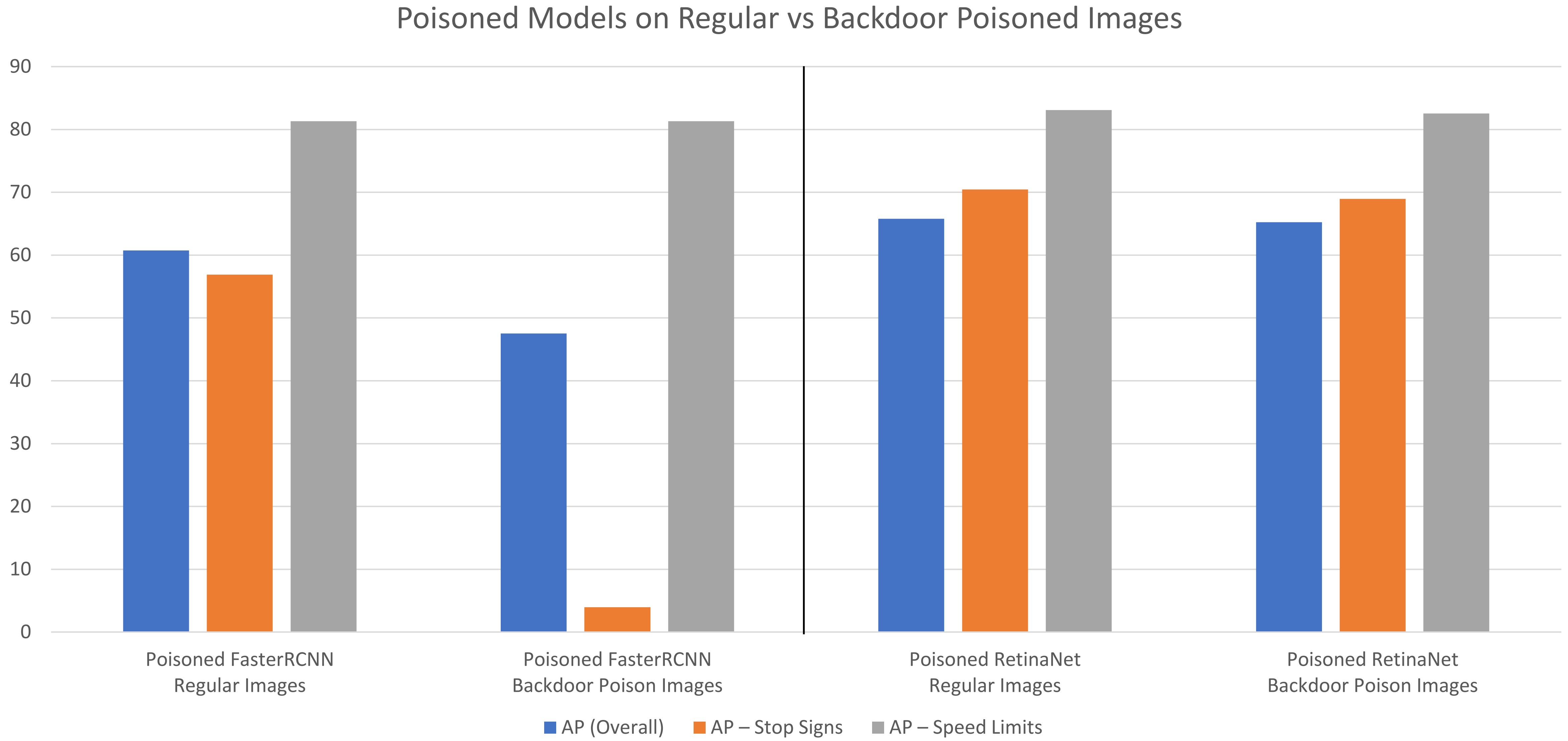
Backdoor poisoning attack
with teal square trigger



