```
!pip install -U numpy==1.23.5 scikit-learn==1.1.3 tensorflow==2.12.0
adversarial-robustness-toolbox==1.14.0
```

```
Requirement already satisfied: numpy==1.23.5 in
/usr/local/lib/python3.11/dist-packages (1.23.5)
Requirement already satisfied: scikit-learn==1.1.3 in
/usr/local/lib/python3.11/dist-packages (1.1.3)
Requirement already satisfied: tensorflow==2.12.0 in
/usr/local/lib/python3.11/dist-packages (2.12.0)
Requirement already satisfied: adversarial-robustness-toolbox==1.14.0
in /usr/local/lib/python3.11/dist-packages (1.14.0)
Requirement already satisfied: scipy>=1.3.2 in
/usr/local/lib/python3.11/dist-packages (from scikit-learn==1.1.3)
(1.15.3)
Requirement already satisfied: joblib>=1.0.0 in
/usr/local/lib/python3.11/dist-packages (from scikit-learn==1.1.3)
(1.5.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.11/dist-packages (from scikit-learn==1.1.3)
(3.6.0)
Requirement already satisfied: absl-py>=1.0.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(1.6.3)
Requirement already satisfied: flatbuffers>=2.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(25.2.10)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(1.71.0)
Requirement already satisfied: h5py>=2.9.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(3.13.0)
Requirement already satisfied: jax>=0.3.15 in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(0.4.30)
Requirement already satisfied: keras<2.13,>=2.12.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(2.12.0)
Requirement already satisfied: libclang>=13.0.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(18.1.1)
```

```
Requirement already satisfied: opt-einsum>=2.3.2 in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(3.4.0)
Requirement already satisfied: packaging in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(24.2)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!
=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(4.25.8)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(75.2.0)
Requirement already satisfied: six>=1.12.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(1.17.0)
Requirement already satisfied: tensorboard<2.13,>=2.12 in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(2.12.3)
Requirement already satisfied: tensorflow-estimator<2.13,>=2.12.0
in /usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(2.12.0)
Requirement already satisfied: termcolor>=1.1.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(3.1.0)
Requirement already satisfied: typing-extensions>=3.6.6 in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(4.13.2)
Requirement already satisfied: wrapt<1.15,>=1.11.0 in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(1.14.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in
/usr/local/lib/python3.11/dist-packages (from tensorflow==2.12.0)
(0.37.1)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-
packages (from adversarial-robustness-toolbox==1.14.0) (4.67.1)
Requirement already satisfied: wheel<1.0,>=0.23.0 in
/usr/local/lib/python3.11/dist-packages (from astunparse>=1.6.0-
>tensorflow==2.12.0) (0.45.1)
Requirement already satisfied: jaxlib<=0.4.30,>=0.4.27 in
/usr/local/lib/python3.11/dist-packages (from jax>=0.3.15-
>tensorflow==2.12.0) (0.4.30)
Requirement already satisfied: ml-dtypes>=0.2.0 in
/usr/local/lib/python3.11/dist-packages (from jax>=0.3.15-
>tensorflow==2.12.0) (0.4.1)
Requirement already satisfied: google-auth<3,>=1.6.3 in
/usr/local/lib/python3.11/dist-packages (from tensorboard<2.13,>=2.12-
>tensorflow==2.12.0) (2.38.0)
Requirement already satisfied: google-auth-oauthlib<1.1,>=0.5 in
```

```
/usr/local/lib/python3.11/dist-packages (from tensorboard<2.13,>=2.12-
>tensorflow==2.12.0) (1.0.0)
Requirement already satisfied: markdown>=2.6.8 in
/usr/local/lib/python3.11/dist-packages (from tensorboard<2.13,>=2.12-
>tensorflow==2.12.0) (3.8)
Requirement already satisfied: requests<3,>=2.21.0 in
/usr/local/lib/python3.11/dist-packages (from tensorboard<2.13,>=2.12-
>tensorflow==2.12.0) (2.32.3)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0
in /usr/local/lib/python3.11/dist-packages (from
tensorboard<2.13,>=2.12->tensorflow==2.12.0) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in
/usr/local/lib/python3.11/dist-packages (from tensorboard<2.13,>=2.12-
>tensorflow==2.12.0) (3.1.3)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/usr/local/lib/python3.11/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.13,>=2.12->tensorflow==2.12.0) (5.5.2)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/usr/local/lib/python3.11/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.13,>=2.12->tensorflow==2.12.0) (0.4.2)
Requirement already satisfied: rsa<5,>=3.1.4 in
/usr/local/lib/python3.11/dist-packages (from google-auth<3,>=1.6.3-
>tensorboard<2.13,>=2.12->tensorflow==2.12.0) (4.9.1)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/usr/local/lib/python3.11/dist-packages (from google-auth-
oauthlib<1.1,>=0.5->tensorboard<2.13,>=2.12->tensorflow==2.12.0)
(2.0.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.13,>=2.12->tensorflow==2.12.0) (3.4.2)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.13,>=2.12->tensorflow==2.12.0) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.13,>=2.12->tensorflow==2.12.0) (2.4.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.21.0-
>tensorboard<2.13,>=2.12->tensorflow==2.12.0) (2025.4.26)
Requirement already satisfied: MarkupSafe>=2.1.1 in
/usr/local/lib/python3.11/dist-packages (from werkzeug>=1.0.1-
>tensorboard<2.13,>=2.12->tensorflow==2.12.0) (3.0.2)
Requirement already satisfied: pyasn1<0.7.0,>=0.6.1 in
/usr/local/lib/python3.11/dist-packages (from pyasn1-modules>=0.2.1-
>google-auth<3,>=1.6.3->tensorboard<2.13,>=2.12->tensorflow==2.12.0)
(0.6.1)
Requirement already satisfied: oauthlib>=3.0.0 in
/usr/local/lib/python3.11/dist-packages (from requests-
oauthlib>=0.7.0->google-auth-oauthlib<1.1,>=0.5-
>tensorboard<2.13,>=2.12->tensorflow==2.12.0) (3.2.2)
```

