Exploring Ways To Counter Adversarial Attacks Against Image Classifiers

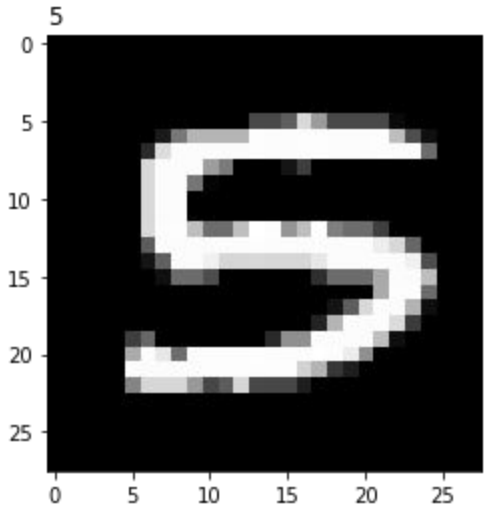
By: Emy Parparita

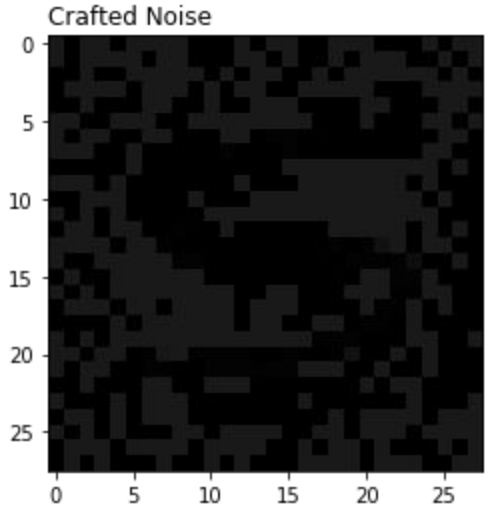
Description Of The Problem

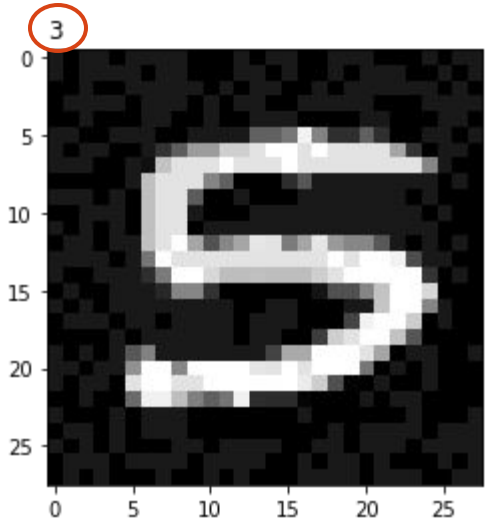
Adversarial attacks against image classifiers use small perturbations applied to input images to cause a misclassification.

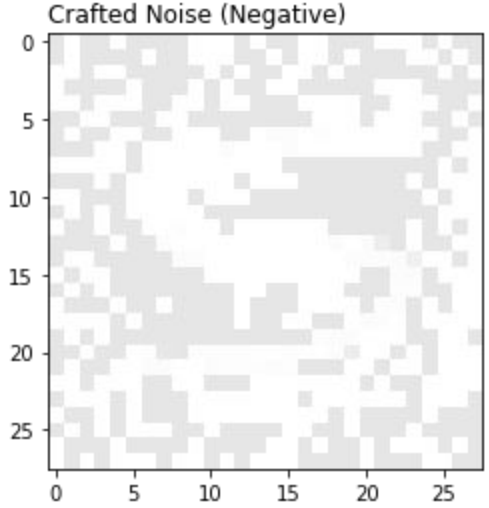
The perturbations are hard to detect by the human eye because they are artificially constructed by adding the smallest amount of noise to the original input along an optimal path that would cross a decision boundary.

Robustness to malicious attacks or random variations in input for that matter is crucial for image classifiers deployed in mission critical systems (e.g. self-driving cars).



