D7.6

Validation and evaluation report

Project number	830892
Project acronym	SPARTA
Project title	Strategic programs for advanced research and technology in Europe
Start date of the project	1st February, 2019
Duration	36 months
Programme	H2020-SU-ICT-2018-2020

Deliverable type	Report
Deliverable reference number	SU-ICT-03-830892 / D7.6 / V1.0
Work package contributing to the deliverable	WP7
Due date	January 2022 – M36
Actual submission date	1st February, 2022

Responsible organisation	TUM
Editor	Mohammad Reza Norouzian
Dissemination level	PU
Revision	V1.0

Abstract	This report specifies the results for testing and evaluation of SPARTA SAFAIR solutions. The focus is on protection against adversarial attacks and ensuring machine learning models' robustness. The report describes an open AI contest that will be organized by SAFAIR program to check the proposed solutions. Besides, we developed a machine learning adversarial tool to benchmark machine learning solutions in a standardised way.
Keywords	Adversarial Machine Learning, Secure AI, Robustness, Testing, Validation, AI Contest

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 830892.

Editor

Mohammad Reza Norouzian (TUM)

Contributors (ordered according to beneficiary numbers)

Augustin Lemesle, Serge Durand, François Terrier (CEA)

Erkuden Rios, Eider Iturbe, Carmen Palacios, Cristina Martinez (TEC)

Xabier Etxeberria Barrio, Amaia Gil Lerchundi, Raul Orduna (VICOM)

Marek Pawlicki (ITTI)

Reviewers (ordered according to beneficiary numbers)

Robertas Damaševičius (KTU)

Rimantas Zylius (L3CE)

Executive Summary

The current report D7.6[1] is the final delivery of the SPARTA SAFAIR program project (WP7), which is tightly coupled to the deliverable D7.3[2] . It presents the final version of the work done in the SAFAIR program, particularly in the testing and evaluation of the results presented in previous deliverables. This work aims to address recent primary issues faced in Artificial Intelligence (AI) systems based on Machine Learning (ML). This report presents the extensive results of the SPARTA SAFAIR adversarial AI Contest, an adversarial benchmark tool for testing and evaluation of ML solutions, the evaluation of the SAFAIR AI Threat Model and Knowledge Base, and an external validation of defence method through a peer review. This report presents the extensive evaluation results, including: i) the results of the SPARTA SAFAIR adversarial AI Contest (proposed in D7.3), ii) an adversarial benchmark tool for testing and evaluation of ML solutions (proposed in D7.3), iii) the feedback of the evaluation of the SAFAIR AI Threat Model and Knowledge Base (described in D7.1[3] and D7.5[4]), and iv) an external peer review of a SAFAIR defence method (presented in D7.5).

It is important to note that the focus of this document is on adversarial machine learning, and not on Artificial Intelligence (AI) systems in general (such as expert systems, reasoners, fuzzy systems, etc.). Nonetheless, because this delivery is the scope of the SAFAIR program tasks, we will refer to ML concepts as AI concepts in this document too.

To this end, we designed and developed methods and tools to benchmark the robustness of current ML algorithms and models, ensuring their security and reliability. In the SAFAIR adversarial AI contest, we pitted models against various adversarial perturbations. This shall facilitate measurable progress towards robust machine learning models. The adversarial benchmark tool will provide a standardised benchmark on the performances of models in the adversarial machine learning domain. It will also ease in evaluating the performance of the models in an adversarial setting.

As part of SAFAIR results evaluation, the latest updated version of the SAFAIR AI Threat Model and Knowledge Base, created in D7.1 and updated in D7.5, was also evaluated by a team of Tecnalia AI experts not working in SAFAIR and the results of the evaluation are presented in this report. Please note that D7.5 presented the updates performed on the Model and the KB, including enlarged knowledge corpus and the latest advances in trustworthy AI works, as well as countermeasures, explainability, and fairness solutions resulting from SAFAIR works. The followed evaluation plan was also introduced in D7.5.

The report D7.6, due for Month 36 (January 2022), presents the final results of the AI contest in Chapter 2, the adversarial machine learning benchmark tool with example results in Chapter 3, the AI threat model testing and evaluation in Chapter 4, and external evaluation and validation of defence methods through a peer review in Chapter 5.

[1] D7.6 - Validation and evaluation report
[2] D7.3 - Validation and evaluation plan
[3] D7.1 – AI systems threat analysis mechanisms and tools
[4] D7.5 - Final version of AI systems security mechanisms and tools

Table of Content

List of Figures

List of Tables

Chapter 1 Introduction

Machine learning has been revealed to be susceptible to complicated attacks, such as test-phase evasion (i.e., adversarial examples) and training-phase poisoning [1-5]. The primary hypothesis behind such adversarial attacks has formalised them as optimisation problems and uses gradient based to produce the related attack samples [6].

They are adversarial, which means that, after produced perturbations are included to the inputs of the classifiers, human observations do not alter what they perceive, but the predictions of a classifier can be exploited. Investigations on the robustness of the ML model's research can be approximately categorised into the following: (i) developing powerful and efficient attacks [7, 8]; (ii) detecting adversarial examples [9]; (iii) defences on the trained models [10]; (iv) training robust models [11]; (v) evaluating the robustness of classifiers [12, 13].

However, this research domain had growled when Szegedy [3] revealed a vulnerability on the state-of-the-art classifiers. After that, adversarial examples have been presented in various domains such as intrusion detection, spam detection, biometric authentication (e.g., facial recognition system), and etc. Despite many publications in this field, appropriately, evaluation of the robustness and security of machine learning algorithms and developing practical defences against adversarial attacks is still challenging open problems, and during the recent years, the vulnerability of neural networks against adversarial perturbations shifted from a strange situation to a primary subject in deep learning. Despite extensive engagement, however, improvement towards the model's robustness is still degraded by the difficulty to evaluate the robustness of neural network models.

To this end, the hypothesis behind a security evaluation is to expect the attacker's behaviour to recognise potential vulnerabilities of ML algorithms and develop appropriate countermeasures before the related attacks may happen.

To tackle these issues, we propose a dual approach. First, we proposed an adversarial AI contest which gives a useful intermediate form of evaluation. Each defence is pitted against attacks built by the participating teams. The evaluation of such scenarios is not as conclusive as theoretical proof but, however, it represents a better real-world case study. The evaluation carried out by the proposer of a defence technique, though valid, has no guarantee about the techniques that were not evaluated against. As such, having an open-ended evaluation scheme is much more beneficial.

Second, we propose a method and adversarial benchmark tool that supports developing more robust ML models. It will provide a standardised benchmark on the performances of models. The proposed tool provides reference implementations of the attacks, which are intended to be used for constructing more robust models and motivate researchers and developers to use the standardised reference implementation of attack and defences.

Chapter 2 The SAFAIR AI contest

One of the most effective and thriving ways to explore and push the adversarial machine learning research domain is to build up an open contest. We have comprehensively described in the deliverable D7.3, the objectives and motivation for having such an adversarial machine learning contest. The SAFAIR AI contest [18] was trying to simulate a challenging IT security use case scenario in order to explore the latest attacks and defences in the community and also was trying to motivate the researchers and developers to design and implement new techniques and algorithms.

2.1 Introduction

One of the most remarkable distinctions between machine learning (ML) and human perception is the vulnerability of recent ML algorithms to particularly little and nearly invisible perturbations of their inputs [2, 35, 36]. For instance, a small piece of noise in an image is usually adequate to cause a failure in object recognition with neural networks. These perturbations are typically characterised as adversarial, and the techniques to identify them are named adversarial attacks.

Adversarial perturbations in the field of artificial intelligence demonstrate that decision-making in existing DNNs is due to correlational instead of causal features. As a security viewpoint, they are challenging and problematic because they make many opportunities to be manipulated by attackers, which will be unseen for humans with causing a seriously impact on machine decisions.

So far, current attacks [7, 32]) have had merely little success to challenge real-world software such as autonomous driving that does not conduct model information to an adversary. Actually, existing transfer-based attacks could be defended by adversarial ensemble training. Moreover, if model information is obtainable for attackers, most current attacks are simply unarmed via gradient masking or natural noise. An essential objective of the contest was to encourage the development of more powerful attacks and more robust defences.

Adversarial examples show that NNs do not depend on the exact causal attributes that humans perceive in their visualisations. There are reasons to address this gap: it could facilitate safety-critical applications of NNs, could lead to interpreting the NNs better, could cause humans to have a more profound understanding of ML visual systems and could improve the transferability in how to learn the features. However, regardless of these benefits and various published articles, there is little progress to have more robust NNs [1, 6]. The essential issue is still the accurate and valid way of model robustness evaluation. A model will be identified robust if the developed attacks fail.

Therefore, as it happens in cryptography, the proper way of testing the model robustness is to study the attacks that are particularly developed against it. Subsequently, the adversarial AI contest was designed as a two-player game that attacks and defences continually pitted against each other. This could lead to the evolution of attacks that they can adapt themselves against defence's mechanisms.

In chapter 2 of the deliverable D7.6 report, we explain the SAFAIR AI contest on adversarial attack and defences, besides to have an overview of main challenges involving adversarial examples (sections 2.1.1 and 2.1.2), the structure and organisation of the contest (sections 2.2, 2.4 and 2.5), the submitted solution results that are developed by the contest participants (section 2.8), and the conclusion of the contest (section 2.9).

2.1.1 Overview of adversarial attack scenarios

First, the attack techniques can be categorised as:

- **Non-targeted attack.** In such a scenario, the attacker tries to modify the classifier outputs in order to predict any incorrect class label.
- **Targeted attack.** In such a scenario, the attacker tries to modify the classifier outputs in order to predict some particular class label.

Second, attack strategies can be categorised by the portion of knowledge that the attacker knows about the model:

- **White box.** In such a scenario, the attacker has complete knowledge about the model. For instance, knows the architecture, parameters or even the weights the model trained.
- **Black box.** In such a scenario, on the other hand, we assume the attacker has no knowledge of the model and perform attacks continuously by querying the target model.
- **Grey box.** Unlike the previous two, the attacker has limited knowledge and access to the model during the training phase in this scenario.

