МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«МИРЭА – Российский технологический университет»

Институт кибербезопасности и цифровых технологий

Кафедра КБ-4 «Интеллектуальные системы информационной безопасности»

Отчёт по практической работе № 3

По дисциплине

«Анализ защищенности систем искусственного интеллекта»

Выполнил:

ББМО–02–22

Шмарковский М. Б.

Проверил:

Спирин А. А.

Москва, 2024

Практическая работа №3

Выполнил Шмарковский МБ ББМО-02-22

Ход работы

Вариант 31

```
#Установим ART adversarial-robustness-toolbox
!pip install adversarial-robustness-toolbox
```

```
Collecting adversarial-robustness-toolbox
  Downloading adversarial_robustness_toolbox-1.17.0-py3-none-any.whl (1.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.7/1.7 MB 7.4 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.18.0 in /usr/local/lib/python3.10/dist-packages (from adversarial-robustness-toolbox) (1.23
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from adversarial-robustness-toolbox) (1.11.
Collecting scikit-learn<1.2.0,>=0.22.2 (from adversarial-robustness-toolbox)
  Downloading scikit_learn-1.1.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (30.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 30.5/30.5 MB 36.4 MB/s eta 0:00:00
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from adversarial-robustness-toolbox) (1.16.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from adversarial-robustness-toolbox) (67.7.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from adversarial-robustness-toolbox) (4.66.1)
Requirement already satisfied: joblib>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn<1.2.0,>=0.22.2->adversar
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn<1.2.0,>=0.22.2->a
Installing collected packages: scikit-learn, adversarial-robustness-toolbox
  Attempting uninstall: scikit-learn
    Found existing installation: scikit-learn 1.2.2
    Uninstalling scikit-learn-1.2.2:
      Successfully uninstalled scikit-learn-1.2.2
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the so
bigframes 0.19.2 requires scikit-learn>=1.2.2, but you have scikit-learn 1.1.3 which is incompatible.
Successfully installed adversarial-robustness-toolbox-1.17.0 scikit-learn-1.1.3
```

```
#Импортируем необходимые библиотеки
import numpy as np
import tensorflow as tf
from art.attacks.poisoning.backdoor_attack_dgm.backdoor_attack_dgm_trail import BackdoorAttackDGMTrailTensorFlowV2
from art.estimators.gan.tensorflow import TensorFlowV2GAN
from art.estimators.generation.tensorflow import TensorFlowV2Generator
from art.estimators.classification.tensorflow import TensorFlowV2Classifier

np.random.seed(100)
tf.random.set_seed(100)


#Создаем класс для модели-генератора изображений
def make_generator_model(capacity: int, z_dim: int) -> tf.keras.Sequential():
  model = tf.keras.Sequential()

  model.add(tf.keras.layers.Dense(capacity * 7 * 7 * 4, use_bias=False, input_shape=(z_dim,)))
  model.add(tf.keras.layers.BatchNormalization())
  model.add(tf.keras.layers.LeakyReLU())

  model.add(tf.keras.layers.Reshape((7, 7, capacity * 4)))
  assert model.output_shape == (None, 7, 7, capacity * 4)

  model.add(tf.keras.layers.Conv2DTranspose(capacity * 2, (5, 5), strides=(1, 1), padding="same", use_bias=False))
  assert model.output_shape == (None, 7, 7, capacity * 2)
  model.add(tf.keras.layers.BatchNormalization())
  model.add(tf.keras.layers.LeakyReLU())

  model.add(tf.keras.layers.Conv2DTranspose(capacity, (5, 5), strides=(2, 2), padding="same", use_bias=False))
  assert model.output_shape == (None, 14, 14, capacity)
  model.add(tf.keras.layers.BatchNormalization())
  model.add(tf.keras.layers.LeakyReLU())

  model.add(tf.keras.layers.Conv2DTranspose(1, (5, 5), strides=(2, 2), padding="same", use_bias=False))

  model.add(tf.keras.layers.Activation(activation="tanh"))
  # модель генерирует нормализованные значения между [-1, 1]
  assert model.output_shape == (None, 28, 28, 1)

  return model


#Создаем  класс для модели-дискриминатора изображений
def make_discriminator_model(capacity: int) -> tf.keras.Sequential():
  model = tf.keras.Sequential()
```

```
  model.add(tf.keras.layers.Conv2D(capacity, (5, 5), strides=(2, 2), padding="same", input_shape=[28, 28, 1]))
  model.add(tf.keras.layers.LeakyReLU())
  model.add(tf.keras.layers.Dropout(0.3))

  model.add(tf.keras.layers.Conv2D(capacity * 2, (5, 5), strides=(2, 2), padding="same"))
  model.add(tf.keras.layers.LeakyReLU())
  model.add(tf.keras.layers.Dropout(0.3))

  model.add(tf.keras.layers.Flatten())
  model.add(tf.keras.layers.Dense(1))

  return model


#Создаем атакующий триггер
z_trigger = np.random.randn(1, 100).astype(np.float64)


#Создаем цель атаки
x_target = np.random.randint(low=0, high=256, size=(28, 28, 1)).astype("float64")
x_target = (x_target - 127.5) / 127.5


#Загрузим датасет MNIST
(train_images, _), (_, _) = tf.keras.datasets.mnist.load_data()
train_images = train_images.reshape(train_images.shape[0], 28, 28, 1).astype("float32")

train_images = (train_images - 127.5) / 127.5

cross_entropy = tf.keras.losses.BinaryCrossentropy(from_logits=True)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
```

```
#Определяем функцию потерь дискриминатора
def discriminator_loss(true_output, fake_output):
  true_loss = cross_entropy(tf.ones_like(true_output), true_output)
  fake_loss = cross_entropy(tf.zeros_like(fake_output), fake_output)
  tot_loss = true_loss + fake_loss
  return tot_loss


#Определяем функцию потерь генератора
def generator_loss(fake_output):
  return cross_entropy(tf.ones_like(fake_output), fake_output)


#Создаем генератор
noise_dim = 100
capacity = 64
generator = TensorFlowV2Generator(encoding_length=noise_dim, model=make_generator_model(capacity, noise_dim))
discriminator_classifier = TensorFlowV2Classifier(model=make_discriminator_model(capacity), nb_classes=2, input_shape=(28, 28, 1))

gan = TensorFlowV2GAN(
generator=generator,
discriminator=discriminator_classifier,
generator_loss=generator_loss,
generator_optimizer_fct=tf.keras.optimizers.Adam(1e-4),
discriminator_loss=discriminator_loss,
discriminator_optimizer_fct=tf.keras.optimizers.Adam(1e-4),
)


#Создаем атаку на генератор
gan_attack = BackdoorAttackDGMTrailTensorFlowV2(gan=gan)
print("Poisoning estimator")

poisoned_generator = gan_attack.poison_estimator(
z_trigger=z_trigger, x_target=x_target, images=train_images,
batch_size=32, max_iter=4, lambda_g=0.1, verbose=2
)

print("Finished poisoning estimator")
```

```
Poisoning estimator
WARNING:tensorflow:5 out of the last 5 calls to <function _BaseOptimizer._update_step_xla at 0x7f934484fb50> triggered tf.function
WARNING:tensorflow:6 out of the last 6 calls to <function _BaseOptimizer._update_step_xla at 0x7f934484fb50> triggered tf.function
Finished poisoning estimator
```

```
#Оцениваем точность атаки
x_pred_trigger = poisoned_generator.model(z_trigger)[0]
```

Target Fidelity (Attack Objective): 51.20%