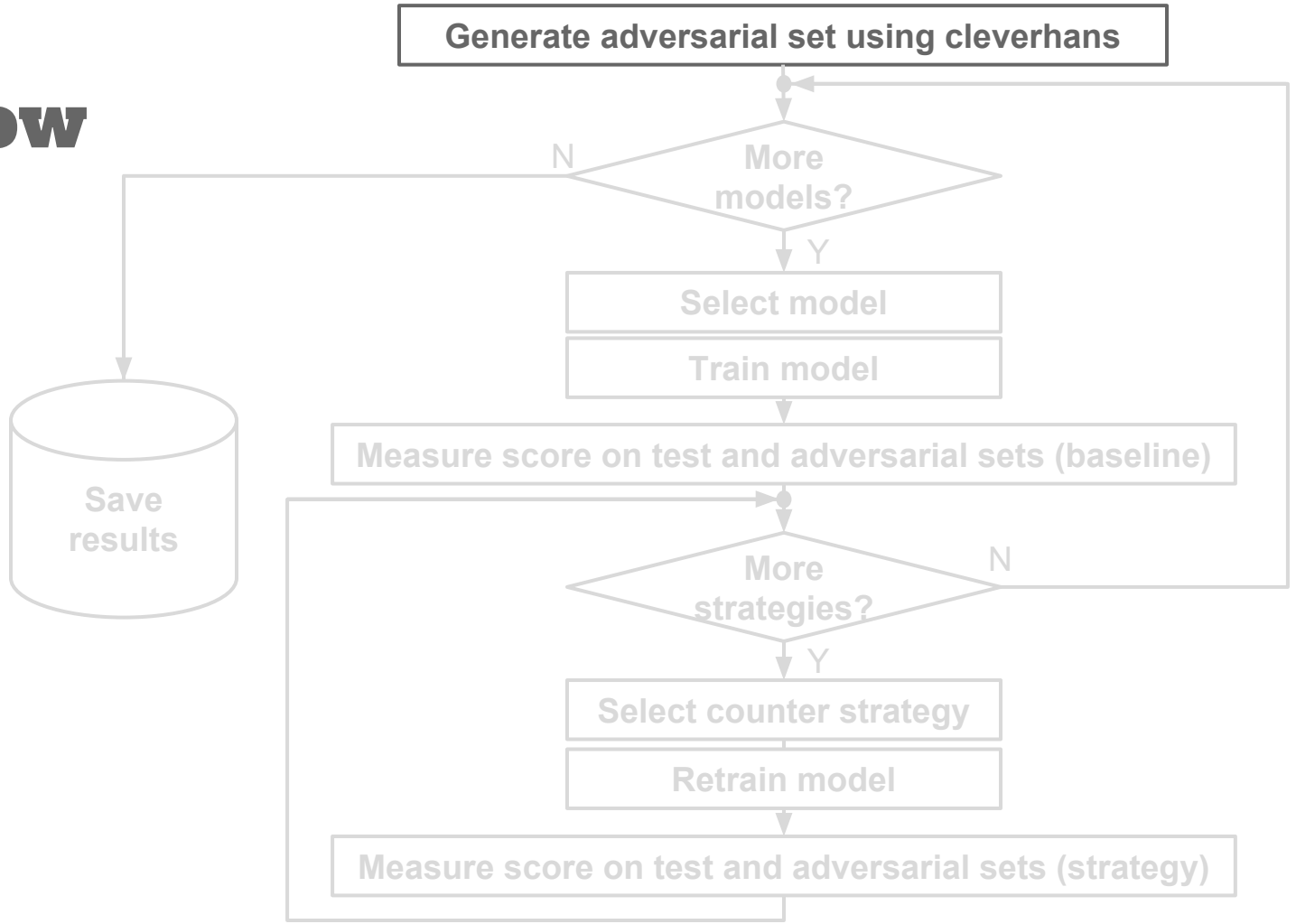




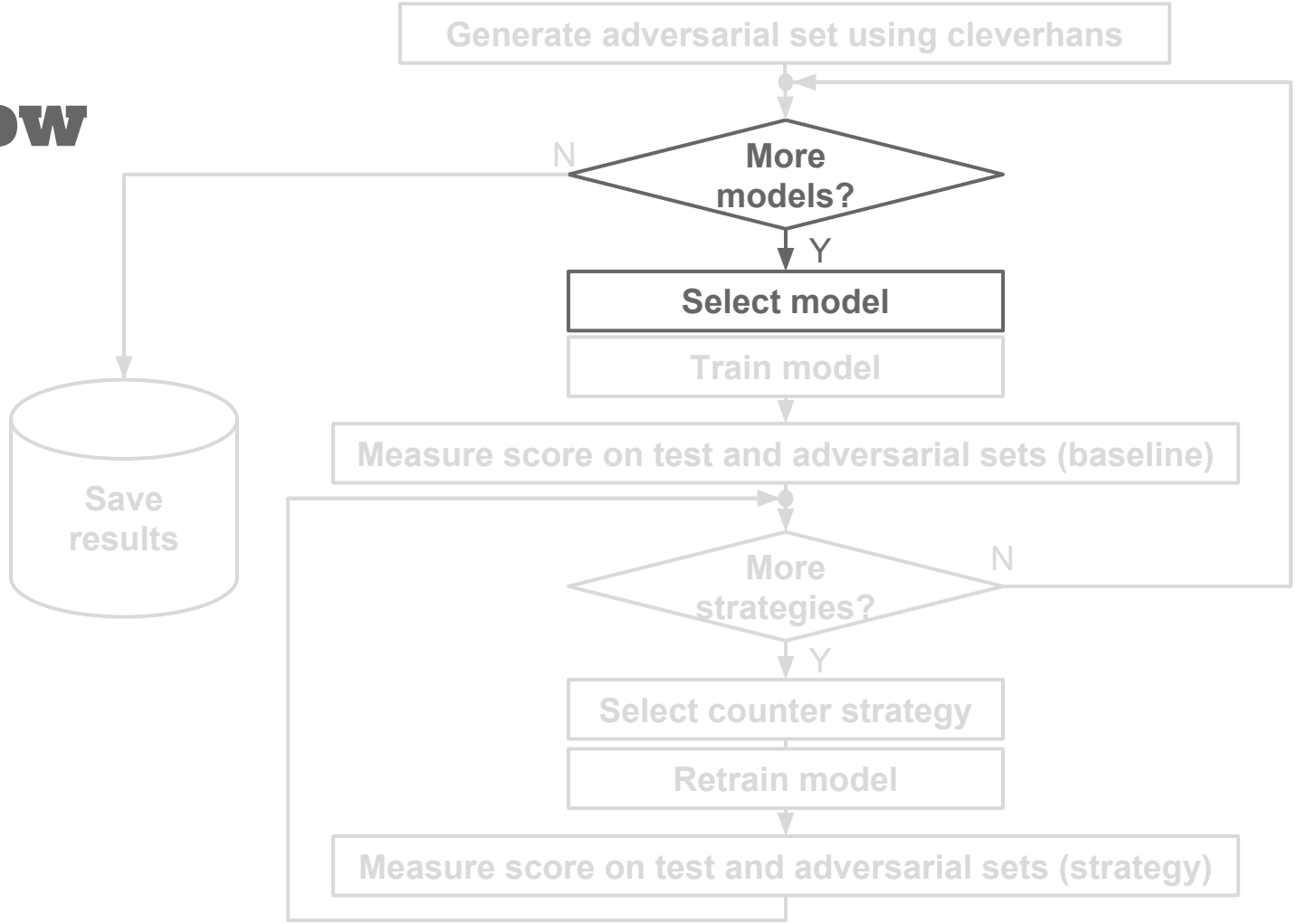


Attack Example

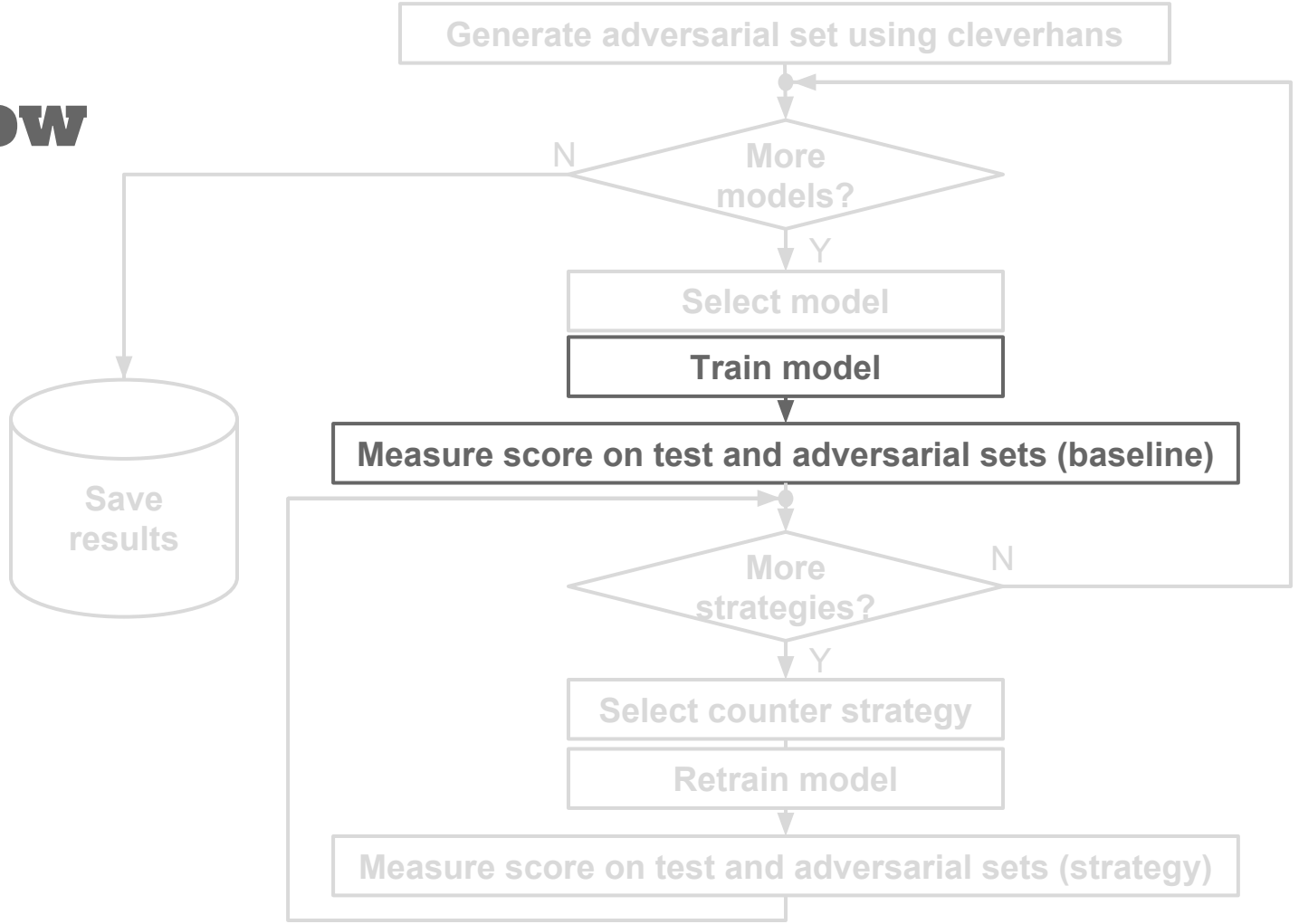
Workflow


Generate adversarial set using cleverhans
More models?
N
Y
Select model
Train model
Measure score on test and adversarial sets (baseline)
Save results
More strategies?
N
Y
Select counter strategy
Retrain model