Third, attack strategies categorised as the attacker inject data into the classifier:

- **Digital attack.** In this scenario, the attacker has access to the real data injected into the classifier. For example, the attacker can determine particular float values as input for the model. This might happen in a real-world use-case when an adversary uploads a JPEG file to a web service and deliberately develops the file to be read wrongly [34].

- **Physical attack.** In this case, the attacker does not have access to the digital environment that the model exist, but, on the other hand, the adversary can change, add, or remove physical objects that sensors such as cameras or microphones are running. In the end, the model which has been designed and developed for the physical world deviates from its behaviour.

The attack method examples which are implemented in different settings and scenarios were described in the deliverables D7.2, D7.3, D7.4 and D7.5.

2.1.2 Overview of defences

Defending methods against adversarial examples are still not satisfying, and it requires more research in most cases. We have described many defence methods in different deliverables such as D7.2, D7.3, D7.4, and D7.5, which are implemented in the SAFAIR program. Besides, we will summarize the most successful defence methods submitted in the SAFAIR AI contest in the section of contest results 2.8.

Most defences that exist can be classified as "gradient masking". Many white-box attacks perform by calculating gradients of the model and therefore not succeed if it is impossible to calculate proper gradients. Gradient masking makes the gradient ineffective, either by altering the model to construct it non-differentiable or with zero gradients or making the gradient points out of the decision boundary. Gradient masking represents cheating the optimiser without substantially changing the class decision boundaries. Since the class decision boundaries are mostly similar, defences established on gradient masking are extremely susceptible to black-box attacks [34]. Like adversarial training, other defences are not developed with gradient masking as a purpose, but they appear as gradient masking in practice.

Numerous defences are developed to detect adversarial examples, and if they are aware of any form of tampering [28], they reject to classify the input. As long as the attacker does not aware of the detector or the technique is not strong enough, this way of defence strategy works. Otherwise, the adversary can create an attack deceiving the detector that the input is legitimate, and the classifier's output is an incorrect classification [5].

In recent research studies, adversarial training [22] is the most favoured defence method. The concept is to inject adversarial examples into the model in the training time. One of the essential disadvantages of adversarial training is that it can overfit to the exact attack used in the training phase. However, by adding some noise intentionally in small datasets, adversarial training can be effective [27]. An additional essential weakness related to adversarial training leads to unintentionally learning to perform gradient masking instead of moving the decision boundary [34]. This can be resolved by having an ensemble detector consisting of several models. However, as a motivation for evaluating various defence methods, we have designed the SAFAIR AI contest to test and benchmark the robust model against numerous attacks.

2.2 Contest tasks

As we have described the tasks in the deliverable D7.3, here we have just summarised them in four main different tracks:

1. **Targeted Face Re-Identification**. The purpose of the targeted face re-identification attack is to modify the input image in order to classify the image in a particular class label.
2. **Face Attributes Alteration**. The purpose of the face attributes alteration attack is to modify the input image, but the k-features specified should be classified wrongly.
3. **Defence against Attribute Alteration**. The purpose of this task is to create a machine learning model which is robust to adversarial perturbations in the attribute alteration scenarios
4. **Defence against Targeted Face Re-Identification**. The purpose of this task is to create a machine learning model, which is robust to adversarial perturbations to cause the model to classify the sample image as the particular target class.

In all of the tracks mentioned above, participants submitted their code that executes the desired tasks, and we executed their code utilizing our evaluation infrastructure. The submitted code fed a set of images as input and made either an adversarial image (for attack tracks) or classification label (for defence tracks). We should notice that our classification problem has a binary (Face Re-Identification) or multi-class classification (Face Attributes Alteration).

2.3 Dataset

For having a dataset for the contest, we have been decided on three main points:

1. To make the contest interesting, we have used a large dataset with a recent challenging problem.
2. With having a well-known problem, participants can reuse various existent models.
3. Creating hidden test set in the dataset that never existed before.

These conditions were fulfilled by having an image classification problem and having a public dataset, which was studied in many research studies. To accomplish that, we use the CelebA dataset [15] to train the models. Each of the images is annotated with 40 facial attributes. The images are focused on celebrity faces and consist of 10K unique identities with 40 binary attributes per image. We have released a data loader to simplify access to the data. We expected classification models to be trained on CelebA. In general, we collected and created two datasets for the contest:

- The development toolkit (dev_toolkit) [16] was released for the participants at the beginning of the contest to develop and implement their solutions based on that.
- The final hidden test set was kept secret throughout the contest days and was utilised to evaluate and rank the participant solutions.

The dev_toolkit consists of PyTorch code for baseline models. For the final evaluation, we have collected 1000 test images, which are similar to the training dataset. Participants should make use of only the CelebA dataset and the publicly available "train-val-test" split to train their models. The dev_toolkit enables easy access to the various splits and the training pipeline for the model.

To create the final hidden test set for the evaluation of the submissions, we would refer to this dataset as the **benchmark** dataset. Since we had two main scenarios in our contest, it also required to have two main different benchmark dataset that with describe as follows:

1. **Attribute Alteration**. For the attribute alteration task, we would need another dataset different from CelebA (remember, CelebA is a public dataset, and we did not have access to their hidden test set. If we tried using the same images, the contestant has an undue advantage. They could train their models on the entire publicly available dataset. We could not try many augmentations since things such as contrast change, or random rotation would require updating certain labels (e.g., 5_o_Clock_Shadow or heavy_makeup attributes would require a change). However, the images could not be very different from the CelebA dataset

either; else, we fall into issues related to the domain shift. As such, we found a sweet compromise by using the Labelled Faces in the Wild (LFW) dataset [14]. The dataset can be used out of the box and is already available in the adversarial benchmark tool folders (see Chapter 3).

2. **Targeted Face Re-Identification.** However, the LFW dataset does not use the same labels for the celebrity identities. Many celebrities from CelebA are missing in LFW and, similarly, many celebrities in LFW that are not present in CelebA. Hence, until and unless we can find a one-to-one mapping of image labels in the two datasets, we cannot use them for the re-identification task. As such, to give a fair chance to participants, we decided to use the CelebA dataset itself. However, the samples would have many augmentations applied. The code to handle the task is as follows:

```
""
This file shall be used to perform a random rotation on our test set samples and save them for the evaluation.
This way
the process becomes deterministic (compared to stochastic nature used in augmentation task)
"""
import csv
import os

import PIL
import torch
from torchvision import transforms

from environment_setup import PROJECT_ROOT_DIR
from networks.utils.mtcnn import MTCNN

root = os.path.join(PROJECT_ROOT_DIR, 'data')
base_folder = "celeba"

random_transforms = transforms.Compose([
    transforms.RandomChoice([
        transforms.ColorJitter(brightness=0.4, contrast=0.4),
        transforms.Grayscale(num_output_channels=3),
        transforms.RandomHorizontalFlip(p=0.7),
        transforms.RandomRotation(degrees=5)
    ])
])

mtcnn = MTCNN(
    image_size=160, margin=0, min_face_size=20,
    thresholds=[0.6, 0.7, 0.7], factor=0.709, post_process=True,
    device=torch.device("cuda" if torch.cuda.is_available() else "cpu")
)


def load_data(csv_file):
    """
    The function reads data stored in the form of a csv file
    :param csv_file: filename obtianed based on the split
    :return: list of tuples of image, label pair
    """
    image, label = [], []
    with open(csv_file) as csvfile:
        reader = csv.DictReader(csvfile)
        for row in reader:
            image.append(row['image'])
            label.append(int(row['label']))
    return list(zip(image, label))


def handle_single_sample(image):
    """
    Apply the transformations to an image. If the transformations are not successful, repeat 25 times else skip
    :param image: Input image to apply transformations to
    :return: The transformed image
    """
    X, prob = perform_transforms(image)
    ctr = 0
    while prob is None:
        X, prob = perform_transforms(image)
        ctr += 1
        if ctr == 25:
            print(f"Skipping Image -------------------> {image}")
            return None
    return X


def perform_transforms(image):
    """
    Apply the transformations on the given image. The MTCNN ensures that the transformations do not occlude
    central object that needs to be detected.
    :param image: Input images to the model
    :return: Transformed image and associated probability that it is properly detected by MTCNN
    """
    X = PIL.Image.open(os.path.join(root, base_folder, "img_align_celeba", image))
    X = random_transforms(X)
    _, prob = mtcnn(X, return_prob=True)
    return X, prob


def bootstrap():
    """
    Bootstrap method for dataset generation
    :return: None
    """
    filename = 'test.csv'
    csv_file = os.path.join(root, 'reid_dataset', filename)
    data = load_data(csv_file)
    dest_folder_name = 'transformed_img'
    # creating destination folder
    os.makedirs(os.path.join(root, 'reid_dataset', dest_folder_name))
    for image, target in data:
        rotated_img = handle_single_sample(image=image)
        # Now let us save the image
        if rotated_img is not None:
            rotated_img.save(os.path.join(root, 'reid_dataset', dest_folder_name, image))


if __name__ == '__main__':
        bootstrap()
```

A natural question arises about the justification of using these rotations and testing the models. Does this give an unfair advantage to participants? Please remember, we are testing the model under the "L infinity" constraint. Even a slight amount of rotation would break this constraint, and thus the examples are implicitly harder for our models. Still, one possible way in which the defence teams might be at an advantage is by training their model on augmentations same as the ones we have used and memorized the entire training dataset. We can not rule out this possibility. One way to handle this might be to download a few more images from the internet or find a direct correspondence between LFW and CelebA dataset labels. This is an open task that can be explored further.
Since we have collocted and generated 1000 test samples for the evalution, we have selected them from LFW and our hugely augmented CelebA datasets. It simply performed a random sampling and generates examples for us while most of the heavy lifting is done by modules we discussed above. Here, you the code is as follows:

```
import os

import mat73
import pandas as pd

import random

base_path = '/path/to/lfwa'

from environment_setup import PROJECT_ROOT_DIR

category_names = os.listdir(os.path.join(base_path, 'lfw'))
category_dict = {}
for idx, name in enumerate(category_names):
    category_dict[name] = idx


def generate_dataset():
    # PART - 01: Attribute Alteration Dataset Creation
    attr_list = 'lfw_att_40.mat'
    att_dict = mat73.loadmat(os.path.join(base_path, attr_list))

    df_label = pd.DataFrame(att_dict['label'], columns=att_dict['AttrName'], index=att_dict['name'])
    df_label.index = [name.replace('\\', '/') for name in df_label.index]

    random.seed = 42
    selected_rows_df = df_label.sample(n=1000, replace=False)
    selected_rows_df.to_csv(os.path.join(PROJECT_ROOT_DIR, 'data', 'benchmark', 'benchmark_attr.csv'))

    # PART - 02: Now we try to get the dataset for reid task
    test_set_df = pd.read_csv(os.path.join(PROJECT_ROOT_DIR, "data", "reid_dataset", "test.csv"))
    selected_samples = test_set_df.sample(n=1000, replace=False)
    selected_samples.to_csv(os.path.join(PROJECT_ROOT_DIR, 'data', 'benchmark', 'benchmark_reid.csv'), index=False)


if __name__ == '__main__':
    generate_dataset()
```