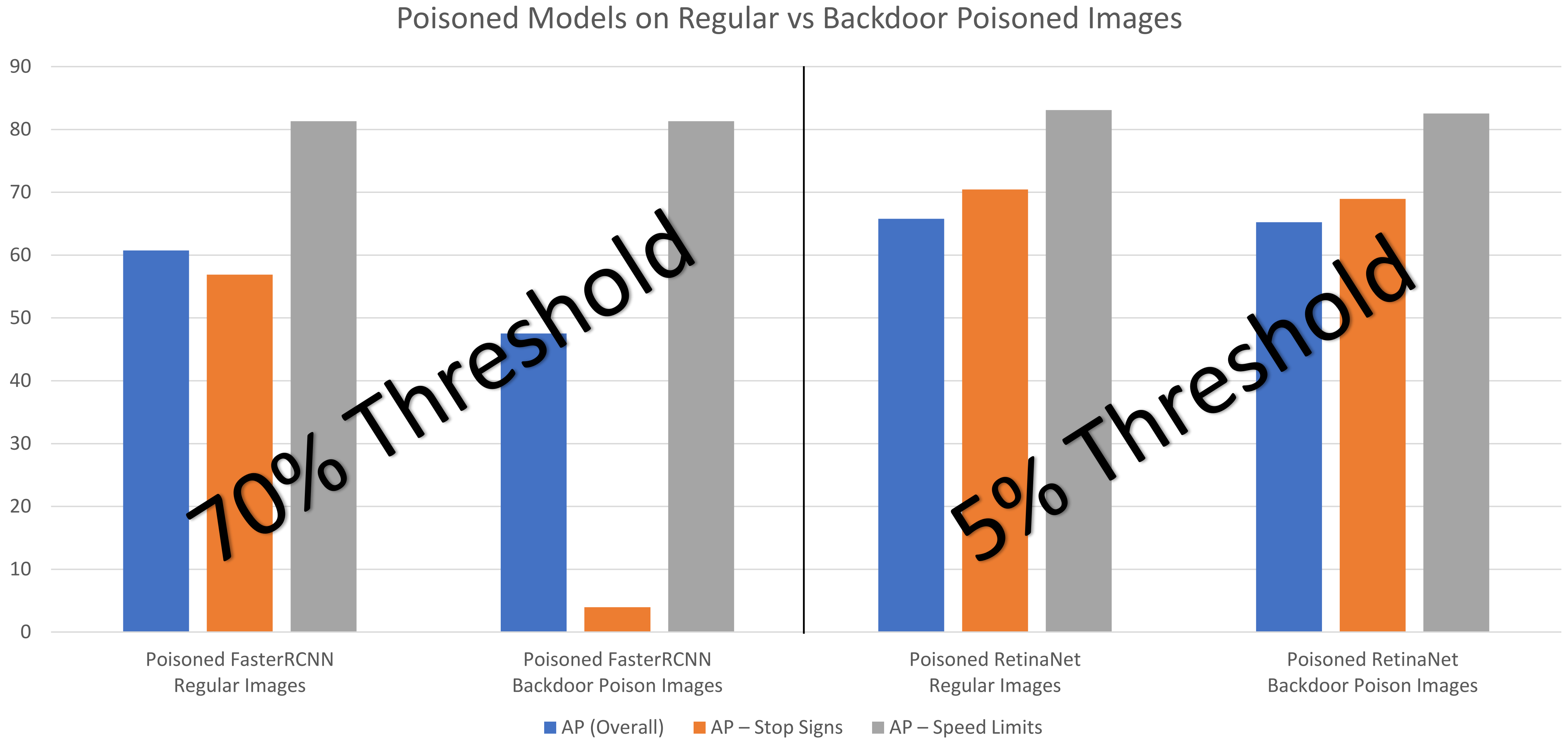
Poisoned Stop Sign labeled as Speed Limit Sign