Measure score on test and adversarial sets (strategy)

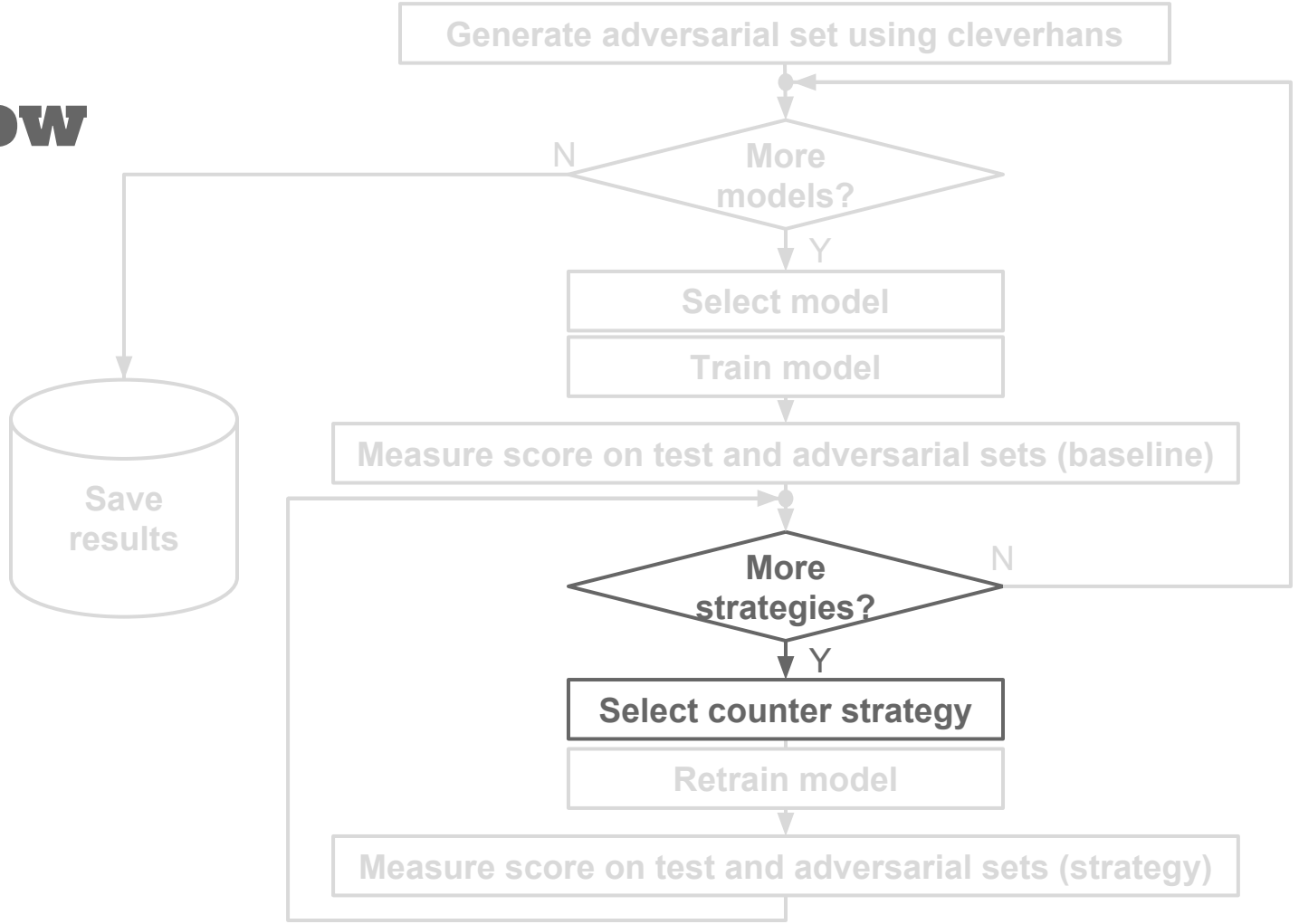
Workflow


Generate adversarial set using cleverhans
More models?
N
Y
Select model
Train model
Measure score on test and adversarial sets (baseline)
Save results
More strategies?
N
Y
Select counter strategy
Retrain model
Measure score on test and adversarial sets (strategy)