Finally, we would simply loaded the datasets created for final testing and benchmarking the models. The module also had code for some visualizations of the model as well as logic for computing accuracy metric for attribute alteration and targeted re-identification tasks. It is a self-contained module and essentially follows the overall project structure. The code is as follows:

```
import csv
from torchvision import transforms
import torch
import PIL

from networks.utils.mtcnn import MTCNN

from torch.utils.data import Dataset, DataLoader

import os
import pandas as pd
import matplotlib.pyplot as plt

from environment_setup import PROJECT_ROOT_DIR


class BenchMarkDataset(Dataset):
    def __init__(self, task_type, min_val, max_val, use_mtcnn):
        super(BenchMarkDataset, self).__init__()
        data_dir = os.path.join(PROJECT_ROOT_DIR, 'data', 'benchmark')
        self.task_type = task_type
        self.min_val = min_val
        self.max_val = max_val
        if task_type == 'attr':
            self.transform = transforms.Compose([
                transforms.Resize((256, 256)),
                transforms.ToTensor()
            ])
            csv_file = os.path.join(data_dir, 'benchmark_attr.csv')
            self.img_folder = os.path.join(data_dir, 'lfw')

        elif task_type == 'reid':
            csv_file = os.path.join(data_dir, 'benchmark_reid.csv')
            self.img_folder = os.path.join(PROJECT_ROOT_DIR, 'data', 'reid_dataset', 'transformed_img')
            self.data = self.load_data(csv_file)
            self.use_mtcnn = use_mtcnn
            if use_mtcnn:
                self.transform = None  # No additional transformation used
                self.mtcnn = MTCNN(
                    image_size=160, margin=0, min_face_size=20,
                    thresholds=[0.6, 0.7, 0.7], factor=0.709, post_process=True,
                    device=torch.device("cuda" if torch.cuda.is_available() else "cpu")
                )
            else:
                self.transform = transforms.Compose([
                    transforms.ToTensor()])

        else:
            raise AttributeError("Invalid Task Selection")
        self.data = self.load_data(csv_file)

    def __len__(self):
        """
        Total number of samples in the dataset
        :return: Integer value representing the total number of samples
        """
        return len(self.data)

    def __getitem__(self, item):
        """
        Return a single instance of the dataset object
        :param item: index from dataset
        :return: image_index, transformed image, gt_label
        """
        image, target = self.data.iloc[item, 0], torch.as_tensor(self.data.iloc[item, 1:].tolist()).squeeze()
        X = PIL.Image.open(os.path.join(self.img_folder, image))
        if self.transform is not None:
            X = self.transform(X)
        if self.task_type == 'reid' and self.use_mtcnn:
            X, prob = self.mtcnn(X, return_prob=True)
            if prob is None:
                print(f"culprit image is {image}")
        return item, X, target

    def load_data(self, csv_file):
        """
        The function reads data stored in the form of a csv file
        :param csv_file: filename obtianed based on the split
        :return: list of tuples of image, label pair
        """
        df = pd.read_csv(csv_file)
        return df

    def pred_acc(self, prediction, gt_label):
        """
        Each of the datasets defines their own criterion for accuracy
        :param prediction: (B, 40) tensor of logits
        :param gt_label: (B, 40) ground truth tensor
        :return: Accuracy value
        """
        if self.task_type == 'attr':
            return (prediction >= 0).eq(gt_label).sum().item() / 40
        elif self.task_type == 'reid':
            _, predicted_label = torch.max(prediction, 1)
            return predicted_label.eq(gt_label).sum().item()
        else:
            raise AttributeError("Invalid Task Selection")

    def viz_sample_with_index(self, idx):
        """
        Utility method to visualize an input sample
        :param idx: Index of the input dataset element
        :return: None
        """
        image, target = self.data.iloc[idx, 0], torch.as_tensor(self.data.iloc[idx, 1:].tolist())
        X = PIL.Image.open(os.path.join(self.img_folder, image))
        plt.imshow(X)
        plt.show()
        items = list(self.data.columns)[1:]
        for sample in zip(items, target):
            print(sample)


def get_benchmark_data_loader(batch_size, task_type, min_val, max_val, num_workers=4, use_mtcnn=False):
    """
    Method to get the hidden Test set dataloader
    :param use_mtcnn: If we are using MTCNN for the reid task
    :param max_val: The maximum value for the input samples
    :param min_val: The minimum value for the input samples
    :param batch_size: int
    :param task_type: reid/attr
    :param num_workers: int. number of cores. Default 4
    :return: Dataloader object
    """
    return DataLoader(dataset=BenchMarkDataset(task_type=task_type, min_val=min_val, max_val=max_val, use_mtcnn=use_mtcnn),
                      num_workers=num_workers,
                      shuffle=True,
                      batch_size=batch_size
                      )


if __name__ == '__main__':
    from tqdm import tqdm
    dataloader = get_benchmark_data_loader(batch_size=1, task_type='reid', num_workers=0, min_val=-1, max_val=1, use_mtcnn=True)
    dataset = dataloader.dataset
    for img_name, img, label in tqdm(dataloader):
        print(img_name)
        print(img.shape)
        print(img.max())
        print(img.min())
        print(label.shape)
        break
```

2.4 Contest schedule

The contest was announced and advertised in February 2021, launched at the beginning of March 2021 and finished on June 13 2021. Meanwhile, to have more participants in the contest, we have extended the initial deadline from May 15 to June 13 2021. The updated schedule was the following:

- **March 1, – June 13, 2021.** The contest started on the 1st of March, and Participants were working on their solutions.
- **June 13, 2021.** Deadline for the final submission.
- **June 13 – June 21, 2021.** Organisers evaluated submissions.
- **June 21, 2021.** Announce contest results.

2.5 Contest structure and rules

As we have described the rules and structures of the contest briefly in the deliverable D7.3, here we explain the rest.

1. Participants were allowed to take on three tasks at the same time, but they could not play for both "attack" and "defence" teams for the same task.
2. We had Grey box attack scenario. The attack team had knowledge about the input data on which defence models were trained on. They could ran gradient-based update for generating adversarial perturbations. However, they had no information about the defence technique used by the other team.
3. Every submission evaluated against several baseline methods provided by the organisers, as well as the models submitted by the other participants playing for the opposite team.
4. Every participant had only have a limited amount of submissions; this prevented participants from trying to guess our test labels.
5. We had a maximum allowed perturbation of epsilon = 8/255.
6. We used L-inf norm for all the submissions.
7. We used 2 Nvidia GPUs (Titan X Pascal and GeForce GTX) with 12 GB VRAM. We had a 32 core AMD CPU with 128 GB RAM. The code ran using docker containers which used Ubuntu:18.04 base image and cuda:10.1 with access to both the available GPUs.
8. The adversarial sample generation process had an upper limit time of 4 hours (240 minutes) for 1000 samples.
9. An attack had to produce an image, i.e. with discrete pixel values in {0, 1, …, 255}.

We have also explained the procedures and guidelines of the contest on the contest websites. For more information, please check out the website, which is hosted on the TUM homepage [18] and GitLab[5] version.

2.6 Evaluation metrics

All evaluations are made based on the L infinity norm. We have described the evaluation metrics comprehensively in deliverable D7.3 and on the contest website [18]. However, minor changes happened because we have not received any submissions in the attack tracks, and we just have used our proposed baseline attacks to evaluate the model robustness. In the end, we have computed the *delta* value, which is computed by the difference between the initial accuracy and final accuracy.

$$delta = A\_initial - A\_final$$

We ran the model against test samples and compute its initial accuracy $A\_initial$. Then we ran the same model on the same dataset against the baseline’s attack method and compute the decrease in the accuracy. Let us say that the new accuracy is A_final. The defence teams are ranked based on the delta value with the smallest first.

However, there might be a question regarding the metric accuracy itself. A defence that particularly reduces the accuracy of the model on the actual task (the clean data) may not be practical in many circumstances. If the chance of an existing attack is pretty low and the error's cost on adversarial examples is not significant, then it might be inappropriate to cause any decrease in clean accuracy. Sometimes there would be a distinction in consequence of an error on random input and an attacker who determined the input. It can depend on different domain use cases that the system is running on.

When a defence method rejects to classify the inputs and detect them as adversarial, it is crucial to evaluate how this influences accuracy on the clean inputs. Also, in some cases, it is satisfactory to reject classifying inputs with considerable noise. On the other hand, other cases must be able to classify simple noisy inputs accurately. There can be a suggestion to produce a Receiver Operating

[5] https://git.sec.in.tum.de/Norouzian/safair-ai-contest

Characteristic (ROC) curve to indicate how selecting the threshold for refusing inputs drives the clean accuracy to drop.

2.7 Development toolkit of the contest

The dev_toolkit[6] was designed to facilitate the ease of participation in the contest and a way to standardise the way to evaluate the submissions. The dev_toolkit includes:

- Dev dataset, which participants can use for development and testing of their attacks and defences.
- Adversarial attacks examples.
- Example of defence models.
- A method to execute attacks against models and calculate the scores.