Импорт библиотек

```
import warnings
warnings.filterwarnings('ignore')
import tensorflow as tf
tf.compat.v1.disable_eager_execution()
import numpy as np
from matplotlib import pyplot as plt
from art.estimators.classification import KerasClassifier
from art.attacks.evasion import FastGradientMethod
from art.defences.trainer import AdversarialTrainer
```

Загрузка датасета и разбиение на выборки

```
(x_train, y_train), (x_test, y_test) =
tf.keras.datasets.mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
```

Создание модели и ее обучение

```
model = tf.keras.models.Sequential([

    # Входной слой: принимает изображения 28x28 пикселей
    tf.keras.layers.InputLayer(input_shape=(28, 28)),
    # "Разворачивает" изображение в вектор из 784 элементов (28*28)
    tf.keras.layers.Flatten(),
    # Полносвязный слой на 128 нейронов с функцией активации ReLU
    tf.keras.layers.Dense(128, activation='relu'),
    # Слой, отключающий случайные нейроны
    tf.keras.layers.Dropout(0.2),
    # Выходной слой на 10 нейронов (по числу классов – цифры от 0 до
9), softmax – для получения вероятностей
    tf.keras.layers.Dense(10, activation='softmax')
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
```

```
Train on 60000 samples
Epoch 1/5
60000/60000 [==============================] - 10s 161us/sample -
loss: 0.2913 - accuracy: 0.9156
Epoch 2/5
60000/60000 [==============================] - 8s 137us/sample - loss:
0.1403 - accuracy: 0.9586
Epoch 3/5
60000/60000 [==============================] - 7s 122us/sample - loss:
0.1050 - accuracy: 0.9681
Epoch 4/5
60000/60000 [==============================] - 4s 69us/sample - loss:
0.0876 - accuracy: 0.9729
Epoch 5/5
60000/60000 [==============================] - 5s 86us/sample - loss:
0.0737 - accuracy: 0.9767

<keras.callbacks.History at 0x7d8425814e50>
```

Оценка модели

```
loss_test, accuracy_test = model.evaluate(x_test, y_test)

print('Точность (чистые данные): {:4.2f}%'.format(accuracy_test *
100))

Точность (чистые данные): 97.72%
```