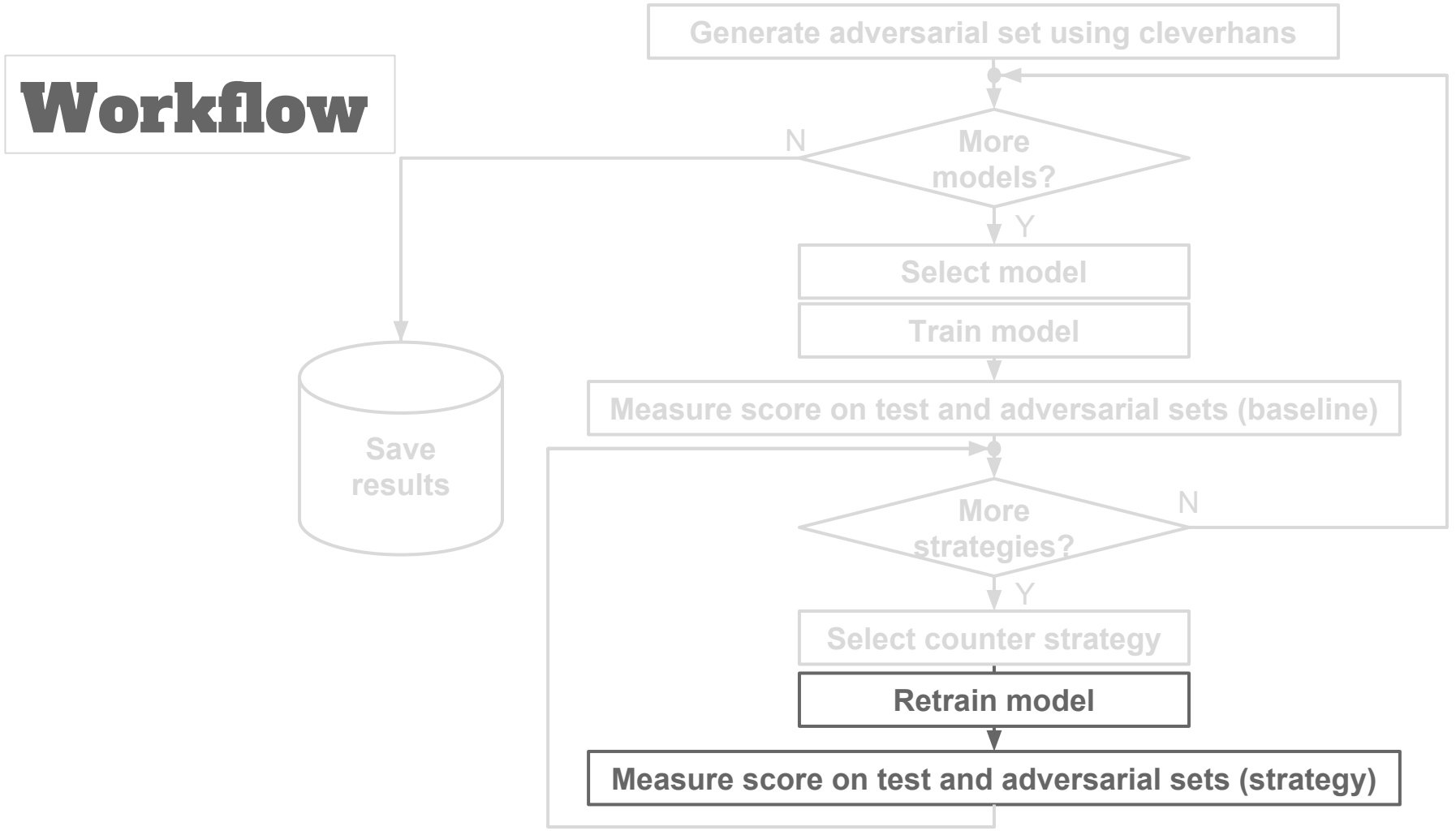
Workflow


Generate adversarial set using cleverhans
More models?
N
Y
Select model
Train model
Measure score on test and adversarial sets (baseline)
Save results
More strategies?
N
Y
Select counter strategy
Retrain model
Measure score on test and adversarial sets (strategy)

