# Attacks on AI made easy

Elizaveta Tishina

Security Analysis Specialist, DeteAct

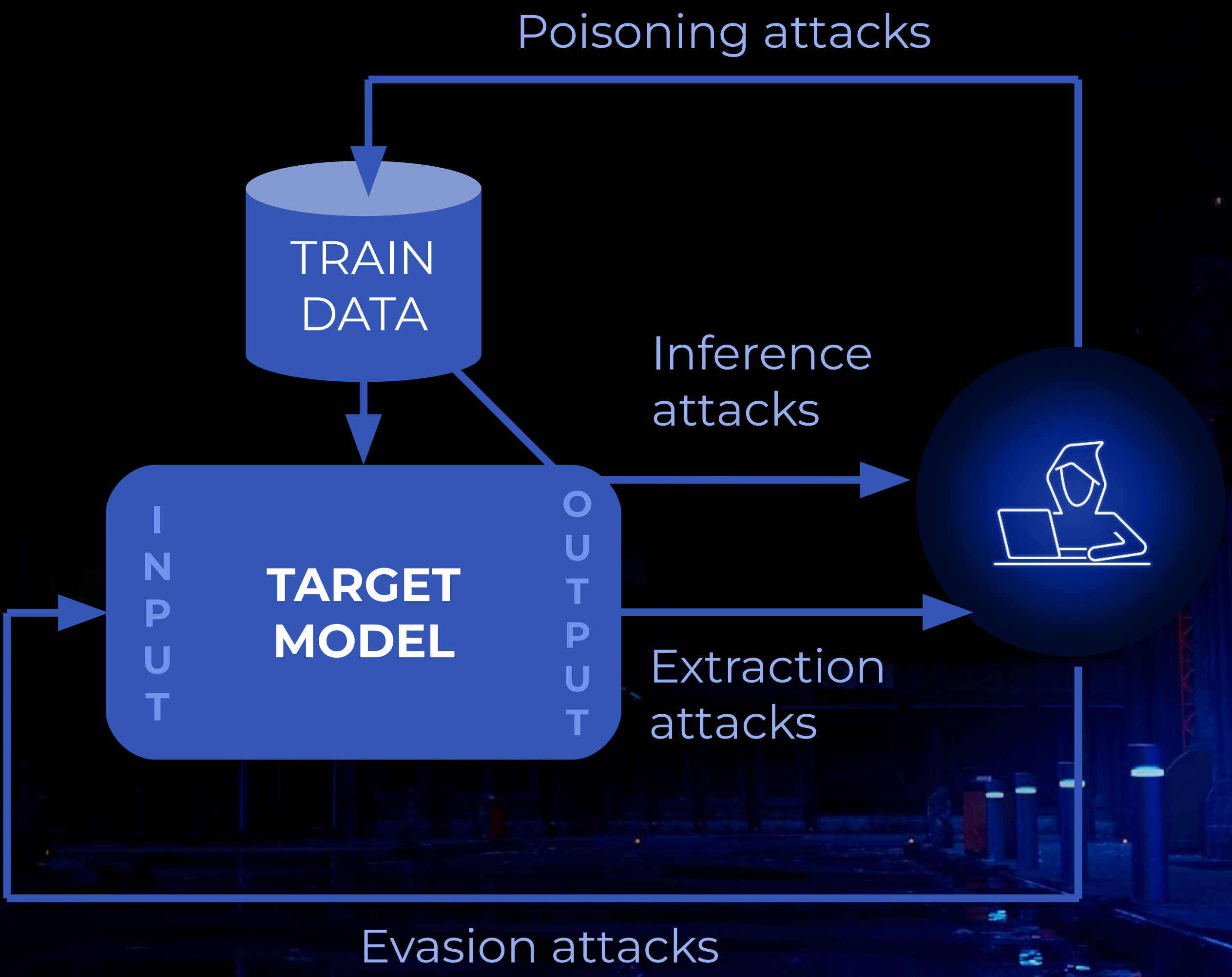
Moscow, August 26, 2022

# Machine Learning in products

- Recommendation systems
- Identity verification
- Malware detection
- Spam detection
- Search engines
- Translation
- Machine vision
- Fraud detection
- Analysis of results of medical tests

. . .