Images From Road Sign Detection Dataset

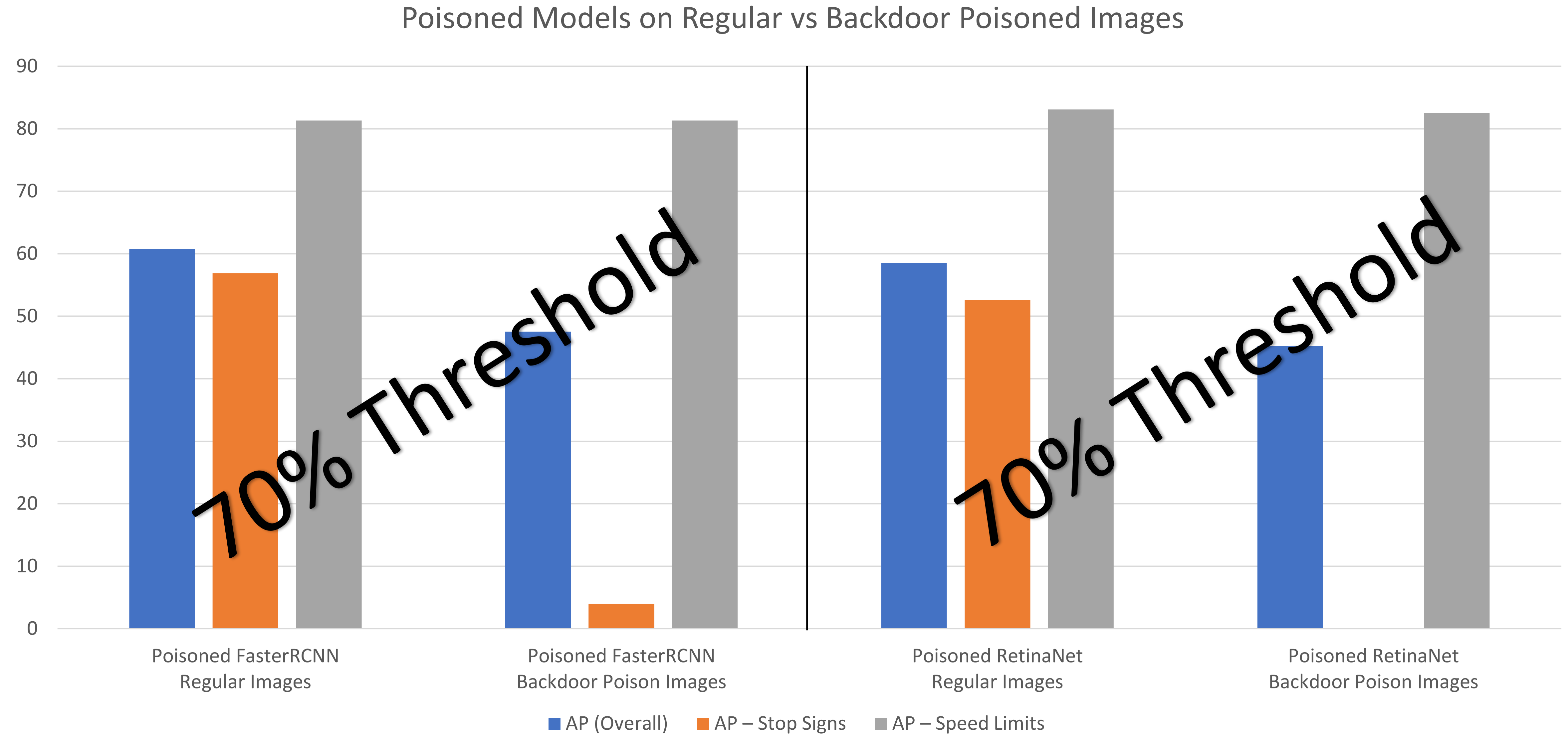
Poisoning Results



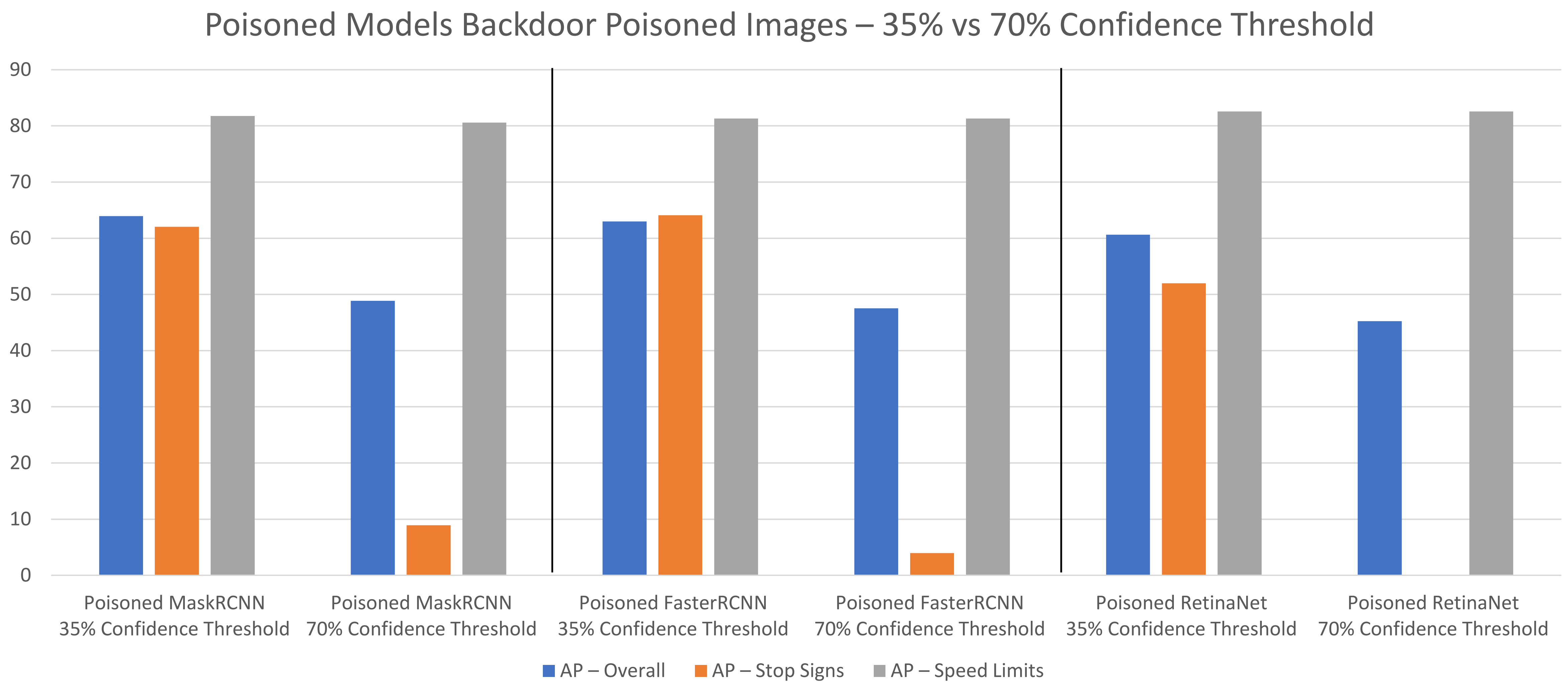
Poisoning Results



Poisoning Results



Poisoning Results



Key Takeaways from Road Sign Detection

- Understand what your metrics are really telling you
  - Understand how they work
  - Breaking them up into smaller ones
- Understand your tools' default parameters
  - Important for fair comparison
- Be aware of how dataset characteristics affect metrics
  - Class imbalances
  - Dataset Artifacts (e.g., "Tracks" of images)
- Testing during development can help improve final product
  - More and better data and metrics
  - Identify external mitigations such as supply chain protections

Conclusions

- There are plenty of things to measure
  - There can be no small set of metrics good for all uses
- It's valuable to develop processes for deciding what to measure
  - Focus on risks when deciding how to test and evaluate AI-enabled systems
- Tools to help manage evaluation are indispensable
  - Customizable automation can help manage the complexity

Future Work

- Continued Evolution of Dioptra
  - Attacks, Defenses, Metrics, Modalities
- Expanding Community of Use

Questions?

https://github.com/usnistgov/dioptra

dioptra@nist.gov

References

- G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, “A study of the effect of JPG compression on adversarial images,” arXiv, arXiv:1608.00853, Aug. 2016. doi: 10.48550/arXiv.1608.00853.
- T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, “Adversarial Patch,” arXiv, arXiv:1712.09665, May 2018. doi: 10.48550/arXiv.1712.09665.
- S. Kotyan and D. V. Vargas, “Adversarial Robustness Assessment: Why both L_0 and L_∞ Attacks Are Necessary.” arXiv, Jul. 16, 2020. Accessed: May 20, 2022. [Online]. Available: http://arxiv.org/abs/1906.06026
- Adversarial Robustness Toolbox (ART) v1.10. Trusted-AI, 2022. Accessed: May 17, 2022. [Online]. Available: https://github.com/Trusted-AI/adversarial-robustness-toolbox.