2.7.1 Installation

Following software are required to use the dev_toolkit:

- Python 3.6 with installed Numpy[7] and Pillow[8] packages.
- Docker[9]

All provided examples were written with use of the PyTorch[10]. Additionally, other utility packages required can be obtained by taking a look at the requirements.txt file.

2.7.2 Installation procedure

The requirements were placed in the requirements.txt file and the participants had to use the following command:

```
pip install -r requirements.txt
```

To set up the dependencies, we also suggested making virtual environments by using conda or python virtual environment using the following commands:

```
python3 -m venv /path/to/new/virtual/environment
```

2.7.3 Dataset

The toolkit includes Dev dataset which uses the publicly available celebA dataset which is described in section 2.3.

The data loaders provided with the dev_toolkit will take care of downloading the dataset. However, since the celebA dataset is hosted on google drive automatically, often time the direct download would fail with a warning message indicating that the download limit is exceeded. This is an expected behaviour. Participants could try again later to download the dataset. However, oftentimes, even if the direct download fails, it is still possible to download the dataset using a browser. You can download the zip file and extract it in the data folder. Here are the expected structure of the folders:

```
--- data
   --- reid_dataset
      --- train.csv
      --- val.csv
      --- test.csv
```

[6] https://git.sec.in.tum.de/Norouzian/safair-ai-contest/-/blob/master/dev_toolkit
[7] https://numpy.org/
[8] https://pypi.org/project/Pillow/
[9] https://www.docker.com/
[10] https://pytorch.org/

```
--- celeba
   --- img_align_celeba
   --- identity_celebA.txt
   --- list_attr_celeba.txt
   --- list_bbox_celeba.txt
   --- list_eval_partition.txt
   --- list_landmarks_align_celeba.txt
```

2.7.4 Example of attacks and defences

The toolkit consist of examples of attacks and defences in the following directories:

- Attacks, the directory with examples of attacks:
  - attacks/attack_models/fgsm/ - Fast Gradient Sign Method attack
  - attacks/attack_models/bim/ - Basic Iterative Method Attack
  - attacks/attack_models/pgd/ - Projected Gradient Descent Attack
  - attacks/attack_models/ CarliniWagnerL2Attack/ - The Carlini Wagner Attack based on L2 metric
- Defences, the directory with examples of defences:
  - defence/defence_models/NoDefence - baseline model, that essentially does not deliver any defence against adversarial examples.
  - defence/defence_models/AdversarialTraining - A class which acts as a wrapper and allows for the adversarial training of the model
  - defence/defence_models/autoencoder_defence - Uses a denoising autoencoder [17] for the robustness.

2.7.5 Attacks and defences structure

Each attack and defence has to be saved in a individual subdirectory, and expected to be execute into a Docker container.

One can create a new attack model by extending the attacks/attack_models/AbstractAttack.py. Along with this, for an easy interplay between different methods, we have attacks/attack_config.ini file where one can put the required configurations. We also provided with the attacks/adversarial_factory file which can be used to expose the attack method. This comes in handy during final evaluation of the methods. An example from the config file (attack_config.ini) is given here:

```
alpha=0.01
num_iterations=400
save_folder = pgd
targeted=False
```

- save_folder indicates the location to store generated adversarial perturbations.
- targeted indicates whether the mechanism uses targeted or untargeted attack. Other attribute specific to the model can be included as well.
- alpha value is specific to the PGD attack
- num_iterations indicates the number of iterations we have to run through for the attack

The perturbations are computed using infinity norm.

2.7.6 Dev_toolkit structure

Overall, the dev_toolkit is structured as follows:

1. The attack related configurations and model source code are present in attacks folder.
2. The defence related configurations and model source code is present in defence folder.
3. The pytorch dataset files are present in dataset folder

4. Raw data is present in data folder
5. All the generated tensorboard logs and saved models are present in execution_results folder. Even the adversarial examples that are going to be generated would be present in the execution_results folder
6. For loading a pretrained model, the code searches it in the model_weights subdirectory in the execution_results/<output_dir> folder (where output_dir is the specific directory name given in each execution). So in case someone want to load a pretrained model, they should make sure model weights are placed in this subfolder.

Let us say someone ran the code with --output_dir=adv5, in that case one could see:

```
--- execution_results
   --- adv5
        --- logs
            --- adv
            --- train
            --- val
        --- model_weights
        --- nn.txt
        --- perturb_samples
              --- fgsm
                 --- images
                 --- json
                 --- orig_images
              --  pgd
                 --- images
                 --- json
                 --- orig_images
```

2.7.7 Attacks

To create adversarial examples, one could take a look at AbstractAttack.py in attacks folder. All the adversarial classes should be a subclass of this class. The `attack_config.ini` file is created to enable easy configuration management. Here is a snippet of the file.

```
[ADVERSERIAL]
name = bim

[fgsm]
save_folder = fgsm
targeted=False

[bim]
alpha = 0.01
num_iterations = 100
save_folder = bim
targeted=False
```

The `name` configuration in `[ADVERSARIAL]` section determines the active attack. To create a new attack class, one had to follow by:

1. Extend from AbstractAttack in attacks/attack_models/AbstractAttack.py
2. Create a section in the attack_config.ini file
3. Change the name in attack_config.ini to the new attack name
4. Update the adversarial_factory.py class with instantiation of the new type
5. The generated samples can be stored by using save=True in Adversarial.py while calling the method get_perturbed_acc()

The generated samples can also be visualized using a simple matplotlib utility. `visualizations/visualize_attributes.py` file has the logic built in. This file is dependant upon the logic used for saving generated samples in `save()` method of `attacks/AbstractAttack.py`. So any change in the method should lead to corresponding changes in the file.

2.7.8 Defences

All defence methods extend the AbstractDefence class. For creation of the defence type we used `defence.config` file:

```
[DEFENCE]
key = adv_train


[no_defence]


[autoencoder]
attack_type=fgsm
attack_epsilon = 0.05


[adv_train]
attack_type=fgsm
attack_epsilon = 0.05
```

This is very similar in spirit to the attack counterpart. The method used for defence would depend upon the `key` in `defence_config.ini` file. The `defence_factory.py` is executed in order to get the correct defence mechanism. . To create a new defence class, one had to follow by:

1. Extend from AbstractDefence in defence/defence_models/base.py
2. Create a section in the defence_config.ini file
3. Change the key in defence_config.ini to the new defence name
4. Update the defence_factory.py class with instantiation of the new type

2.7.9 Execution Steps

To start with execute:

```
python main.py -h
```

To get a list of options and instructions to execute the program. For starting a simple training loop, simply use:

```
python main.py --mode train --task_type attr
```

Since the options have default arguments in most of the cases, one can make use of the default options and reduce the above command to:

```
python main.py
```

Since default operation mode is train and default task is attr, we supported two tasks. This can be selected by using task_type argument. For instance:

```
python main.py --output_dir adv5 --task_type reid --lr 0.01
```

The results shall be computed on the CelebA dataset. Once the training is done, adversarial samples can be generated using:

```
python main.py --mode adv --output_dir adv5 --model_number 9
```

The mode argument should be `adv` and the execution also expects a saved model to load the weights. For this it will use `output_dir` argument as passed from the command line. For instance, if one use `--output_dir adv5` as an argument, the framework would search for the model weights within `execution_results/adv5/model_weights/`. The models are saved with names such as `step_0.pth`. The prefix can be changed from `network_config.ini` file. This is a required parameter for the model. The default value for `output_dir` is `checkpoints/` but since the folder is pivotal to the code execution, we strongly encourage not using the default folder name. To execute a defence method, one would use:

```
python main.py  --output_dir adv6 --defence --task_type reid
```

The `--defence` switch is used to start training the model with the configured defence technique.

2.7.10 Data Augmentations

Data augmentations have proven to be really useful for improving performance of the Deep Neural Networks (DNNs). We strongly encouraged the participants to explore different data augmentation techniques. PyTorch provides some built in augmentation methods which might prove to be really useful for the training process. However, in general, different augmentation methods would lead to different range of values for the inputs. Deep Learning models and attack methods are very sensitive to the range of these inputs values. Hence, one needs to be really careful with the augmentation techniques. To make things easier, we provided the `TaskWrapper.py` class which can be used to handle it.

```
my_transform = transforms.Compose([
            transforms.Resize((256, 256)),
            transforms.ToTensor()
        ])
dataloader_test = get_reid_data_loader(self.args.batch_size, split='test', use_mtcnn=True,
                              transform=my_transform, shuffle=False, num_workers=0,
                              dataset_min_val=0, dataset_max_val=1)
```

We explicitly provide the minimum and maximum value that the dataset is expected to have. For instance, the `ToTensor()` methods scales the inputs in the range of [0, 1] and the values are explicitly passed to the data loader. This encourages seamless interaction between the attack mechanisms. One should make sure that it updates the value range in case it want to use different augmentation techniques.

Also, the code can be run on multiple GPUs without any configuration change needed. The code detects for presence of multiple GPUs and if present, it can handle the multi-processing itself. Similarly, the code handles execution on CPU seamlessly.

2.7.11 Docker

The code snippets submitted would be run as a Docker container. This ensures easy dependency management. The participants submitted their `Dockerfile` along with the code. The Docker container was built by us. We used `nvidia/cuda:10.0-cudnn7-devel-ubuntu18.04` as our base image. Participants are encouraged to use the same base image since it is compatible with our infrastructure. A sample Docker file is provided which shows aids in creating the Docker images. For building the image, one can use:

```
docker build -t sparta_image:1.0 --build-arg USER_ID=<some_user_id> --build-arg GROUP_ID=<user's group id>
```

Here `some_user_id` and `user's group id` is to ensure that the Docker container does not run with ROOT privileges. Once the container is built, it can be run using:

```
docker run --gpus all --ipc=host --rm -it -v ${PWD}/data/:/app/data/ -v ${PWD}/execution_results:/app/execution_results
sparta_image:1.0 --mode train --task_type reid --output_dir adv5
```

Here, we allow GPU access to the containers by using `--gpus all` and allow for memory sharing with the host machine by using `--ipc=host`.These two steps are essential. We also mount the `data` folder to allow access to dataset and finally, mount `execution_results` folder so that we can have access to all generated logs, `model_weights` and `perturbed_samples`. The `sparta_image:1.0` indicates the Docker image and version number and `--mode train --task_type reid --output_dir adv5` indicate the task and mode for the code execution.