# Threats to Machine Learning models

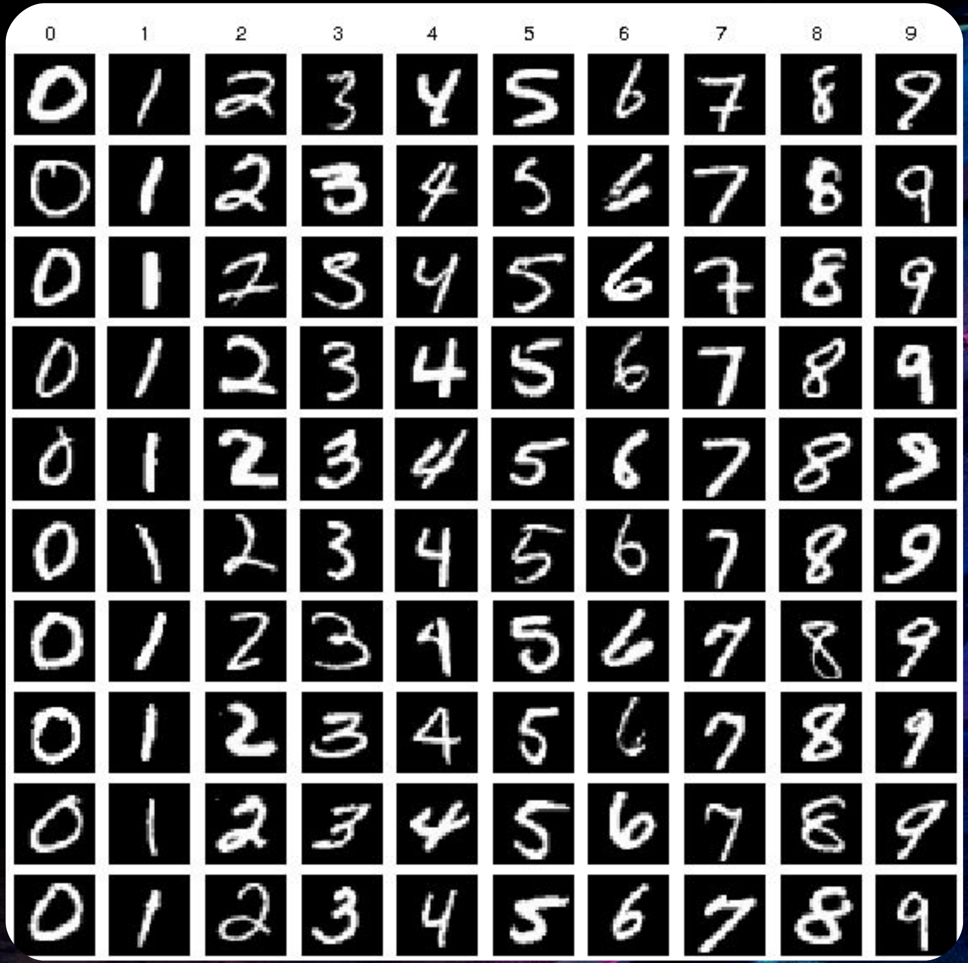


# Example model overview

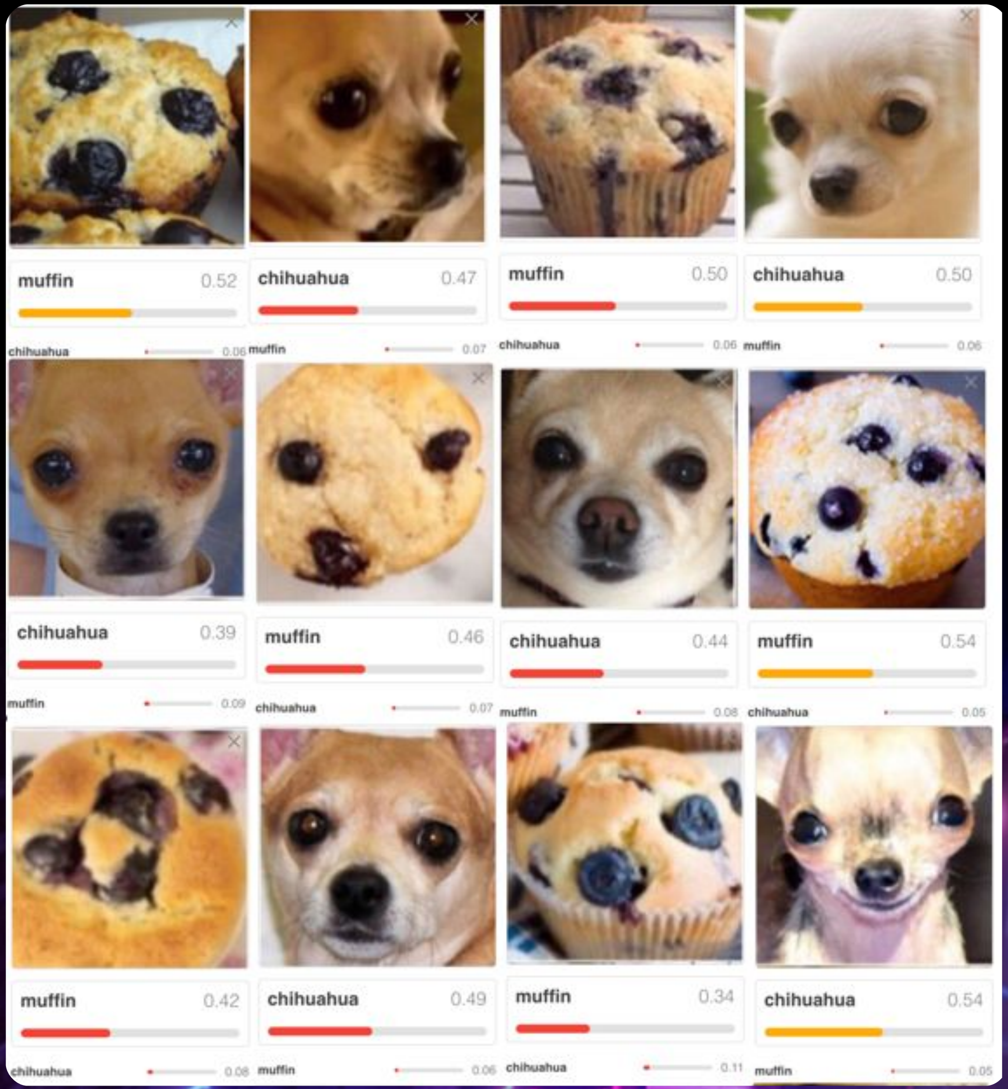
- **Train set**: MNIST dataset
- **Prerequisites**: API access to model
- **Attacker knowledge**: –

- Forms processing

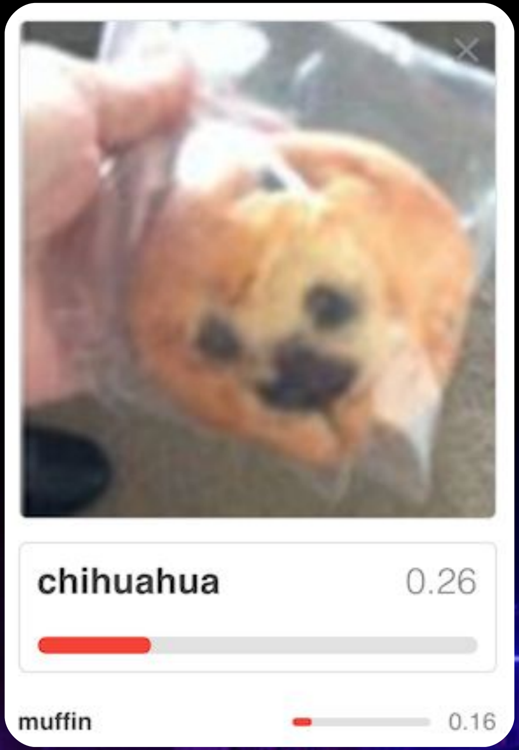
- Bank cheques processing



# Evasion attack: Overview



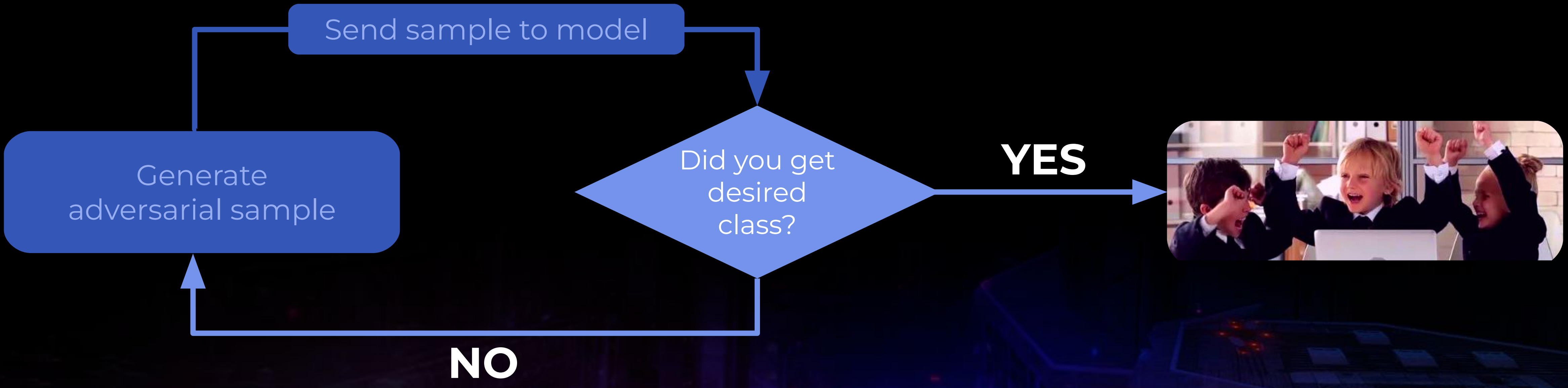
# Evasion attack: Overview



chihuahua 0.26

muffin 0.16

# Evasion attack: Overview



Send sample to model

Generate adversarial sample

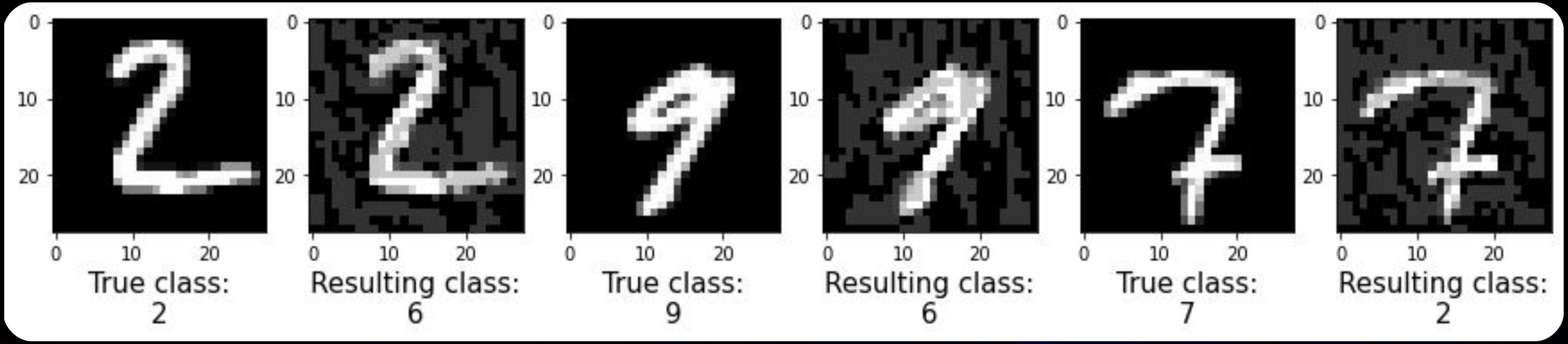
Did you get desired class?