- C. Liao, H. Zhong, A. Squicciarini, S. Zhu, and D. Miller, “Backdoor Embedding in Convolutional Neural Network Models via Invisible Perturbation.” arXiv, Aug. 30, 2018. Accessed: May 20, 2022. [Online]. Available: http://arxiv.org/abs/1808.10307
- T. Gu, B. Dolan-Gavitt, and S. Garg, “BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain.” arXiv, Mar. 11, 2019. Accessed: May 20, 2022. [Online]. Available: http://arxiv.org/abs/1708.06733
- T. J. L. Tan and R. Shokri, “Bypassing Backdoor Detection Algorithms in Deep Learning.” arXiv, Jun. 06, 2020. Accessed: May 20, 2022. [Online]. Available: http://arxiv.org/abs/1905.13409
- Y. Sharma and P.-Y. Chen, “Bypassing Feature Squeezing by Increasing Adversary Strength,” arXiv, arXiv:1803.09868, Mar. 2018. doi: 10.48550/arXiv.1803.09868.
- A. Turner, D. Tsipras, and A. Mądry, “Clean-Label Backdoor Attacks,” p. 21.
- K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition.” arXiv, Dec. 10, 2015. Accessed: May 20, 2022. [Online]. Available: http://arxiv.org/abs/1512.03385
- N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks.” arXiv, Mar. 14, 2016. Accessed: May 20, 2022. [Online]. Available: http://arxiv.org/abs/1511.04508
- X. Liu, H. Yang, Z. Liu, L. Song, H. Li, and Y. Chen, “DPatch: An Adversarial Patch Attack on Object Detectors,” arXiv, arXiv:1806.02299, Apr. 2019. doi: 10.48550/arXiv.1806.02299.
- T. Strauss, M. Hanselmann, A. Junginger, and H. Ulmer, “Ensemble Methods as a Defense to Adversarial Perturbations Against Deep Neural Networks,” arXiv, arXiv:1709.03423, Feb. 2018. doi: 10.48550/arXiv.1709.03423.
- I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and Harnessing Adversarial Examples,” arXiv, arXiv:1412.6572, Mar. 2015. doi: 10.48550/arXiv.1412.6572.
- W. Xu, D. Evans, and Y. Qi, “Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks,” 2018. doi: 10.14722/ndss.2018.23198.
- A. Saha, A. Subramanya, and H. Pirsiavash, “Hidden Trigger Backdoor Attacks.” arXiv, Dec. 20, 2019. Accessed: May 20, 2022. [Online]. Available: http://arxiv.org/abs/1910.00033