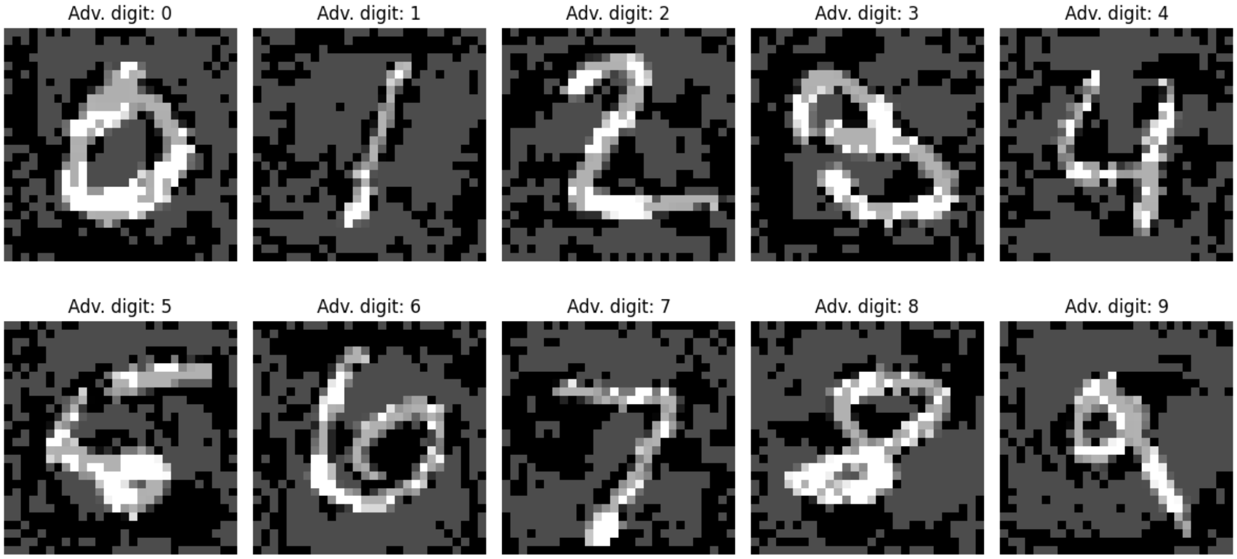
Атака

```
# Оборачивание модели в ART-классификатор (необходим для работы с
библиотекой adversarial-атак)
classifier = KerasClassifier(model=model, clip_values=(0, 1))

# Атака FGSM (Fast Gradient Sign Method)
attack_fgsm = FastGradientMethod(estimator=classifier, eps=0.3)

# Генерация поврежденных примеров
x_test_adv = attack_fgsm.generate(x_test)

loss_test_adv, accuracy_test_adv = model.evaluate(x_test_adv, y_test)

# Вычисление среднего искажения
perturbation = np.mean(np.abs((x_test_adv - x_test)))

print('Точность (поврежденные данные): {:4.2f}
%'.format(accuracy_test_adv * 100))
print('Среднее искажение: {:4.2f}'.format(perturbation))
```

```
Точность (поврежденные данные): 1.52%
Среднее искажение: 0.18
```

Визуализация

```
# Создание словаря с поврежденными цифрами
unique_digits = {}
for i in range(len(x_test)):
    label = y_test[i]  # Истинная метка (цифра)
    if label not in unique_digits:
        # Если для этой цифры ещё нет картинки — добавляем первую
попавшуюся атакованную
        unique_digits[label] = x_test_adv[i]
    if len(unique_digits) == 10:
        break

# Вывод 10 поврежденных изображений
plt.figure(figsize=(12, 6))
for i, digit in enumerate(sorted(unique_digits)):
    plt.subplot(2, 5, i + 1)
    plt.imshow(unique_digits[digit], cmap='gray')
    plt.title(f"Adv. digit: {digit}")
    plt.axis('off')
plt.tight_layout()
plt.show()
```



Защита

```
model_defended = tf.keras.models.Sequential([
    tf.keras.layers.InputLayer(input_shape=(28, 28)),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation='softmax')
])

model_defended.compile(optimizer='adam',
                       loss='sparse_categorical_crossentropy',
                       metrics=['accuracy'])

# Оборачивание модели в классификатор ART — для использования adversarial training и атак
classifier_defended = KerasClassifier(model=model_defended, clip_values=(0, 1))

# Создание FGSM-атаки
attack_for_training = FastGradientMethod(estimator=classifier_defended, eps=0.3)

# Создание объекта AdversarialTrainer с долей adversarial-примеров 50% (ratio=0.5)
trainer = AdversarialTrainer(classifier=classifier_defended,
                             attacks=attack_for_training,
                             ratio=0.5)

# Запускаем обучение защищённой модели с adversarial training
trainer.fit(x_train, y_train, nb_epochs=5, batch_size=64)
```

```
{"model_id":"ebd16613ea1a4a13b03991a0e28c2038","version_major":2,"version_minor":0}

{"model_id":"4b72c6fcaa1c4613b6182624c4e2433c","version_major":2,"version_minor":0}
```

Оценка защиты

```
attack_fgsm_def = FastGradientMethod(estimator=classifier_defended, eps=0.3)
x_test_adv_def = attack_fgsm_def.generate(x_test)

loss_clean, acc_clean_def = model_defended.evaluate(x_test, y_test, verbose=0)
loss_adv, acc_adv_def = model_defended.evaluate(x_test_adv_def, y_test, verbose=0)
```

```
print(f"\nТочность (чистые данные с защитой): {acc_clean_def *
100:.2f}%")
print(f"Точность (поврежденные данные с защитой): {acc_adv_def *
100:.2f}%")
```

```
Точность (чистые данные с защитой): 96.87%
Точность (поврежденные данные с защитой): 59.94%
```

Сравнение результатов

```
print("\n=== Сравнение точностей ===")
print(f"Точность (чистые данные):                   {accuracy_test *
100:.2f}%")
print(f"Точность (поврежденные данные):             {accuracy_test_adv
* 100:.2f}%")
print(f"Точность (чистые данные с защитой):         {acc_clean_def *
100:.2f}%")
print(f"Точность (поврежденные данные с защитой):   {acc_adv_def *
100:.2f}%")
```

```
=== Сравнение точностей ===
Точность (чистые данные):                   97.72%
Точность (поврежденные данные):             1.52%
Точность (чистые данные с защитой):         96.87%
Точность (поврежденные данные с защитой):   59.94%
```