YES

NO

# Evasion attack: Example

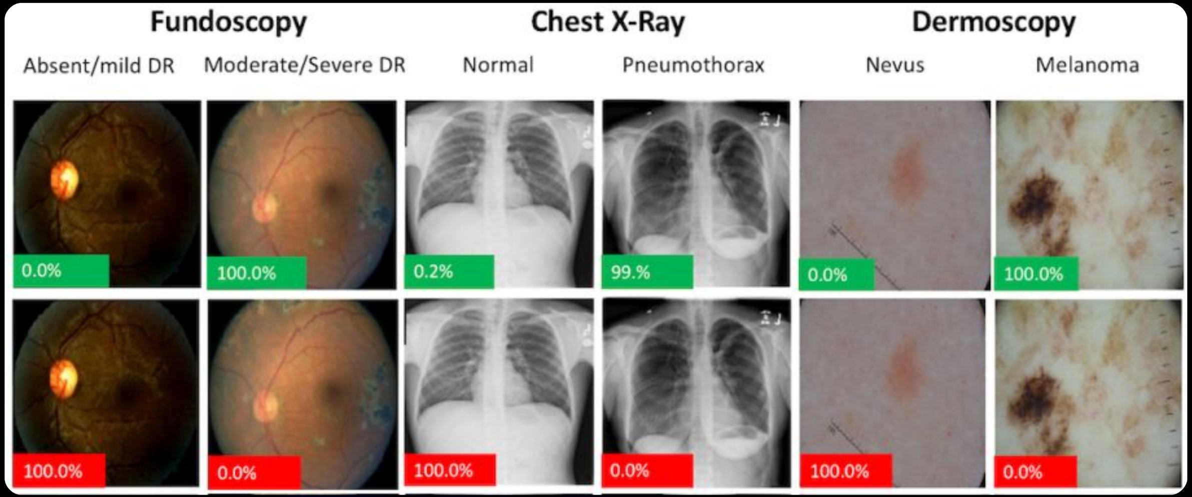
github.com/qwqoro/ML-Talk



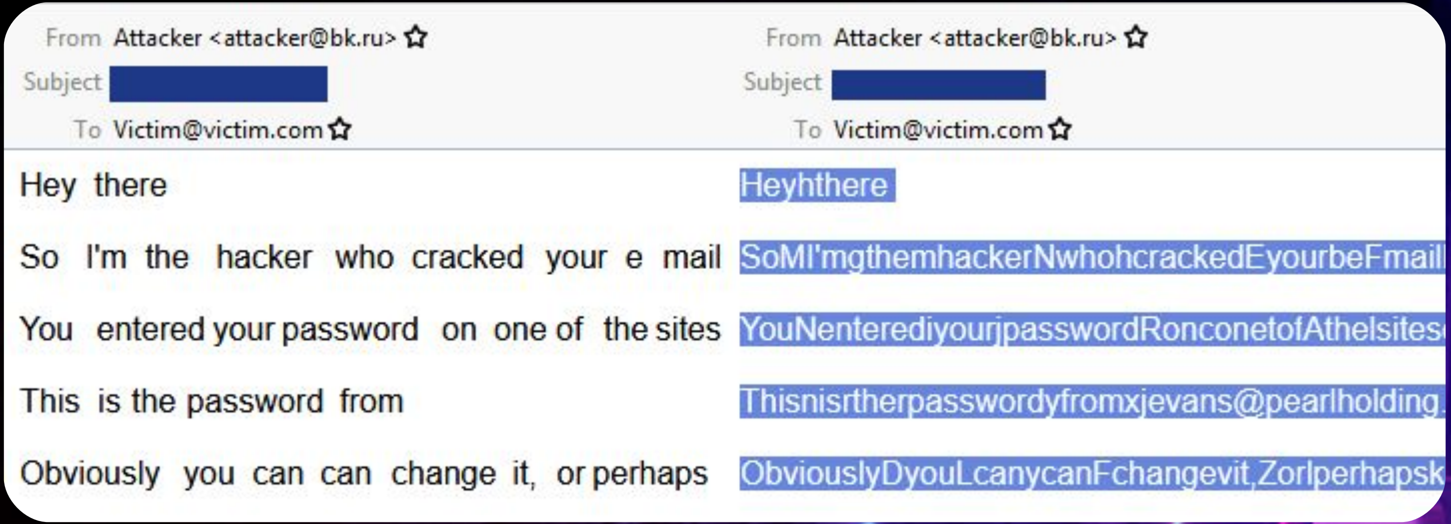
Approach: Fast Gradient Method [arXiv:1412.6572]

# Evasion attack: Impact

- Malware filters bypass / Antivirus evasion
- Spam e-mails & ad filters bypass
- Spoofing against verification systems
- Life-threatening situations

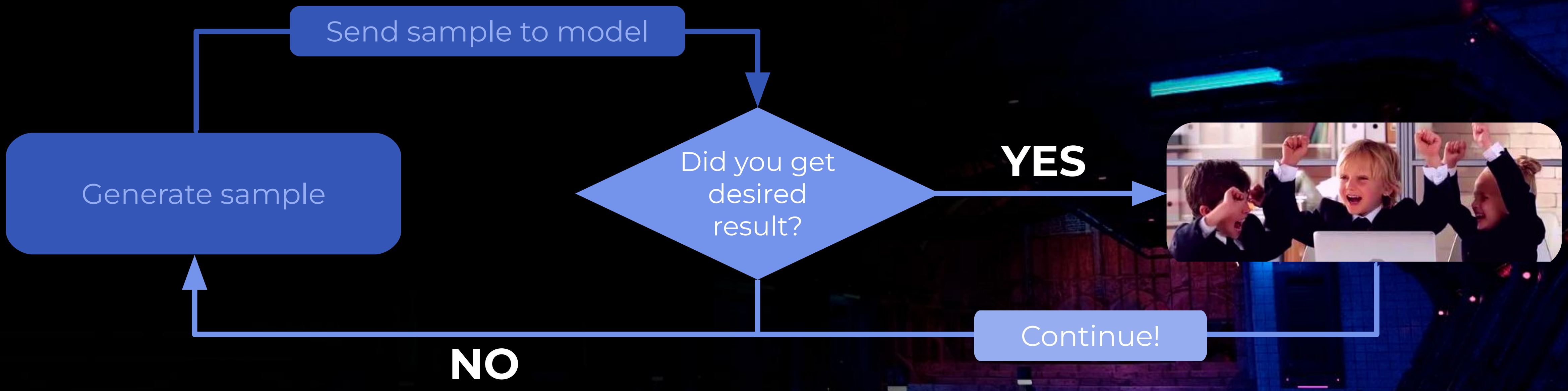


[arXiv:1804.05296]



| | |
|---|---|
| From Attacker <attacker@bk.ru> ☆ | From Attacker <attacker@bk.ru> ☆ |
| Subject [redacted] | Subject [redacted] |
| To Victim@victim.com ☆ | To Victim@victim.com ☆ |
| Hey there | Heyhthere |
| So I'm the hacker who cracked your e mail | SoMI'mgthemhackerNwhohcrackedEyourbeFmail |
| You entered your password on one of the sites | YouNenterediyourjpasswordRonconetofAthelsites |
| This is the password from | Thisnisrtherpasswordyfromxjevans@pearlholding. |
| Obviously you can can change it, or perhaps | ObviouslyDyouLcanycanFchangevit,Zorlperhapsk |

# Model inversion attack: Overview



Send sample to model

Generate sample

Did you get desired result?