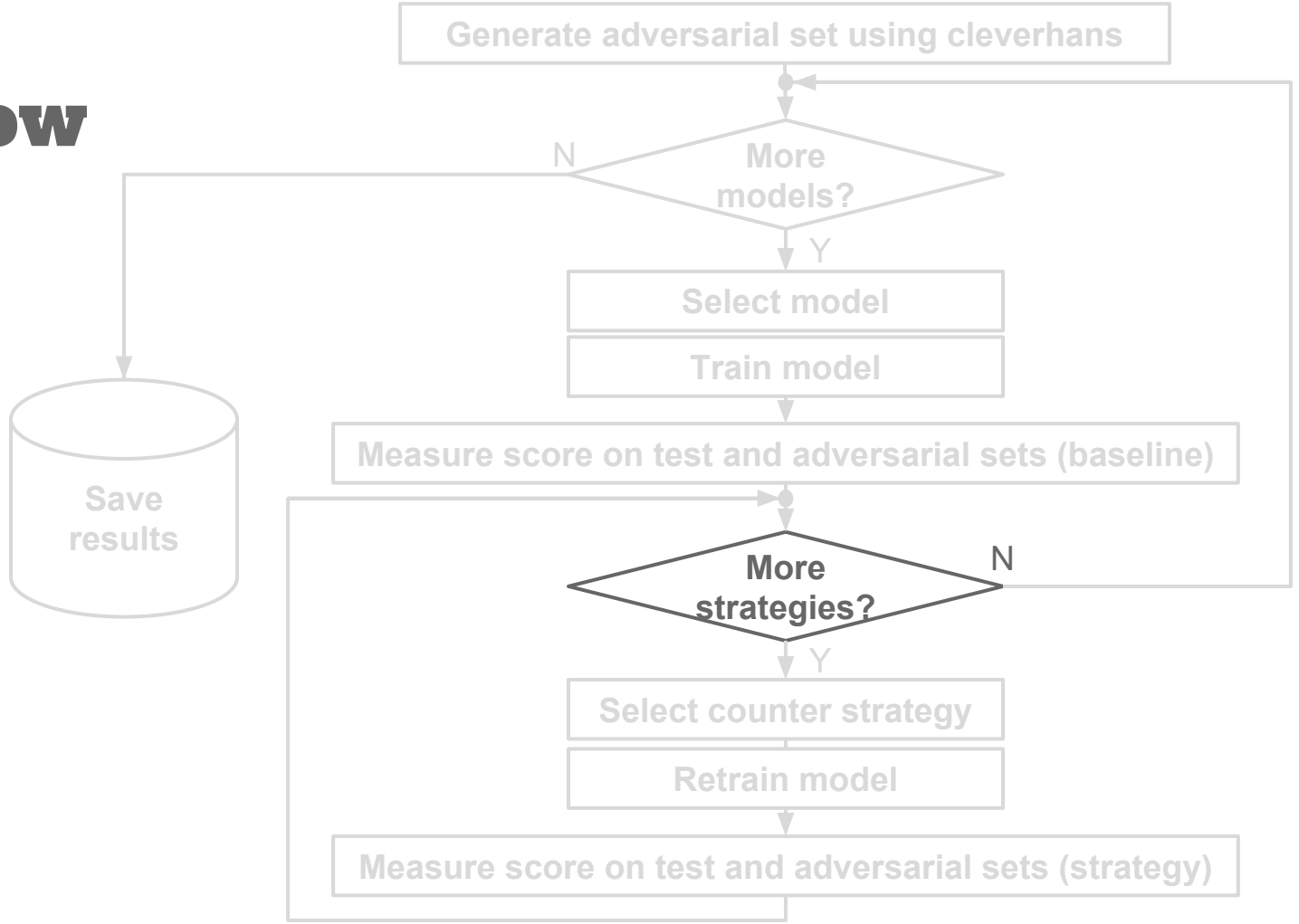
Workflow


Generate adversarial set using cleverhans
More models?
N
Y
Select model
Train model
Measure score on test and adversarial sets (baseline)
Save results
More strategies?
N
Y
Select counter strategy
Retrain model
Measure score on test and adversarial sets (strategy)

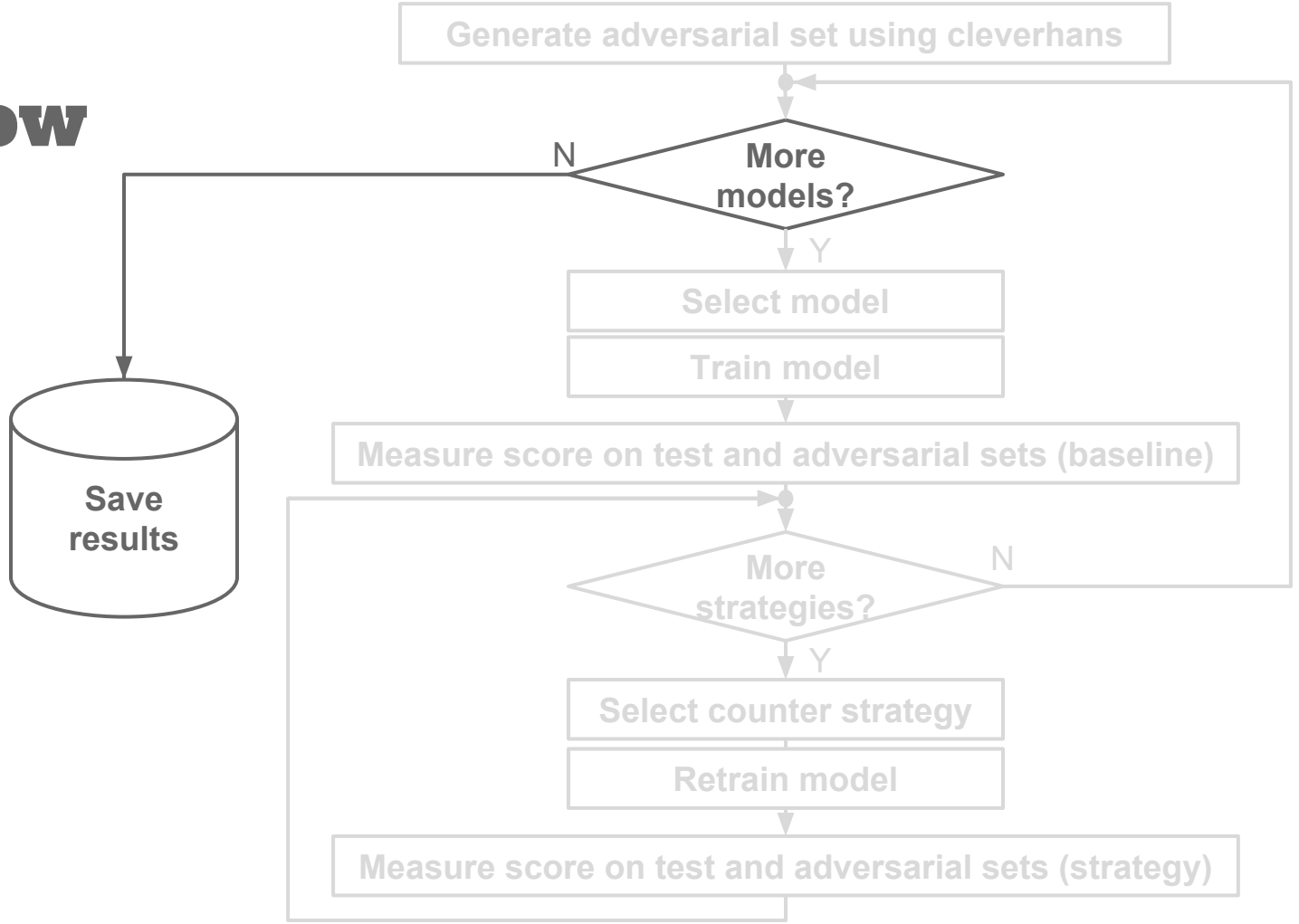
Workflow


Generate adversarial set using cleverhans
More models?
N
Y
Select model
Train model
Measure score on test and adversarial sets (baseline)
Save results
More strategies?
N
Y
Select counter strategy
Retrain model
Measure score on test and adversarial sets (strategy)