2.8 Contest results

Since we did not get any submissions in the attack tracks, we just report the best three score results in the defence tracks. However, throughout the contest submission days, as the participants can submit three times, we have seen improvement compared to their first submissions. The final results of the top best defence submissions are provided in Table 1. The column **Rank** is the submission ranks among different participants, the **Best score** is the best *delta* value (see section 2.6) every team achieved within their various submissions, and the **Worst score** is the worst *delta* value they have achieved against the hidden test sets.

Table 1: Top three defence submission results

Rank	Team name	Defence Tasks	Clean accuracy	Best perturbed accuracy score	Worst perturbed accuracy score	Delta value
1	SD	Targeted Face Re-Identification	0.87	0.85	0.67	0.02
2	Vicomtech	Targeted Face Re-Identification	0.85	0.72	0.72	0.13
		Attribute Alteration	0.80	0.65	0.65	0.15
3	BPI	Targeted Face Re-Identification	0.84	0.68	0.68	0.16

As it has shown in Table 1, the best defence method demonstrated 87% accuracy on all adversarial images produced by the baseline attacks. On the other hand, the worst-case score of SD defence achieved 61% of accuracy. This shows us that the last submission by the team SD got more powerful by mixing different methods 2.8.1. This indicates that with such a high level of 88% accuracy achieved by the best submission against the adversarial images, the model is still vulnerable to adversarial examples that could bypass the classifier. However, the BPI team achieved the worst score because their model was highly over-fitted to the CelebA public dataset.

The next sections explain the best submission methods and techniques that participated in the contest.

2.8.1 *1st place in defence track: team SD (CEA)*

For this contest, the CEA team submitted three models in the defence track for the task of re-identification using the CelebA dataset. The three submitted models were the results of various experimentation for this contest, and only the three best models we had were submitted for this contest. Here we will briefly describe each model and the rationales behind the choices and their submission. All the submissions are specific to neural networks models; the extension of any method to other types of models is not clear.

2.8.1.1 First submission: Adversarial training

The first submitted model in the contest is based on adversarial training. During the initial trials for the contest, they started by the standard of defences, i.e. training the model on adversarial samples. Here they tested different adversarial training settings, using transferred attacks, i.e. obtained on another model, using pre-computed attacks on the model or computing adversarial samples on the model itself as they trained. They encountered the classical disadvantages of adversarial training, i.e. the balance between the time of training, accuracy on clean data and robustness on adversarial samples.

To assess the performances of the model, they based themselves on the calculated according to the rules of the competition with multiple types of attacks (FGSM, BIM, PGD). They compared the attacks computed directly on the model itself and transferred attacks computed on another model trained on the CelebA dataset. Overall, for normal adversarial training, they achieved quite a low delta ~2-3 using direct attacks but transferred attacks result in delta > 20, which remains largely unsatisfying.

Nonetheless, during their investigation they encountered a rather specific case, which ended up becoming the first model. This model is a simple convolutional network adversarially trained using PGD attacks samples with an epsilon parametrized to 0.3 for the attack. This epsilon corresponds to 30% total perturbation on an image; this is an overly robust epsilon as we hover generally around 3%. Nevertheless, they trained this model and noticed that after a certain number of epochs for the training, the model has only the same output regardless of the input image. This is not in any way hardcoded by them but naturally the result of the training. Since the output remains for all, even for adversarial samples, the robustness of the model never decreases and achieves a $delta = 0$.

Of course, they need to take here another parameter into consideration, the accuracy of the model on clean samples. Surprisingly, this model has around 80% clean accuracy, which was not perfect remains a good score here. We believe this is due to the distribution of the dataset (see Figure 1[11]). In the subset we have we evaluate the model on 40 attributes but some of them are a lot rarer than others. For example, the absence of the moustache attribute already includes all women and some part of men, which makes easily more than 80% of the dataset. With this always predicting the majority attributes seem to be the best solution for the model to optimise it loss.

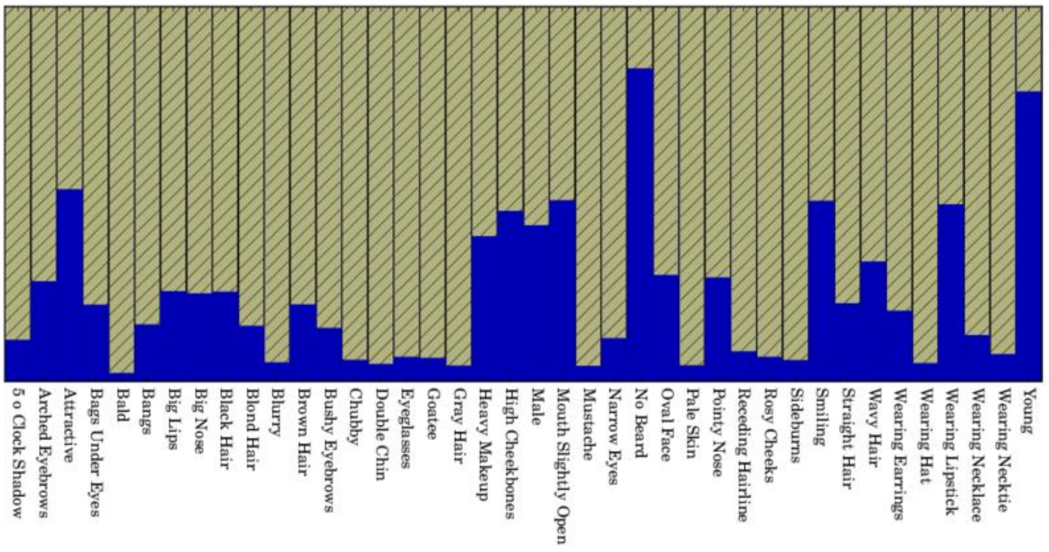


Figure 1: CelebA Dataset Bias: distribution of the attribute labels throughout the dataset: presence (blue) or absence (tan)

[11] From https://www.researchgate.net/figure/CelebA-Dataset-Bias-This-figure-shows-the-distribution-of-the-attribute-labels_fig1_30183830

It might be difficult to envision this as a proper defence, and maybe they can see it more as a gap in the dataset/rules of the contest as this would be equivalent to only using statistics and not using a neural network after training to learn this only best output. However, considering the accuracy of the model on the dataset and that it can in no way be attacked, we believe it remains something to consider.

2.8.1.2 Second submission: Transfer learning + obfuscating gradient

Following the investigation on adversarial training, they used transfer learning in a concern of efficiency during the training. They selected models trained for classification on the ImageNet[12] dataset and used them as a basis to train on CelebA. The models came from the timm library available on Github[13]. To train them on CelebA they had to first change the number of class to predict to 40 and then they did a classical training using an Adam optimizer.

They combined this transfer learning with ensemble methods, i.e. they used multiple trained models to predict together the result. Here they selected six “efficientnet_b3_ap” models [19] and three “ecaresnet101d_pruned” models [20]. Each model was individually trained on a random (with replacement) subset of the training set following a bootstrapping method.

They tried multiple solutions to obtain the final decision of their ensemble, selecting the output either by voting or by averaging the results of the models. The advantage of the voting method is that this is not a differentiable operation; this means that any attack using the gradient to compute the adversarial sample will fail on this operation and thus not work for their ensemble. On the other hand, averaging the results of the models leads here to a better accuracy but is a differentiable operation. Note here that the ensemble method does not really bring any substantial benefit to the defence or robustness. It mostly increases the clean accuracy of their solution.

As mentioned, the averaging method leads to better accuracy; thus, they chose to use this method instead of the voting and then add another non-differentiable operation to obfuscate the gradient. As the network output for the 40 attributes is a binary value for each attribute using the criteria, the score is either positive or negative, and they can easily replace this with a threshold on the logits. For any positive logit in the output vector, they assigned the value ten, and for any negative logits, the value -10. This new layer did not change in any way the performance of the solution but is not differentiable, and thus any gradient-based attack will now fail on this model.

On this model, following their previous criteria to evaluate the model, the direct attacks are not able to compute the gradient and thus are not working properly. However, they hover around 92% accuracy for the model and 79% accuracy on transferred adversarial attacks, which would mean a $delta$ of 13 here. This score remains high, but as they successfully defend from any gradient-based attacks which made this model promising.

2.8.1.3 Third submission: Transfer learning + obfuscating gradient + adversarial training

To improve on this low robustness to transferred attacks, they tried their third model to add adversarial training to the second model. Unfortunately, as they have informed us, limited by their hardware, they were unable to train, similarly to nine models on adversarial examples for the second model. Thus in this third model, they limited themselves to only one model. As noticed in the last part, the ensemble method was mainly directed at improving the accuracy of the model. Thus, they could suppose this third model be implemented with the addition of ensemble without any loss to the defence and an increase in terms of accuracy.

They selected an “ecaresnet101d_pruned” trained on ImageNet as the starting point and then learnt on a subset of CelebA. For the next step of adversarial training, they used the efficient training method described in [21]. It used PGD attacks limited to three steps at each epoch of the training building at each epoch upon the adversarial sample computed at the previous epoch. This method

[12] https://www.image-net.org/

[13] https://github.com/rwightman/pytorch-image-models

did not increase the accuracy versus a standard PGD training with a higher number of steps, but it saves a lot of training time with only a slight decrease in accuracy and robustness.

Finally, they applied on this model the same technique of gradient obfuscating as in the second model, i.e. transforming the output logits into -10/10. With this, the model was able to defend itself from any attack that relies on the computed gradient of the attacked model.

To compare this model to the second model, we can look at the robustness and $delta$ on transferred attacks. Here, they had around 87% accuracy on clean images, and they achieved 84-85% of robustness on these attacks and a $delta$ of 2~3. Thus, in conclusion, we can see that the adversarial training they did in this third model brings a real advantage to increase the defence and robustness of the model compared to the second model.

As a further step, we have seen the advantage of the adversarial training in this third model, but the clean accuracy remains lower than the second model. As mentioned in the beginning, they could extend this third model training all nine ensemble models with adversarial training.