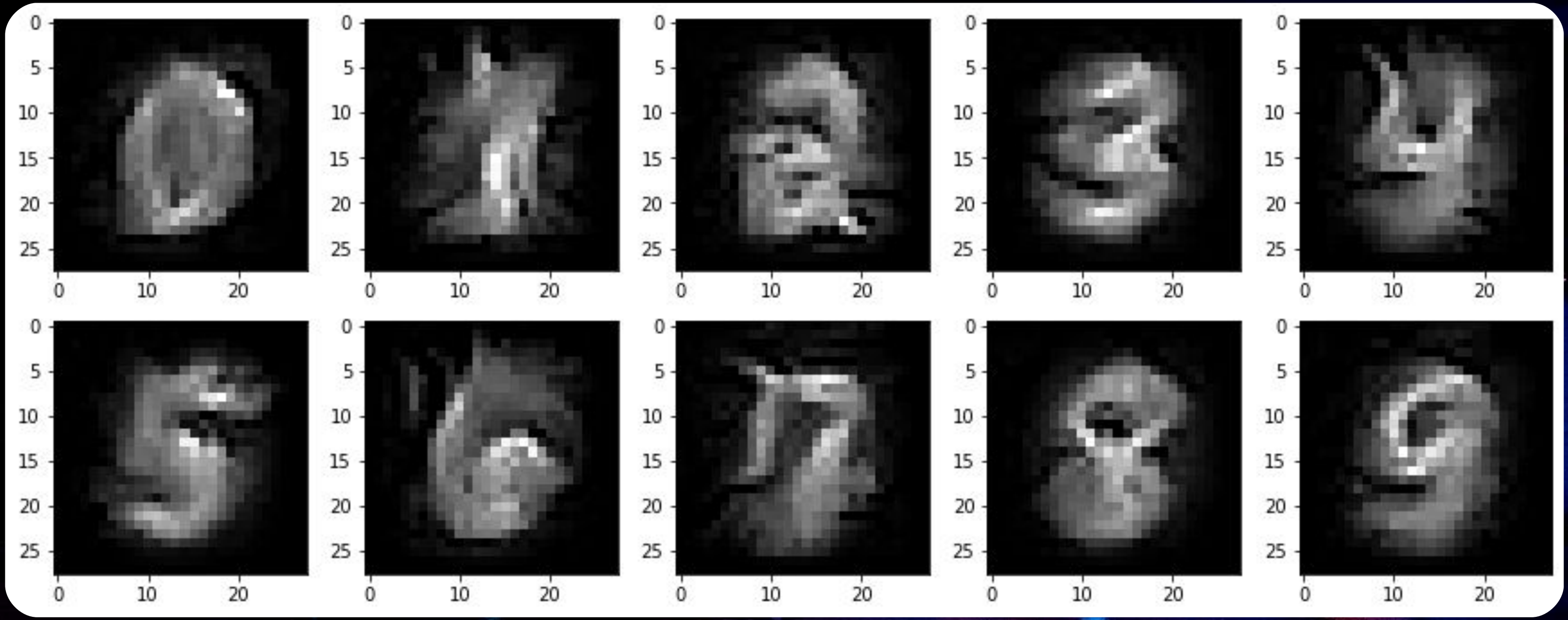
YES

NO

Continue!

# Model inversion attack: Example

github.com/qwqoro/ML-Talk

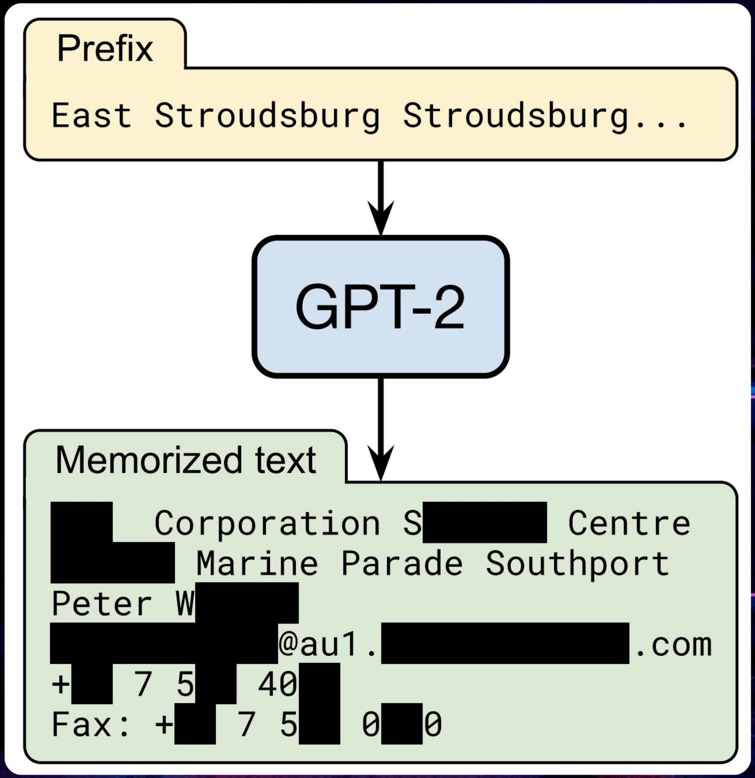


Approach: MIFace [DOI:10.1145/2810103.2813677]

# Model inversion attack: Impact

Leak of sensitive data:

- Contents of documents
- Medical records
- Passwords
- PIN codes
- Other secrets



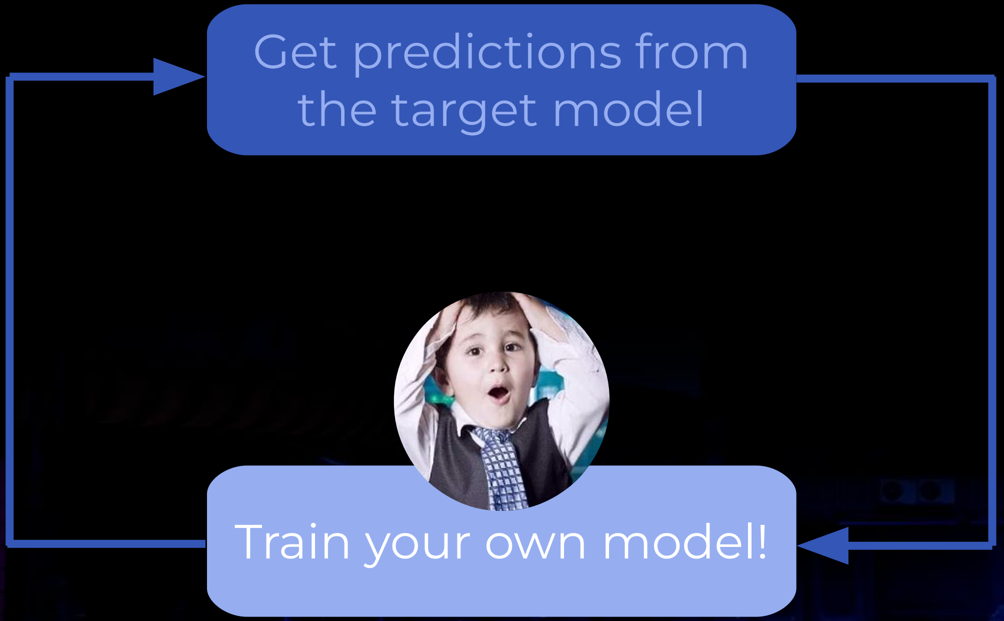
```
East Stroudsburg Stroudsburg...
```

```
[redacted]  Corporation S[redacted] Centre
[redacted]  Marine Parade Southport
Peter W[redacted]
[redacted]@au1.[redacted].com
+[redacted] 7 5[redacted] 40[redacted]
Fax: +[redacted] 7 5[redacted] 0[redacted]0
```

[arXiv:2012.07805]