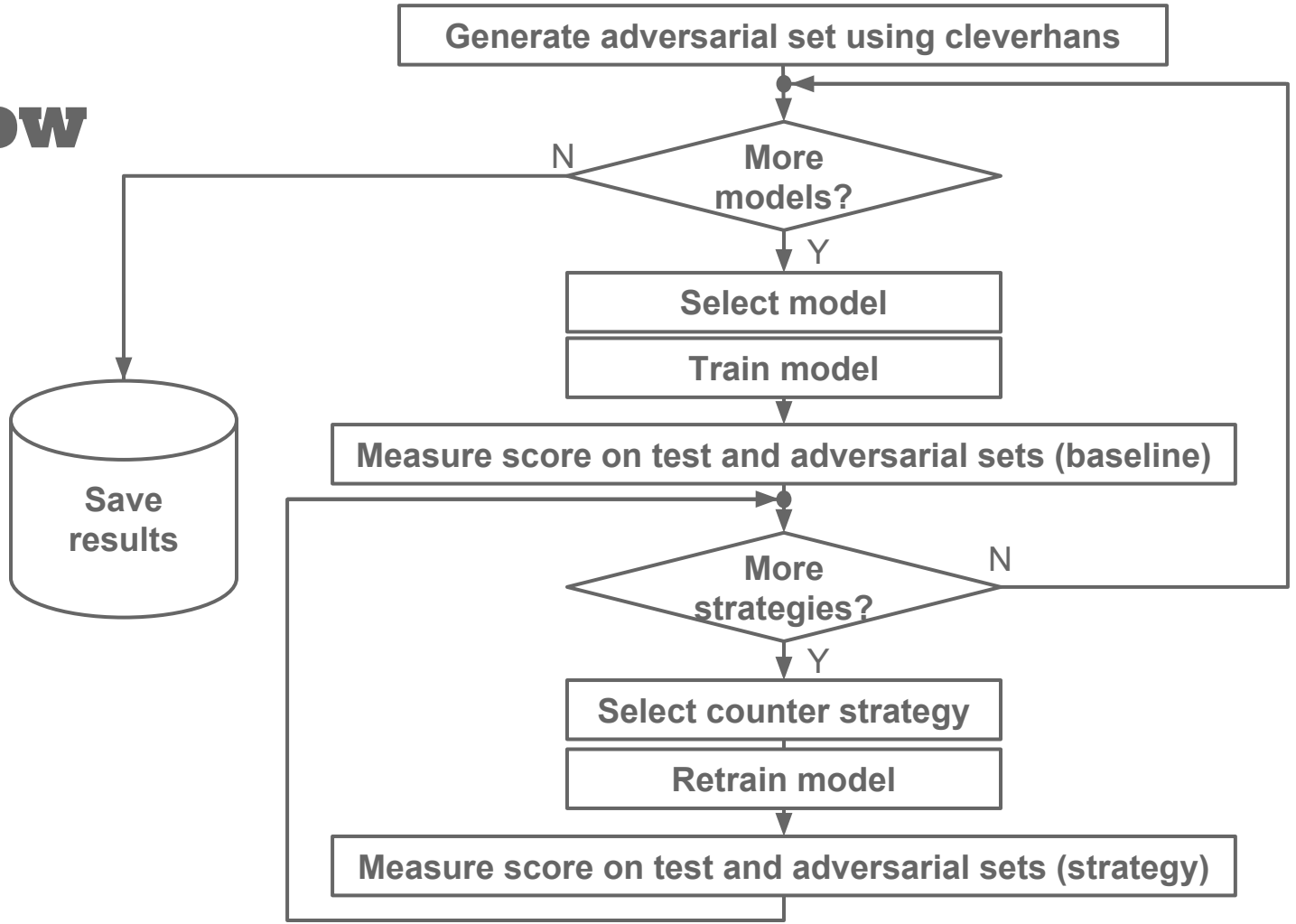
Workflow


Generate adversarial set using cleverhans
More models?
N
Y
Select model
Train model
Measure score on test and adversarial sets (baseline)
Save results
More strategies?
N
Y
Select counter strategy
Retrain model
Measure score on test and adversarial sets (strategy)

Workflow


Generate adversarial set using cleverhans
More models?
N
Y
Select model
Train model
Measure score on test and adversarial sets (baseline)
More strategies?
N
Y
Select counter strategy
Retrain model
Measure score on test and adversarial sets (strategy)
Save results

Workflow

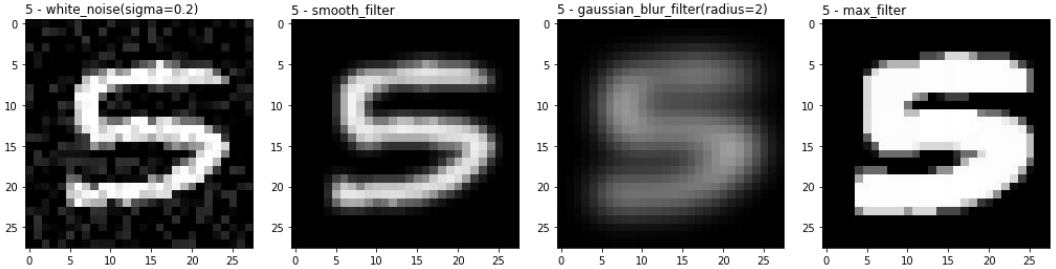

Generate adversarial set using cleverhans
More models?
N
Y
Select model
Train model
Measure score on test and adversarial sets (baseline)
Save results
More strategies?
N
Y
Select counter strategy
Retrain model
Measure score on test and adversarial sets (strategy)

Data Set And Classifiers

- Data set: MNIST 28x28 grayscale digits
- Classifiers:
  - KNN
  - SVM
  - Logistic Regression
  - CNN