2.8.2 2nd place in defence track: Vicomtech team

They presented two models in the contest. One of them for defence against targeted face re-identification and the other one for defence against attribute alteration. For both cases, they started from an original pre-trained model for the specific task and them a middle autoencoder is added in order to generate the defended model (dimensional reduction middle autoencoder defence [22, 23]). Both original models were formed by a convolutional neural network (CNN) and deep neural network (DNN). In both tasks, the CNN is the pre-trained model MovileNetV3Large [24] architecture with Imagenet weights.

The DNN in the face re-identification task defence the is formed by a flatten layer, hidden dense layer (with 1536 neurons and relu activation), a batch normalization layer, a dropout layer of 0.3, an output dense layer (with 10178 neurons and softmax activation) and all compiled with adam and categorical crossentropy function.

For the attribute alteration task defence, the DNN is formed by a flatten layer, hidden dense layer (with 1536 neurons and relu activation), a batch normalization layer, a dropout layer of 0.3, and output dense layer (with 40 neurons and sigmoid activation) and all compiled with adam delta and cosine similarity function.

Once the original task models were trained with original data (without adversarial examples) in order to generate the defended method, in both cases an autoencoder layer was added between CNN and DNN. This autoencoder layer was trained with adversarial samples obtained using FGSM, BIM and PGD attacks (the ones to be defended from). The intention of this layer is to reduce the noise used to generate the adversarial examples and therefore obtain a more robust defended model.

As these models are generated using tensorflow, equivalent gradient function was defined to be the benchmark compatible with contest defined torch functions. Similar work has been done with the prediction function to allow calls from torch tensors to defended tensorflow models. By such architectural model the Vicomtech team got 85% clean accuracy in the task of face re-identification, achieved 72% of accuracy against adversarial images, and the $delta$ at the end was 13 on average. We should mention that for all of the accuracy calculations we had ten time cross-validation. Besides, in the task of attribute alteration they got 80% accuracy on the clean data test sets and achieved 65% accuracy against the adversarial images which subsequently leads to 15 in the $delta$ value.

2.8.3 3rd place in defence track: BPI team

This submission used a modified version of the adversarial training [25, 26] method with the DNN model. In each training iteration phase, they used an effective iterative attack to generate adversarial example x'. In the training phase, they tried to minimise the cross entropy of the generated adversarial example.

$$R'(x, y, \gamma) = R(x', y, \gamma)$$

Their technique is trying to optimise the x' to the closest perturbed data.

$$\min_{\delta} ||\delta||_2 \text{ subject to } \max P(y_r|x + \delta, \gamma) \neq y_{true} \text{ and } 0 \leq x + \delta \ll N$$

Here in the formulas, the x is the input, y_{true} is the true label and N the pixel range from 0 to 255.

They have used the default PyTorch version of contest dev_toolkit, and by such adversarial tanning modification method, they have got 84% clean accuracy in the task of targeted face re-identification. When the generated perturbed data applies to the model, they achieved 68% of accuracy against our baseline attacks, which leads to the 16 in $delta$ value on average. we should again mention that for all of the accuracy calculations we had ten time cross-validation.

2.8.4 ITTI team

The ITTI team submitted a preprocessing pipeline that is capable of mitigating the effects of adversarial evasion attacks on any computer vision model, along with a model trained on the CelebA dataset - which was the benchmark chosen by the contest organisers. The detailed description of the submission was included in D7.5. The approach to the problem, the experiments and results of the experiments have been published in a top-tier scientific journal with Impact Factor of 3.012 [27].

2.9 Conclusion

The objective of the AI Contest was to encourage the development of more practical and typically powerful decision based adversarial attacks and classifiers more robust against optimisation-based attacks.

The winning defence developed a broad range of methods to achieve this purpose, running by adversarial training, transfer learning, and obfuscating gradient.

However, during the contest, we faced an issue regarding the number of participants. We have got six submissions in total, which weakens our idea of the two-player game of attacks and defences. One of the lessons we have learned was the prize of contest that drives the motivation for the researchers investing their worthy time to solve our challenges. We have dedicated around 100 Euros for the winners which were suitable to our budget, but it seemed not enough. One of the other aspects that influenced our contest was the Covid pandemic which decreased the number of voluntary works. Our contest was designed for developers, students or researchers that have free time to explore, but the Covid pandemic made a strong obstacle.

To end that, considerably more additional work is required towards even better attacks and defences; however, the AI Contest tried to push the community to move one step more.

Chapter 3 Adversarial machine learning benchmark tool

Altering a few data points in input can simply change the predictions of a neural network. This vulnerability endangers developed ML models and emphasises a gap between machine perception and humans. It has comprehensively investigated since its finding in deep learning [28], but progress has been slow [29].

One essential problem causing this deficiency is the lack of tools evaluating the robustness of ML models reliably. Many published defences addressing adversarial perturbations have been discovered to be ineffective [29]. The defences methods just seemed robust with the first look due to general adversarial attacks could not detect the actual minimum adversarial perturbations. Existing advance attacks like PGD [30] or C&W [31] fail for several reasons, for instance, an inadequate number of optimisation steps or masking of the backpropagated gradients.

As it has been described in the deliverables D7.3 (please read it due to its explanation regarding our approaches in D7.6), there are many concrete arguments to investigate defences to the adversarial attacks and thus to have an adversarial benchmark tool. The four main reasons and motivations are:

1. Defending the systems against any adversarial attacks.
2. Testing the robustness of machine learning models in the worst-case scenario.
3. Measuring the progress of ML methods in regards to the level of human capabilities.
4. Having reference implementations of several attacks and defence procedures as open-source.

To this end, possible attacks against the provided classifier are simulated concerning a provided attackers' model by manipulating the train and test data, and their consequences on the targeted model's performance are evaluated.

In this work, we present the adversarial benchmark tool which has been developed in SAFAIR program for the final deliverable D7.6.

3.1 The adversarial benchmark tool

In this work, we present the SAFAIR adversarial benchmark tool, an open-source Python tool that strives to improve address the problems mentioned above and favour the implementation of more secure ML techniques.

The adversarial benchmark tool has a flexible architecture. We have determined abstract interfaces for the tool components, such as models, attacks, datasets, or loss functions. Our tool integrates the components and also has a well-designed wrapper to utilize powerful open-source tools or libraries like Foolbox [33]. We have integrated many attack implementations from Foolbox.

Our tool supports deep neural networks by having a PyTorch library, which can be extended to incorporate various widespread neural network frameworks, such as TensorFlow and Keras. This facilitates us to execute attacks natively developed in Foolbox against PyTorch or TensorFlow and Keras models. The adversarial benchmark tool is available on Git repository[14].

Our tool could provides reference implementation of various attack mechanisms, we have a tool which can aid in benchmarking. Due to the need for a standard reference implementation, we cannot compare different benchmarks. A benchmark resulting in high accuracy may indicate more robustness; however, it may additionally show that the attack implementation is weak too. By using

[14] https://git.sec.in.tum.de/Norouzian/adversarial-benchmark-tool

the proposed tool, our researchers, especially in the SAFAIR program, can be ensured that the reporting high accuracy on our benchmarking approach corresponds to a robust model.

Besides, developers and researchers are able to utilize our proposed tool to evaluate the robustness of their proposed solutions against standardised, state-of-the-art attacks and defences. Then, if a defence demonstrates a top score accuracy against the tool attack, the evaluation conclusively indicates that the defence defeats this standardised implementation of attacks; on the other hand, if an attack demonstrates a top score failure rate against a tool defence, the evaluation conclusively indicates that the attack is being capable of defeating a definite implementation of the defence.

3.1.1 Tool structure

The adversarial benchmark tool has a modular architecture oriented software. As it shows in Figure 2 it consists of five primary modules:

1. **Attacks:** The module contains some various attack implementations. One can use these as a template, easily implement other Foolbox adversarial attacks, and extend it attack scenario to the modular tool.
2. **Datasets:** Contains the code for loading different dataset such CelebA.
3. **Models**: Contains the source code for a model. We have a PyTorch model in the directory as an example. Please note that we expect that the model is already trained and we are going to test it against adversarial perturbations.
4. **Use cases:** Contains the logic specific to different use cases such as the Face Re-identification and Face Attribute Alteration tasks in the SAFAIR AI Contest.
5. **Wrapper:** Contains the model converter. This would take a model, for example, implemented in PyTorch and convert it into other open-source tools, such as Foolbox.

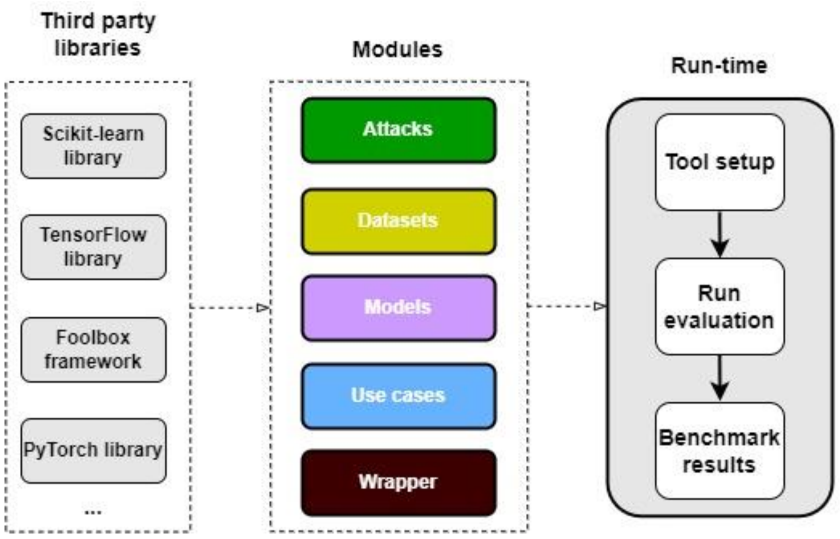


Figure 2: Architecture and main components of Adversarial Benchmark Tool

It can run the following steps: (i) it uses the train_test_split to randomly split the data; (ii) for each training piece, learn the related classifiers; (iii) starting to attack each learned model; and (iv) present the results. Any investigation is run according to the parameters determined in the setup files (like config file), such as the type and parameters of the selected attacks, the chosen datasets, or the use cases. It is possible to execute various investigations via a proper definition of various setup files.