# Model extraction attack: Overview & Impact

- Intellectual property infringement



Get predictions from the target model

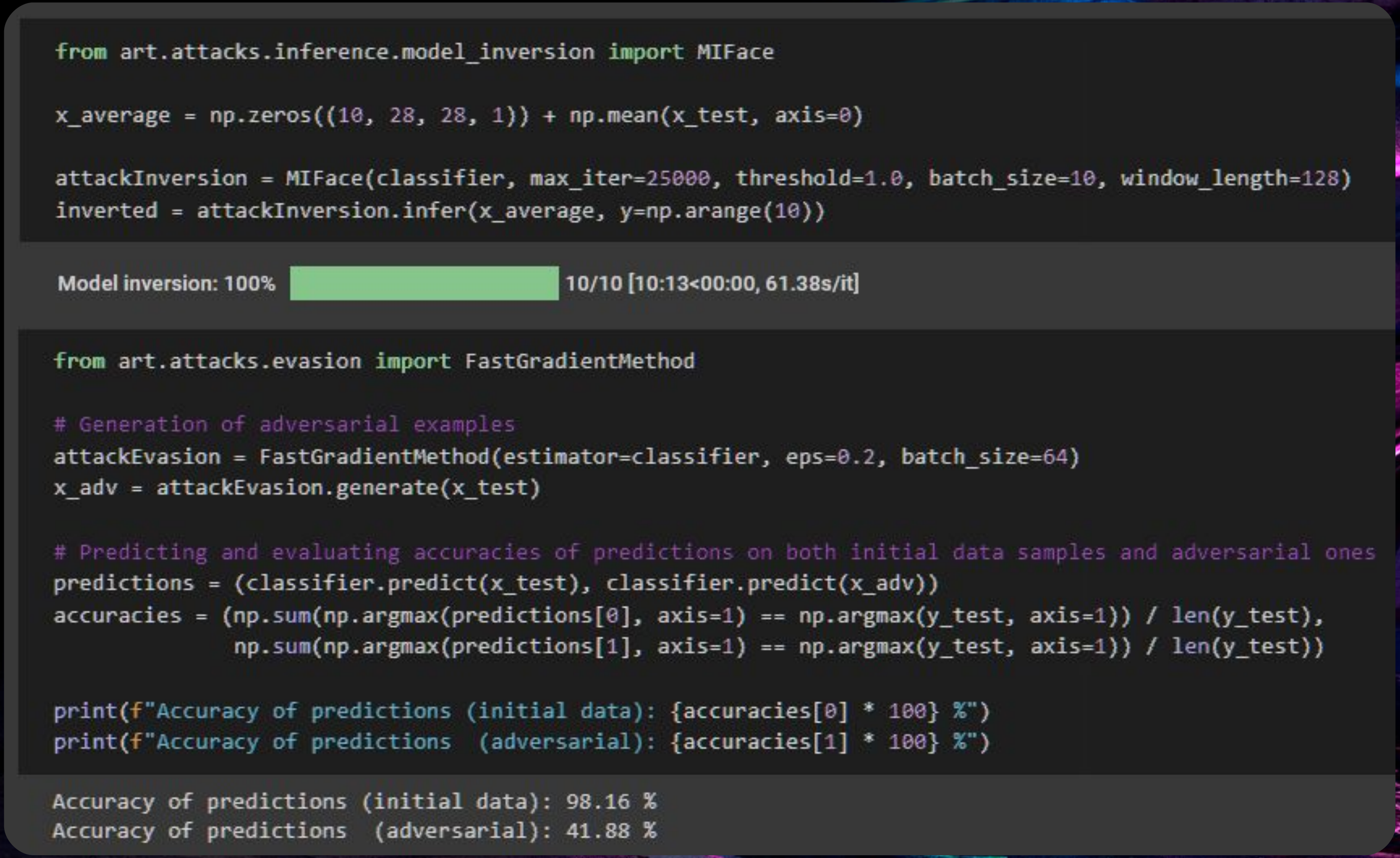
Train your own model!

# Adversarial Robustness Toolbox

github.com/Trusted-AI/adversarial-robustness-toolbox

- Attacks
- Defences
- Estimators
- Metrics
- Data generators

- Examples & detailed notebooks



```
from art.attacks.inference.model_inversion import MIFace

x_average = np.zeros((10, 28, 28, 1)) + np.mean(x_test, axis=0)

attackInversion = MIFace(classifier, max_iter=25000, threshold=1.0, batch_size=10, window_length=128)
inverted = attackInversion.infer(x_average, y=np.arange(10))
```

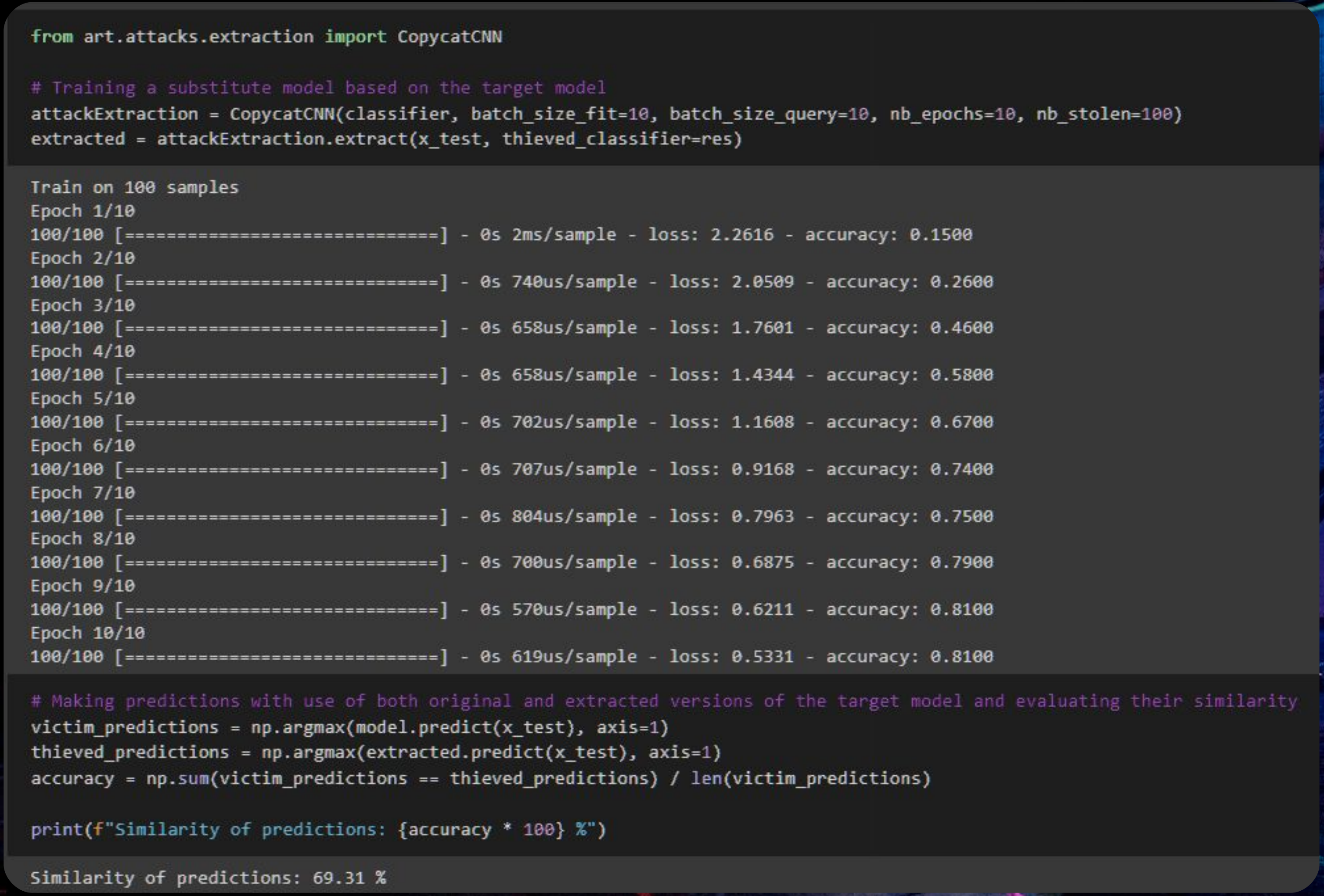
```
Model inversion: 100%                    10/10 [10:13<00:00, 61.38s/it]
```

```
from art.attacks.evasion import FastGradientMethod

# Generation of adversarial examples
attackEvasion = FastGradientMethod(estimator=classifier, eps=0.2, batch_size=64)
x_adv = attackEvasion.generate(x_test)

# Predicting and evaluating accuracies of predictions on both initial data samples and adversarial ones
predictions = (classifier.predict(x_test), classifier.predict(x_adv))
accuracies = (np.sum(np.argmax(predictions[0], axis=1) == np.argmax(y_test, axis=1)) / len(y_test),
              np.sum(np.argmax(predictions[1], axis=1) == np.argmax(y_test, axis=1)) / len(y_test))

print(f"Accuracy of predictions (initial data): {accuracies[0] * 100} %")
print(f"Accuracy of predictions  (adversarial): {accuracies[1] * 100} %")
```

```
Accuracy of predictions (initial data): 98.16 %
Accuracy of predictions  (adversarial): 41.88 %
```