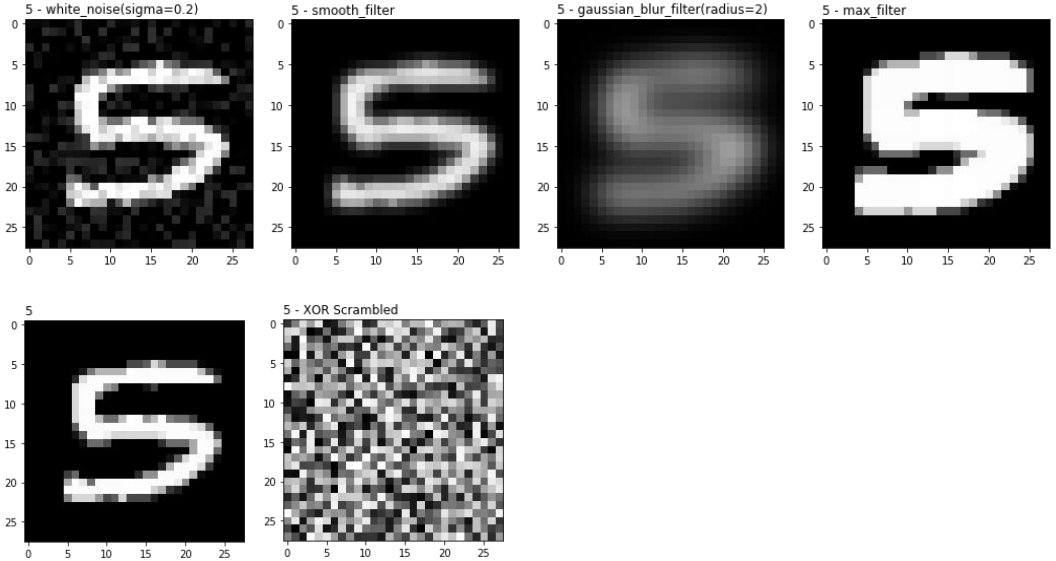
Defense Strategies

- Noise/filter injection during the train and/or test phases



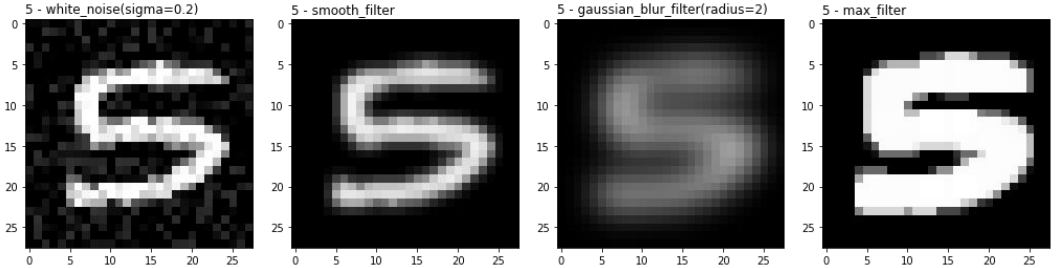
Defense Strategies

- Noise/filter injection during the train and/or test phases
- Input scrambling using a XOR’ed pseudo-random sequence

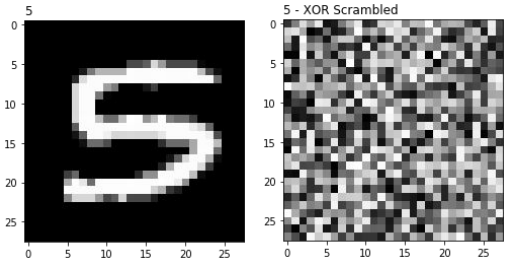


Defense Strategies

- Noise/filter injection during the train and/or test phases

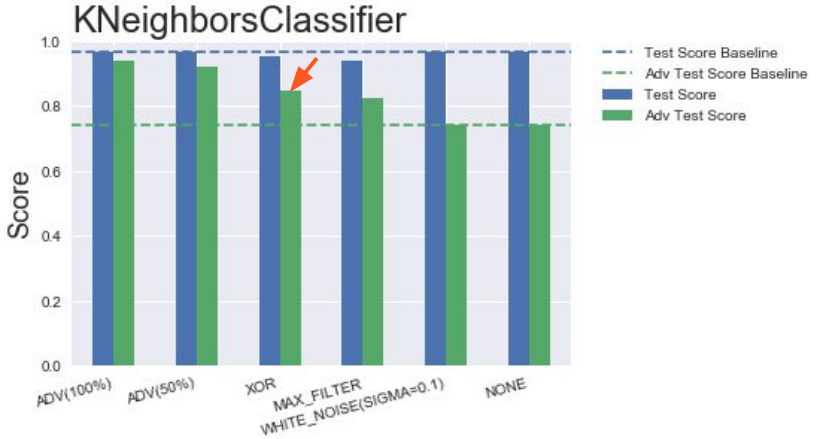


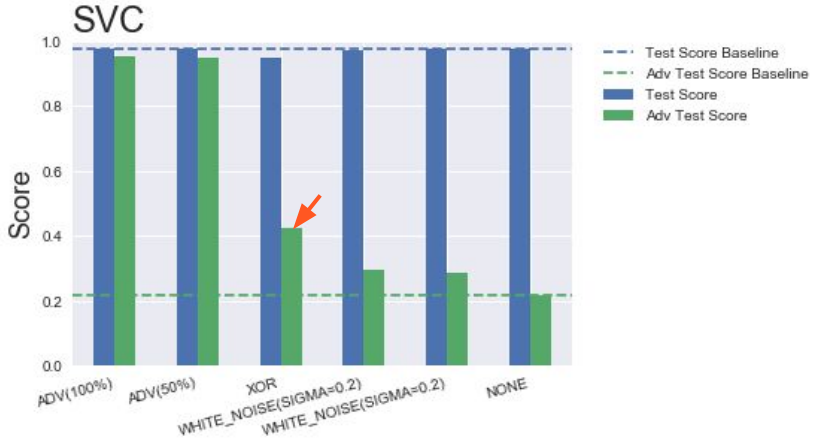
- Input scrambling using a XOR’ed pseudo-random sequence