We would like to reiterate in a way that the model is already trained. Hence, one should make sure have saved the model weights. The model conversion process would first load the model weights and then convert the model to Foolbox.

3.1.1.1 Attacks

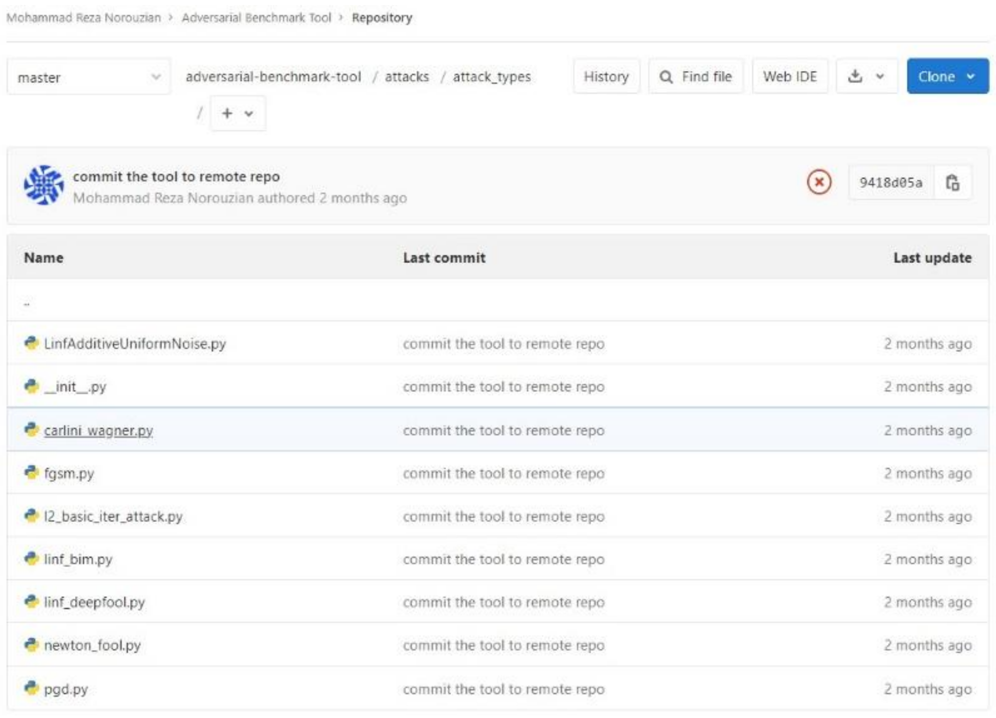
This module implements interfaces to various popular ML adversarial attacks. Each attack takes a model and apply the chosen attacks on the classifiers. The default measure is misclassification. In Figure 3 you can see the attacks that has been implemented in the benchmark tool. Here are two mian examples of the attack_base interface and the list of attacks (attack_lists) which can be use and exten by reseachers and developers.

attack_base

```
import os


import torch


from foolbox.attacks import LinfPGD
import eagerpy as ep


from environment_setup import PROJECT_ROOT_DIR



class Attack:
    def __init__(self):
        pass

    def instantiate_attack(self):
        """
        Attack the model
        :return: NotImplementedError
        """
        raise NotImplementedError

    def attack_description(self):
        """
        String description for the attack
        :return: NotImplementedError
        """
        raise NotImplementedError

    def get_use_case_loss_fn(self, model, labels):
        """
        Selected between reid/attr tasks for proper loss computation. We can switch between cross entropy and bce
        depending upon the task (reid/attr respectively). One can extend this class to accommodate more loss
        functions.
        :param model: Foolbox model :param labels: labels for the inputs
        :return: cross_entropy/bce_with_logits loss
        """
    # can be overridden by users
    def loss_fn(inputs):
        logits = model(inputs)
        if self.task_type == 'reid':
            return ep.crossentropy(logits, labels).sum()
        else:
            # binary cross entropy case in here
            return ep.astensor(
                torch.nn.functional.binary_cross_entropy_with_logits(logits.raw, labels.raw.to(torch.float),
                                                                     reduction="sum"))

    return loss_fn
```

attacks_lists

```
"""
Helper function for loading all the attacks defined in `attacks/attack_types folder`
"""
import importlib
import os
import pkgutil
import sys

from attacks.base import Attack
from environment_setup import PROJECT_ROOT_DIR
import config

def load_all_modules_from_dir(dirname):
    """
    Loads all the attack modules in the current run
    :param dirname: base directory to search from
    :return: None
    """
    for root_dirname, module_name, ispkg in pkgutil.iter_modules([dirname]):
        relative_module_name = f'attacks.attack_types.{module_name}'
        importlib.import_module(relative_module_name, PROJECT_ROOT_DIR)



def get_attacks(task_type):
    """
    Create a list of all the attack instances
    :param task_type: reid/attr attacks
    :return: list of all attack instances
    """
    # First load all modules in the
    attack_base_dir = os.path.join(PROJECT_ROOT_DIR, 'attacks', 'attack_types')
    load_all_modules_from_dir(dirname=attack_base_dir)

    attack_subclasses = Attack.__subclasses__()
    attack_list = []
    for subclass_name in attack_subclasses:
        subclass = subclass_name(task_type=task_type)
        if subclass.attack_description() not in config.skip_list:
            attack_list.append(subclass)
    return attack_list



if __name__ == '__main__':
    print(get_attacks(task_type='attr'))
```


Mohammad Reza Norouzian › Adversarial Benchmark Tool › Repository
master
adversarial-benchmark-tool / attacks / attack_types / +
History
Find file
Web IDE
Clone

commit the tool to remote repo
Mohammad Reza Norouzian authored 2 months ago
9418d05a

Name	Last commit	Last update
..		
LinfAdditiveUniformNoise.py	commit the tool to remote repo	2 months ago
__init__.py	commit the tool to remote repo	2 months ago
carlini_wagner.py	commit the tool to remote repo	2 months ago
fgsm.py	commit the tool to remote repo	2 months ago
l2_basic_iter_attack.py	commit the tool to remote repo	2 months ago
linf_bim.py	commit the tool to remote repo	2 months ago
linf_deepfool.py	commit the tool to remote repo	2 months ago
newton_fool.py	commit the tool to remote repo	2 months ago
pgd.py	commit the tool to remote repo	2 months ago

Figure 3: List of attacks available at Git repository

3.1.2 How to use the tool

Following software required to use the package:

- It has been tested on GNU/Linux, and macOS systems running Python 3.6, and 3.7 installed.
- Other utility packages required can be obtained by taking a look at the requirements.txt file.

3.1.2.1 Installation procedure

The requirements are placed in the requirements.txt file.

```
pip install -r requirements.txt
```

To set up the dependencies. We also suggest making virtual environments by using conda or python virtual environment using:

```
python3 -m venv /path/to/new/virtual/environment
```

3.1.2.2 Execution Steps

To start with execute:

```
python main.py -h
```

To get a list of options and instructions to execute the program. For starting a simple training loop, simply use:

```
python main.py --task_type attr --checkpoint_dir saved_models --model_number 0
```

In our examples, we support two use cases currently (targeted face re-identification and attribute alterations). Everyone can easily develop its own use case to use the tool. This can be selected by using task_type argument. For instance:

```
python main.py --checkpoint_dir saved_models --task_type reid --model_number 0
```

Here we specify the `checkpoint_dir` and `model_number` (specific model to load). We expect that model weights are present in the `saved_models` folder before starting the conversion process. The results shall be computed on the CelebA dataset.

3.1.2.3 Creating a new Attack

To create a new Attack, one has to create a new python file in the `attacks/attack_types` package. Please make sure all attacks extend the `attacks.base.Attack` class. The class has three methods that are used:

- instantiate_attack(). Which is used in order to create an instance of `attack` type defined in Foolbox framework.
- attack_description(). A string representation of the attack.

Once this is done, the tool would automatically recognize the new Attack and compute the model performance against the new attack along with the previous ones (basically all the attacks that are present in the `attacks.attack_types` package).

3.1.2.4 Configuration

There may be scenarios in which you want to skip certain attack types (for instance Carlini and Wagner). This can be done by editing the `config.py` file. The `skip_list` can be used to skip attacks. The tool performs string matching based on the string representation of each attack class. For instance, if you want to skip C&W attack, just use:

```
skip_list = ['carlini_wagner']
```

The string representation of Carlini and Wagner attack is `carlini_wagner` and the same name needs to be used in the configuration list. If the `skip_list` is empty, we run the model against all the test types.

3.1.2.5 Targeted Attacks

One can perform targeted attacks as well. For this simply pass --target label flag with an appropriate target label. For instance:

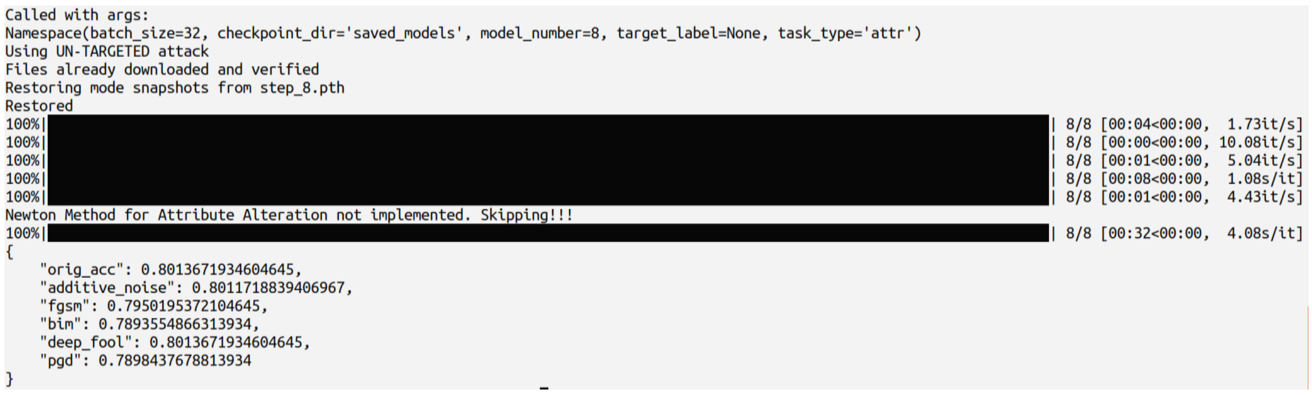
```
python main.py --checkpoint_dir saved_models --task_type reid --model_number 3 --target_label 17
```

The targeted attacks are not defined for attribute alteration tasks and the tool would throw an error before starting. There are certain attacks such as `FGSM` for which targeted attacks are not defined by Foolbox. They would be skipped and the tool would compute results for the remaining attack types.