# [ART] Model extraction attack: Example

github.com/qwqoro/ML-Talk



```python
from art.attacks.extraction import CopycatCNN

# Training a substitute model based on the target model
attackExtraction = CopycatCNN(classifier, batch_size_fit=10, batch_size_query=10, nb_epochs=10, nb_stolen=100)
extracted = attackExtraction.extract(x_test, thieved_classifier=res)
```

```
Train on 100 samples
Epoch 1/10
100/100 [==============================] - 0s 2ms/sample - loss: 2.2616 - accuracy: 0.1500
Epoch 2/10
100/100 [==============================] - 0s 740us/sample - loss: 2.0509 - accuracy: 0.2600
Epoch 3/10
100/100 [==============================] - 0s 658us/sample - loss: 1.7601 - accuracy: 0.4600
Epoch 4/10
100/100 [==============================] - 0s 658us/sample - loss: 1.4344 - accuracy: 0.5800
Epoch 5/10
100/100 [==============================] - 0s 702us/sample - loss: 1.1608 - accuracy: 0.6700
Epoch 6/10
100/100 [==============================] - 0s 707us/sample - loss: 0.9168 - accuracy: 0.7400
Epoch 7/10
100/100 [==============================] - 0s 804us/sample - loss: 0.7963 - accuracy: 0.7500
Epoch 8/10
100/100 [==============================] - 0s 700us/sample - loss: 0.6875 - accuracy: 0.7900
Epoch 9/10
100/100 [==============================] - 0s 570us/sample - loss: 0.6211 - accuracy: 0.8100
Epoch 10/10
100/100 [==============================] - 0s 619us/sample - loss: 0.5331 - accuracy: 0.8100
```

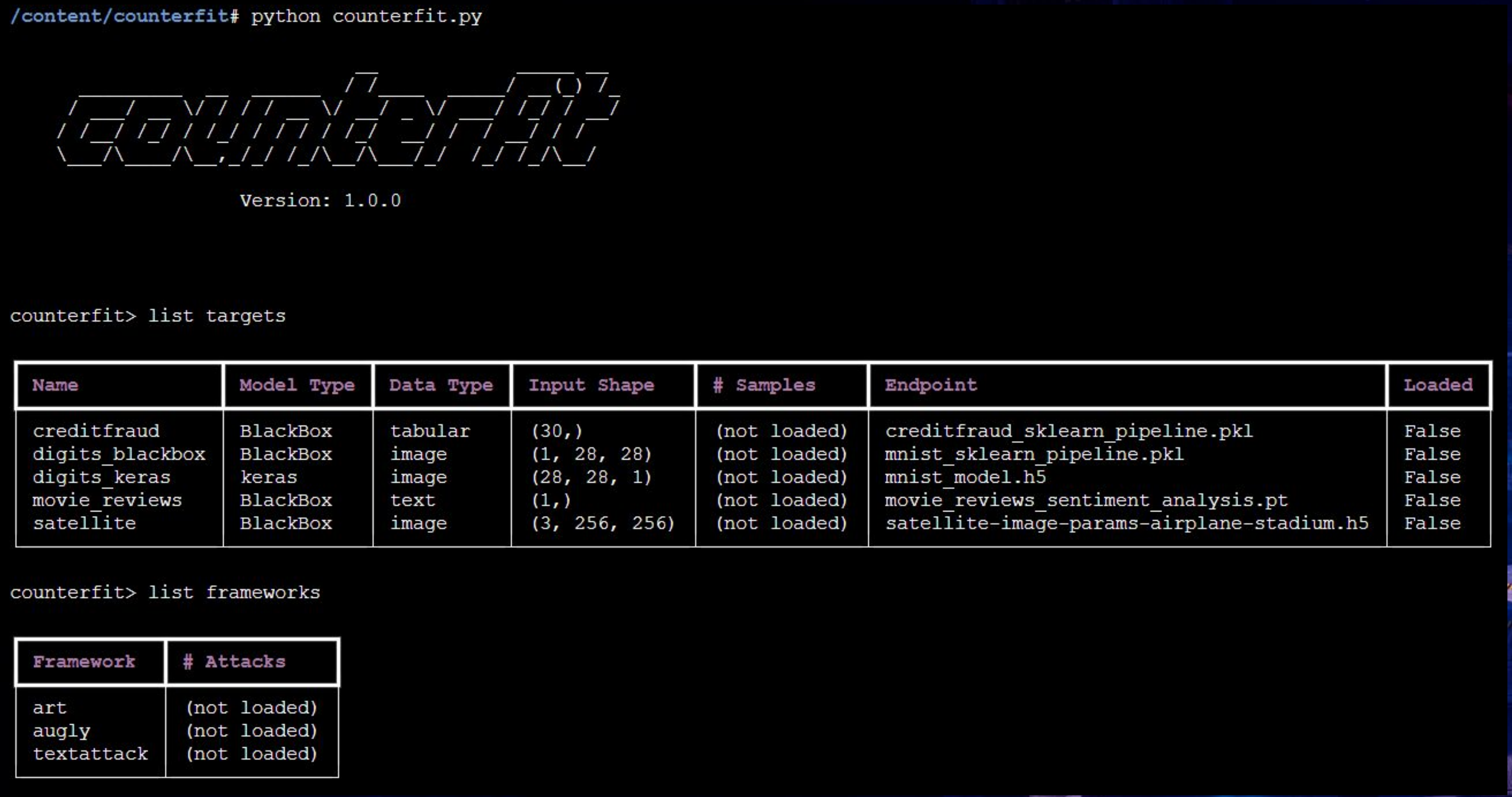
```python
# Making predictions with use of both original and extracted versions of the target model and evaluating their similarity
victim_predictions = np.argmax(model.predict(x_test), axis=1)
thieved_predictions = np.argmax(extracted.predict(x_test), axis=1)
accuracy = np.sum(victim_predictions == thieved_predictions) / len(victim_predictions)

print(f"Similarity of predictions: {accuracy * 100} %")
```

```
Similarity of predictions: 69.31 %
```

Approach: Copycat CNN [arXiv:1806.05476]