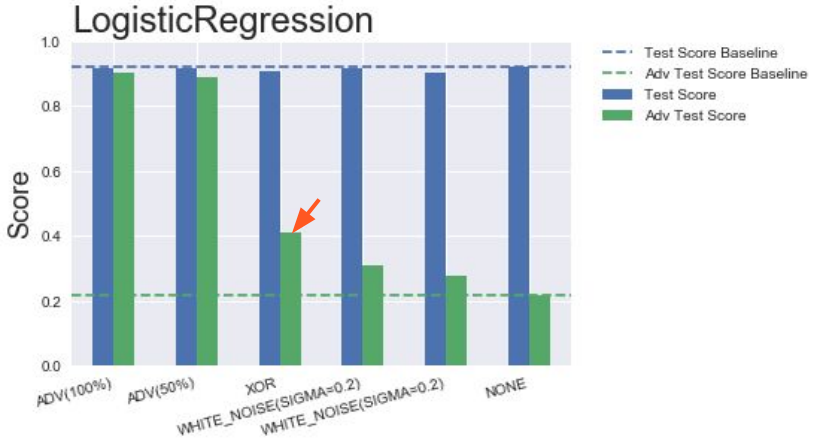


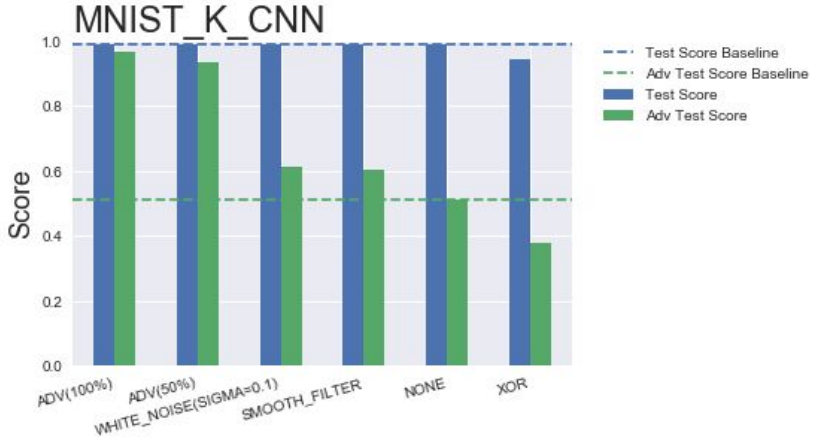
- The addition of a percentage of adversarial generated data to the training set

Results


KNeighborsClassifier
Score
1.0
0.8
0.6
0.4
0.2
0.0
ADV(100%)
ADV(50%)
XOR
MAX_FILTER
WHITE_NOISE(SIGMA=0.1)
NONE
Test Score Baseline
Adv Test Score Baseline
Test Score
Adv Test Score


SVC
Score
1.0
0.8
0.6
0.4
0.2
0.0
ADV(100%)
ADV(50%)
XOR
WHITE_NOISE(SIGMA=0.2)
WHITE_NOISE(SIGMA=0.2)
NONE
Test Score Baseline
Adv Test Score Baseline
Test Score
Adv Test Score


LogisticRegression
Score
1.0
0.8
0.6
0.4
0.2
0.0
ADV(100%)
ADV(50%)
XOR
WHITE_NOISE(SIGMA=0.2)
WHITE_NOISE(SIGMA=0.2)
NONE
Test Score Baseline
Adv Test Score Baseline
Test Score
Adv Test Score


MNIST_K_CNN
Score
1.0
0.8
0.6
0.4
0.2
0.0
ADV(100%)
ADV(50%)
WHITE_NOISE(SIGMA=0.1)
SMOOTH_FILTER
NONE
XOR
Test Score Baseline
Adv Test Score Baseline
Test Score
Adv Test Score

Conclusions

- The noise strategy doesn’t help

Conclusions

- The noise strategy doesn’t help
- The best results are achieved by adding the adversarial input to the training set. However this is attuned to a specific type of attack

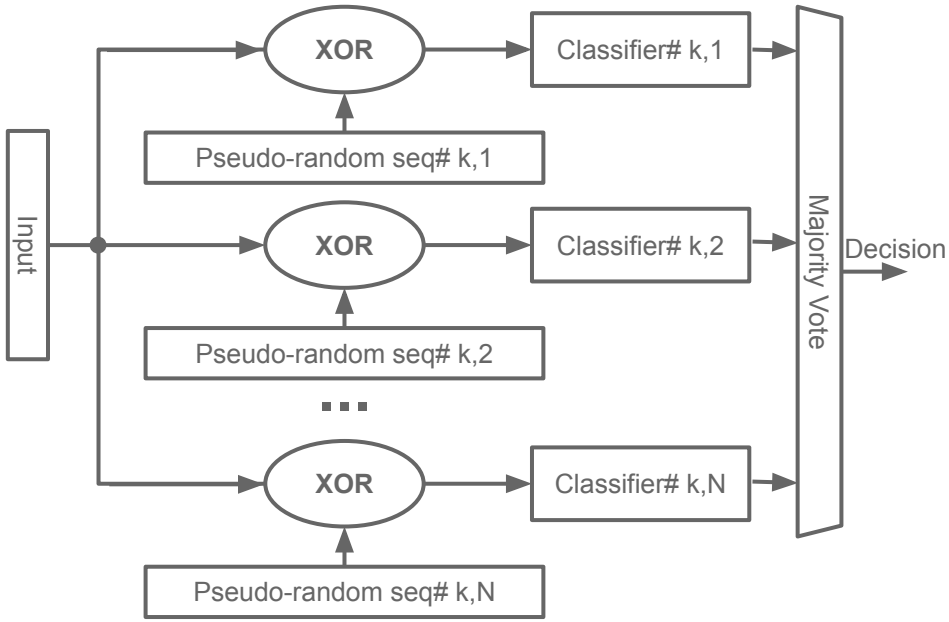
Conclusions

- The noise strategy doesn’t help
- The best results are achieved by adding the adversarial input to the training set. However this is attuned to a specific type of attack
- XOR scrambling shows some promise, especially for KNN. The intuition behind it is that it changes the decision boundaries in ways that the attack cannot anticipate

Conclusions

- The noise strategy doesn't help
- The best results are achieved by adding the adversarial input to the training set. However this is attuned to a specific type of attack
- XOR scrambling shows some promise, especially for KNN. The intuition behind it is that it changes the decision boundaries in ways that the attack cannot anticipate

Suggested XOR based architecture using k sets of pseudo-random sequences and specifically trained classifiers, k = 1..M



Thank You

Debbie

Joe

Robert

Damien

Metis

Skip

Rebekah

Lord Savage

Classmates

References

- http://www.cleverhans.io/
- https://github.com/tensorflow/cleverhans
- https://arxiv.org/abs/1602.02697
- https://blog.openai.com/adversarial-example-research/
- https://github.com/anishathalye/obfuscated-gradients