References

- A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems*, 2012, vol. 25. Accessed: May 20, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html
- C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, “Label-Only Membership Inference Attacks.” arXiv, Dec. 05, 2021. Accessed: May 20, 2022. [Online]. Available: http://arxiv.org/abs/2007.14321
- R. Shokri, M. Stronati, C. Song, and V. Shmatikov, “Membership Inference Attacks against Machine Learning Models.” arXiv, Mar. 31, 2017. Accessed: May 20, 2022. [Online]. Available: http://arxiv.org/abs/1610.05820
- Z. Li and Y. Zhang, “Membership Leakage in Label-Only Exposures.” arXiv, Sep. 17, 2021. Accessed: May 20, 2022. [Online]. Available: http://arxiv.org/abs/2007.15528
- “MLflow - A platform for the machine learning lifecycle,” *MLflow*. https://mlflow.org/ (accessed May 17, 2022).
- A. Athalye, N. Carlini, and D. Wagner, “Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples,” arXiv, arXiv:1802.00420, Jul. 2018. doi: 10.48550/arXiv.1802.00420.
- M. Lee and Z. Kolter, “On Physical Adversarial Patches for Object Detection,” arXiv, arXiv:1906.11897, Jun. 2019. doi: 10.48550/arXiv.1906.11897.
- N. Papernot and P. McDaniel, “On the Effectiveness of Defensive Distillation,” arXiv, arXiv:1607.05113, Jul. 2016. doi: 10.48550/arXiv.1607.05113.
- J. Su, D. V. Vargas, and S. Kouichi, “One pixel attack for fooling deep neural networks,” *IEEE Trans. Evol. Computat.*, vol. 23, no. 5, pp. 828–841, Oct. 2019, doi: 10.1109/TEVC.2019.2890858.
- A. Shafahi *et al.*, “Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks.” arXiv, Nov. 10, 2018. Accessed: May 20, 2022. [Online]. Available: http://arxiv.org/abs/1804.00792
- X. Liu *et al.*, “Privacy and Security Issues in Deep Learning: A Survey,” *IEEE Access*, vol. 9, pp. 4566–4593, 2021, doi: 10.1109/ACCESS.2020.3045078.
- “Project Jupyter.” https://jupyter.org (accessed May 17, 2022).
- “Road Sign Detection | Kaggle.” https://www.kaggle.com/datasets/andrewmvd/road-sign-detection (accessed May 20, 2022).
- X. Hu, D. Wu, H. Li, F. Jiang, and H. Lu, “ShallowNet: An Efficient Lightweight Text Detection Network Based on Instance Count-Aware Supervision Information,” in *Neural Information Processing*, Cham, 2021, pp. 633–644. doi: 10.1007/978-3-030-92185-9_52.
- “Stanford Cars Dataset.” https://www.kaggle.com/jessicali9530/stanford-cars-dataset (accessed May 20, 2022).
- K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition.” arXiv, Apr. 10, 2015. Accessed: May 20, 2022. [Online]. Available: http://arxiv.org/abs/1409.1556
- “What is Dioptra? — Dioptra 0.0.0 documentation.” https://pages.nist.gov/dioptra/ (accessed May 20, 2022).
- J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection.” arXiv, May 09, 2016. Accessed: May 20, 2022. [Online]. Available: http://arxiv.org/abs/1506.02640