# Counterfit

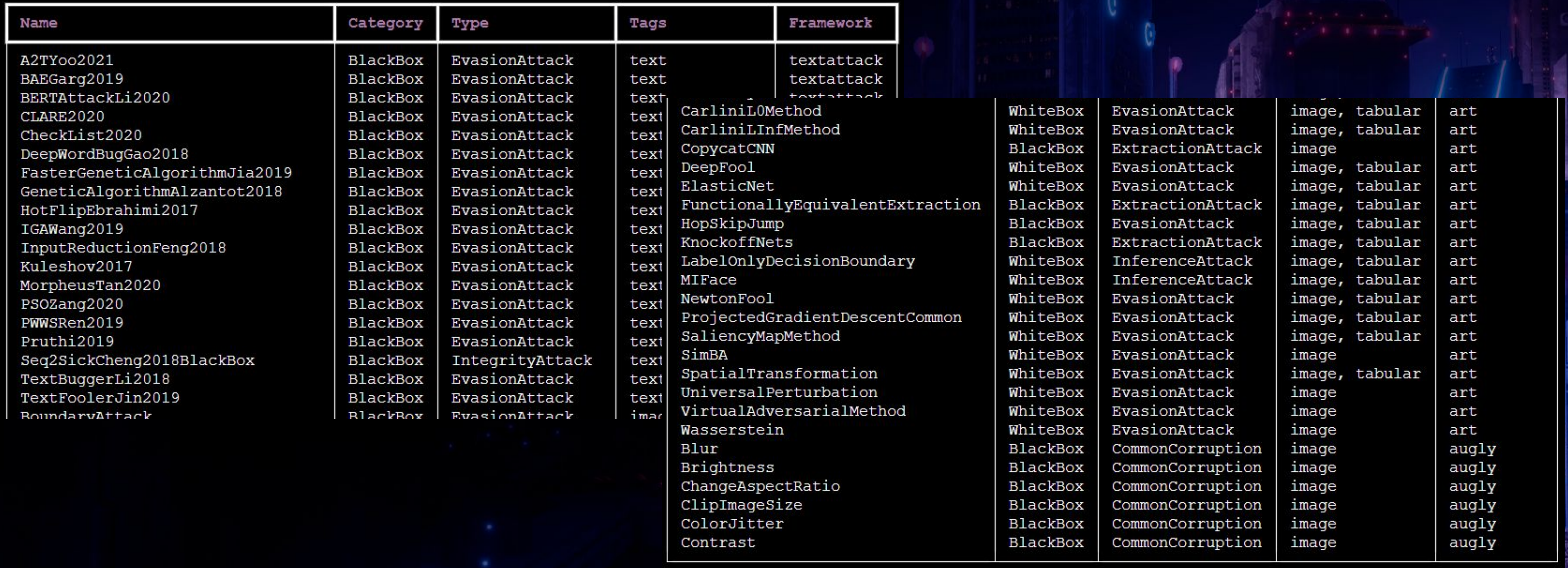
github.com/Azure/counterfit



```
/content/counterfit# python counterfit.py

                              __            _____ __
  _________  __  ______  / /____  _____/ __(_) /_
 / ___/ __ \/ / / / __ \/ __/ _ \/ ___/ /_/ / __/
/ /__/ /_/ / /_/ / / / / /_/  __/ /  / __/ / /_
\___/\____/\__,_/_/ /_/\__/\___/_/  /_/ /_/\__/

                Version: 1.0.0


counterfit> list targets

 Name             Model Type   Data Type   Input Shape     # Samples      Endpoint                                      Loaded

 creditfraud      BlackBox     tabular     (30,)           (not loaded)   creditfraud_sklearn_pipeline.pkl              False
 digits_blackbox  BlackBox     image       (1, 28, 28)     (not loaded)   mnist_sklearn_pipeline.pkl                    False
 digits_keras     keras        image       (28, 28, 1)     (not loaded)   mnist_model.h5                                False
 movie_reviews    BlackBox     text        (1,)            (not loaded)   movie_reviews_sentiment_analysis.pt           False
 satellite        BlackBox     image       (3, 256, 256)   (not loaded)   satellite-image-params-airplane-stadium.h5    False

counterfit> list frameworks

 Framework    # Attacks

 art          (not loaded)
 augly        (not loaded)
 textattack   (not loaded)
```

# Counterfit

github.com/Azure/counterfit

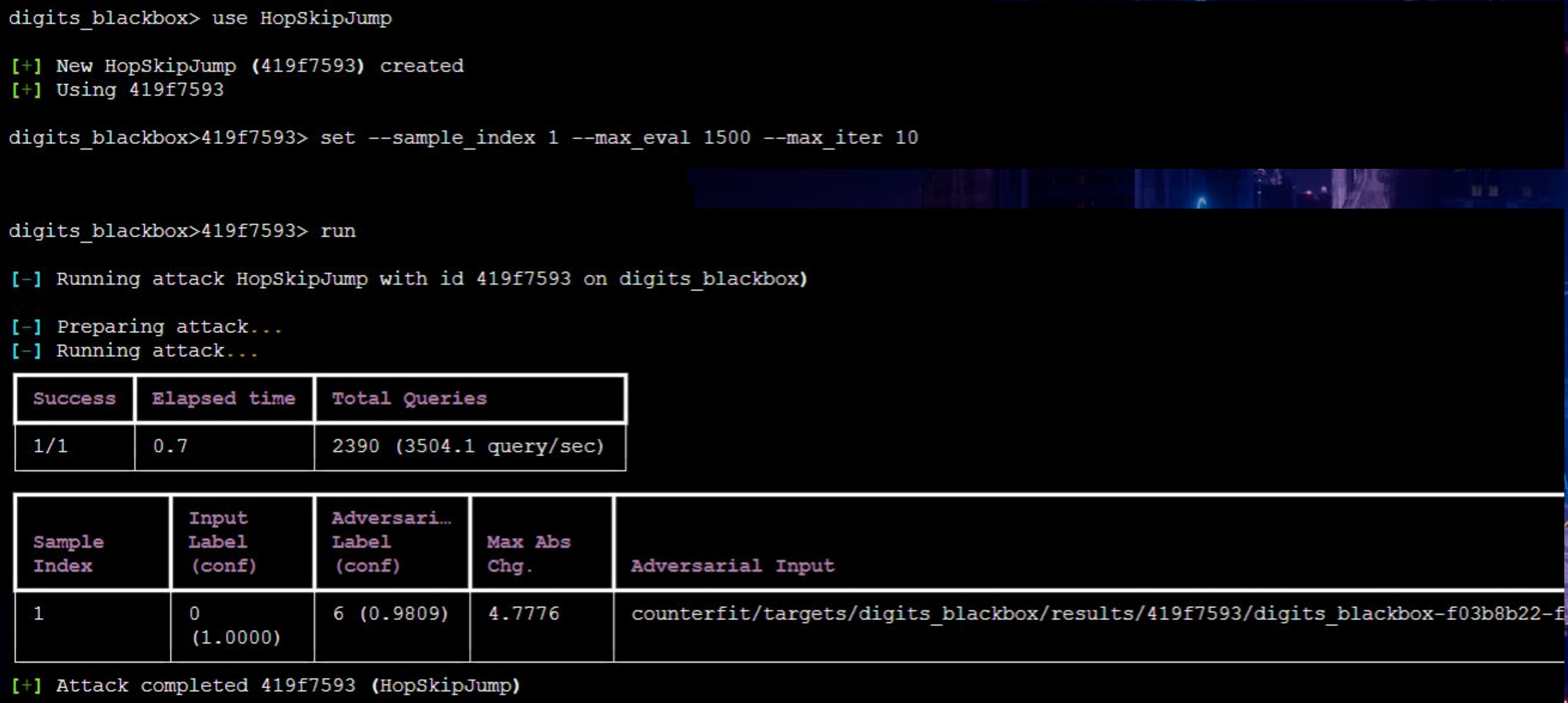


```
counterfit> list attacks

┌──────────────────────────────────┬──────────┬──────────────────┬────────────────┬────────────┐
│ Name                             │ Category │ Type             │ Tags           │ Framework  │
├──────────────────────────────────┼──────────┼──────────────────┼────────────────┼────────────┤
│ A2TYoo2021                       │ BlackBox │ EvasionAttack    │ text           │ textattack │
│ BAEGarg2019                      │ BlackBox │ EvasionAttack    │ text           │ textattack │
│ BERTAttackLi2020                 │ BlackBox │ EvasionAttack    │ text           │ textattack │
│ CLARE2020                        │ BlackBox │ EvasionAttack    │ text
│ CheckList2020                    │ BlackBox │ EvasionAttack    │ text
│ DeepWordBugGao2018               │ BlackBox │ EvasionAttack    │ text
│ FasterGeneticAlgorithmJia2019    │ BlackBox │ EvasionAttack    │ text
│ GeneticAlgorithmAlzantot2018     │ BlackBox │ EvasionAttack    │ text
│ HotFlipEbrahimi2017              │ BlackBox │ EvasionAttack    │ text
│ IGAWang2019                      │ BlackBox │ EvasionAttack    │ text
│ InputReductionFeng2018           │ BlackBox │ EvasionAttack    │ text
│ Kuleshov2017                     │ BlackBox │ EvasionAttack    │ text
│ MorpheusTan2020                  │ BlackBox │ EvasionAttack    │ text
│ PSOZang2020                      │ BlackBox │ EvasionAttack    │ text
│ PWWSRen2019                      │ BlackBox │ EvasionAttack    │ text
│ Pruthi2019                       │ BlackBox │ EvasionAttack    │ text
│ Seq2SickCheng2018BlackBox        │ BlackBox │ IntegrityAttack  │ text
│ TextBuggerLi2018                 │ BlackBox │ EvasionAttack    │ text
│ TextFoolerJin2019                │ BlackBox │ EvasionAttack    │ text
│ BoundaryAttack                   │ BlackBox │ EvasionAttack    │ imag
```

```
│ CarliniL0Method                     │ WhiteBox │ EvasionAttack      │ image, tabular │ art   │
│ CarliniLInfMethod                   │ WhiteBox │ EvasionAttack      │ image, tabular │ art   │
│ CopycatCNN                          │ BlackBox │ ExtractionAttack   │ image          │ art   │
│ DeepFool                            │ WhiteBox │ EvasionAttack      │ image, tabular │ art   │
│ ElasticNet                          │ WhiteBox │ EvasionAttack      │ image, tabular │ art   │
│ FunctionallyEquivalentExtraction    │ BlackBox │ ExtractionAttack   │ image, tabular │ art   │
│ HopSkipJump                         │ BlackBox │ EvasionAttack      │ image, tabular │ art   │
│ KnockoffNets                        │ BlackBox │ ExtractionAttack   │ image, tabular │ art   │
│ LabelOnlyDecisionBoundary           │ WhiteBox │ InferenceAttack    │ image, tabular │ art   │
│ MIFace                              │ WhiteBox │ InferenceAttack    │ image, tabular │ art   │
│ NewtonFool                          │ WhiteBox │ EvasionAttack      │ image, tabular │ art   │
│ ProjectedGradientDescentCommon      │ WhiteBox │ EvasionAttack      │ image, tabular │ art   │
│ SaliencyMapMethod                   │ WhiteBox │ EvasionAttack      │ image, tabular │ art   │
│ SimBA                               │ WhiteBox │ EvasionAttack      │ image          │ art   │
│ SpatialTransformation               │ WhiteBox │ EvasionAttack      │ image, tabular │ art   │
│ UniversalPerturbation               │ WhiteBox │ EvasionAttack      │ image          │ art   │
│ VirtualAdversarialMethod            │ WhiteBox │ EvasionAttack      │ image          │ art   │
│ Wasserstein                         │ WhiteBox │ EvasionAttack      │ image          │ art   │
│ Blur                                │ BlackBox │ CommonCorruption   │ image          │ augly │
│ Brightness                          │ BlackBox │ CommonCorruption   │ image          │ augly │
│ ChangeAspectRatio                   │ BlackBox │ CommonCorruption   │ image          │ augly │
│ ClipImageSize                       │ BlackBox │ CommonCorruption   │ image          │ augly │
│ ColorJitter                         │ BlackBox │ CommonCorruption   │ image          │ augly │
│ Contrast                            │ BlackBox │ CommonCorruption   │ image          │ augly │
└─────────────────────────────────────┴──────────┴────────────────────┴────────────────┴───────┘
```

# [Counterfit] Evasion attack: Example

github.com/qwqoro/ML-Talk

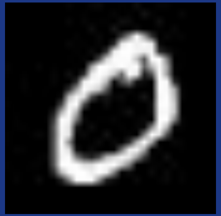
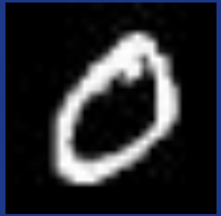
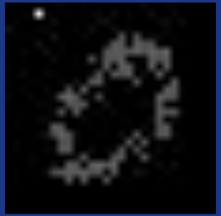
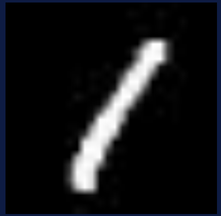
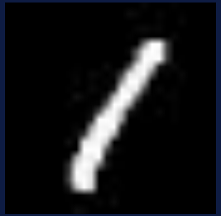
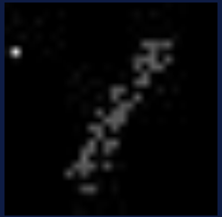
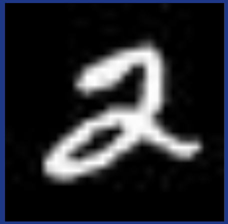
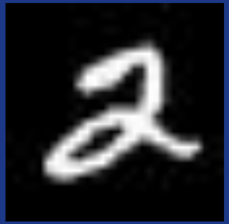
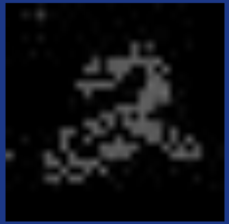


```
digits_blackbox> use HopSkipJump

[+] New HopSkipJump (419f7593) created
[+] Using 419f7593

digits_blackbox>419f7593> set --sample_index 1 --max_eval 1500 --max_iter 10


digits_blackbox>419f7593> run

[-] Running attack HopSkipJump with id 419f7593 on digits_blackbox)

[-] Preparing attack...
[-] Running attack...
┌─────────┬──────────────┬─────────────────────────┐
│ Success │ Elapsed time │ Total Queries           │
├─────────┼──────────────┼─────────────────────────┤
│ 1/1     │ 0.7          │ 2390 (3504.1 query/sec) │
└─────────┴──────────────┴─────────────────────────┘

┌─────────────┬──────────┬─────────────┬─────────┬─────────────────────────────────────────────────────────────────────
│             │ Input    │ Adversari…  │         │
│ Sample      │ Label    │ Label       │ Max Abs │
│ Index       │ (conf)   │ (conf)      │ Chg.    │ Adversarial Input
├─────────────┼──────────┼─────────────┼─────────┼─────────────────────────────────────────────────────────────────────
│ 1           │ 0        │ 6 (0.9809)  │ 4.7776  │ counterfit/targets/digits_blackbox/results/419f7593/digits_blackbox-f03b8b22-f
│             │ (1.0000) │             │         │
└─────────────┴──────────┴─────────────┴─────────┴─────────────────────────────────────────────────────────────────────
[+] Attack completed 419f7593 (HopSkipJump)
```

Approach: HopSkipJump [arXiv:1904.02144]

# [Counterfit] Evasion attack: Example

github.com/qwqoro/ML-Talk

| Original image | Adversarial version of image | Difference (exaggerated) | Original class | Original confidence | Resulting class | Resulting confidence |
|---|---|---|---|---|---|---|
|  |  |  | 0 | 100% | 6 | 98% |
|  |  |  | 1 | 100% | 8 | 56% |
|  |  |  | 2 | 100% | 4 | 73% |

# Counterfit

github.com/Azure/counterfit

- Hiding malicious queries
- Using a proxy
- Sending and collecting outputs from different locations
- Adding startup commands
- Overriding functions in the parent target
- Training a local model to attack

OFFZONE
2022

DETEACT

@qwqoro

qwqoro/ML-Talk

OFFZONE

2022