3.1.2.6 Sample Output

Here are some sample output obtained from the execution of the tool. For example, When executed with the command:

```
python main.py --checkpoint_dir saved_models --model_number 8
```

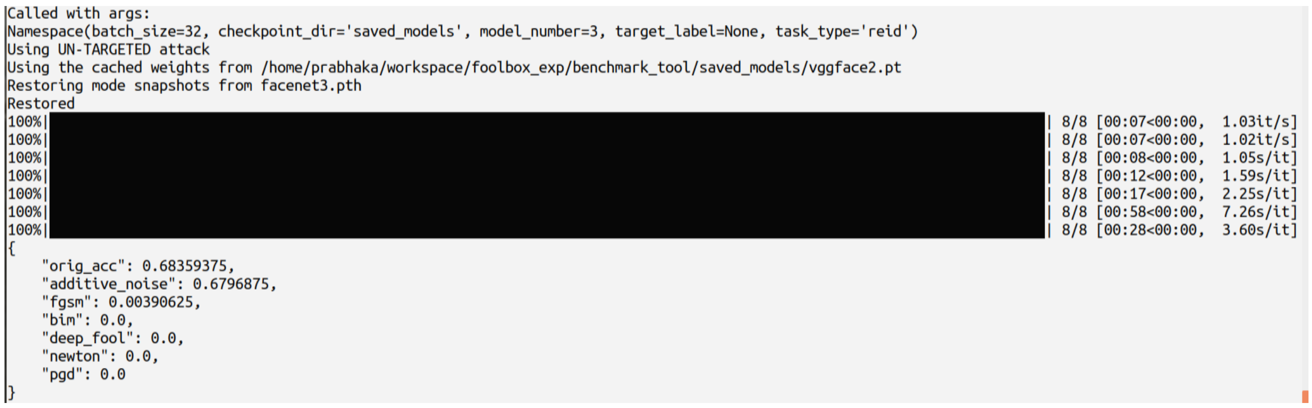


```
Called with args:
Namespace(batch_size=32, checkpoint_dir='saved_models', model_number=8, target_label=None, task_type='attr')
Using UN-TARGETED attack
Files already downloaded and verified
Restoring mode snapshots from step_8.pth
Restored
100%|████████████████████████████████████████| 8/8 [00:04<00:00,  1.73it/s]
100%|████████████████████████████████████████| 8/8 [00:00<00:00, 10.08it/s]
100%|████████████████████████████████████████| 8/8 [00:01<00:00,  5.04it/s]
100%|████████████████████████████████████████| 8/8 [00:08<00:00,  1.08s/it]
100%|████████████████████████████████████████| 8/8 [00:01<00:00,  4.43it/s]
Newton Method for Attribute Alteration not implemented. Skipping!!!
100%|████████████████████████████████████████| 8/8 [00:32<00:00,  4.08s/it]
{
    "orig_acc": 0.8013671934604645,
    "additive_noise": 0.8011718839406967,
    "fgsm": 0.7950195372104645,
    "bim": 0.7893554866313934,
    "deep_fool": 0.8013671934604645,
    "pgd": 0.7898437678813934
}
```

Figure 4: Example output of benchmark tool for attribute alteration task

Similarly, when we execute the tool for performing “Un-targeted Re-Identification” task.

```
python main.py --checkpoint_dir saved_models --task_type reid --model_number 3
```

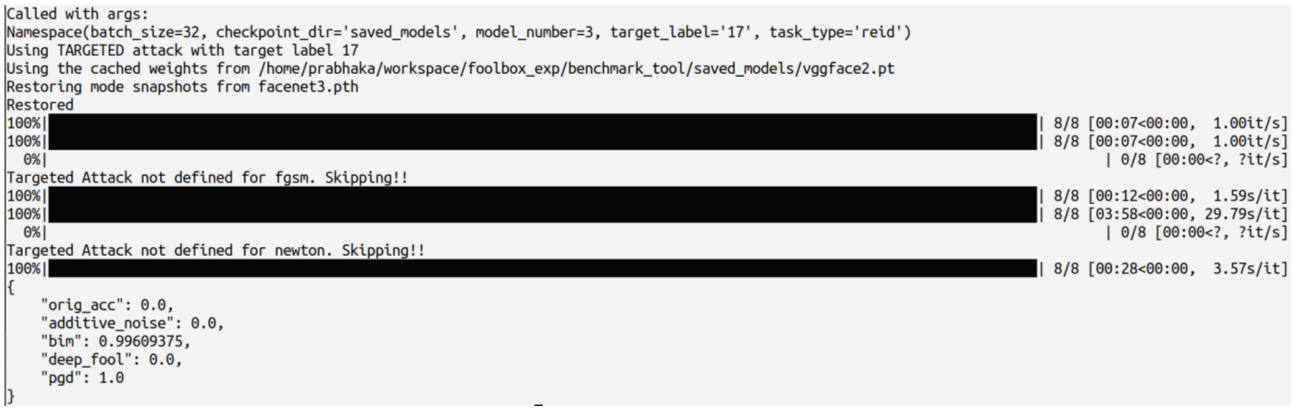


```
Called with args:
Namespace(batch_size=32, checkpoint_dir='saved_models', model_number=3, target_label=None, task_type='reid')
Using UN-TARGETED attack
Using the cached weights from /home/prabhaka/workspace/foolbox_exp/benchmark_tool/saved_models/vggface2.pt
Restoring mode snapshots from facenet3.pth
Restored
100%|████████████████████████████████████████| 8/8 [00:07<00:00,  1.03it/s]
100%|████████████████████████████████████████| 8/8 [00:07<00:00,  1.02it/s]
100%|████████████████████████████████████████| 8/8 [00:08<00:00,  1.05s/it]
100%|████████████████████████████████████████| 8/8 [00:12<00:00,  1.59s/it]
100%|████████████████████████████████████████| 8/8 [00:17<00:00,  2.25s/it]
100%|████████████████████████████████████████| 8/8 [00:58<00:00,  7.26s/it]
100%|████████████████████████████████████████| 8/8 [00:28<00:00,  3.60s/it]
{
    "orig_acc": 0.68359375,
    "additive_noise": 0.6796875,
    "fgsm": 0.00390625,
    "bim": 0.0,
    "deep_fool": 0.0,
    "newton": 0.0,
    "pgd": 0.0
}
```

Figure 5: Example output of benchmark tool for untargeted re-identification task

Finally, when we execute the tool for targeted ReIdentification task, we would get:

```
python main.py --checkpoint_dir saved_models --task_type reid --model_number 3 --target_label 17
```



```
Called with args:
Namespace(batch_size=32, checkpoint_dir='saved_models', model_number=3, target_label='17', task_type='reid')
Using TARGETED attack with target label 17
Using the cached weights from /home/prabhaka/workspace/foolbox_exp/benchmark_tool/saved_models/vggface2.pt
Restoring mode snapshots from facenet3.pth
Restored
100%|████████████████████████████████████████| 8/8 [00:07<00:00,  1.00it/s]
100%|████████████████████████████████████████| 8/8 [00:07<00:00,  1.00it/s]
  0%|                                                  | 0/8 [00:00<?, ?it/s]
Targeted Attack not defined for fgsm. Skipping!!
100%|████████████████████████████████████████| 8/8 [00:12<00:00,  1.59s/it]
100%|████████████████████████████████████████| 8/8 [03:58<00:00, 29.79s/it]
  0%|                                                  | 0/8 [00:00<?, ?it/s]
Targeted Attack not defined for newton. Skipping!!
100%|████████████████████████████████████████| 8/8 [00:28<00:00,  3.57s/it]
{
    "orig_acc": 0.0,
    "additive_noise": 0.0,
    "bim": 0.99609375,
    "deep_fool": 0.0,
    "pgd": 1.0
}
```

Figure 6: Example output of benchmark tool for targeted re-identification task

3.1.2.7 Discussion

The benchmark tool also supports Tensorflow or Keras models too. The respective model definitions using the desired library should be included. Once the class members are defined, one can import the model class in the `main.py` file and then convert it using `tensorflow_to_foolbox` module. If your method uses multiple models, it can be handled easily by properly organizing it under parent model and then call it accordingly. Necessary data processing steps should be included while passing the batches of data from the benchmark tool.

Also, the Foolbox framework does not support Multi-label classification out of the box. Hence, it does not support Attribute Alteration Task of contest directly. We have modified the tool by adding a wrapper which can manage all instances of `FixedEpsilonAttack`.

However, for `AttackWithDistance` subclasses (for instance `carlini_wagner`) it is not possible to perform monkey patching and manage the code. The only possible way is to override the `run` method and basically copy everything with changes to the code as per need. For the sake of keeping the framework simpler and easily extensible, we decided to skip the operation for such cases. Developers are encouraged to update the code themselves and reach out to us for any help.

3.1.3 Reporting benchmark results

This section reports an example of the benchmark tool that Vicomtech has used to test and evaluate their submitted defence methods in D7.2. Any researcher or developer can extend this way of reporting when using the tool to evaluate their solutions.

3.1.3.1 Vicomtech use case evaluation results

All defences were generated for avoiding adversarial attacks in an originally trained model formed by a vgg16 pre-trained layer with a dense layer after it. This model was trained with a breast cancer image dataset for classification task. Due to its vulnerability to adversarial attack, multiple defences were implemented and are detailed next:

- **Adversarial training:** the model is retrained with a dataset containing adversarial examples to learn to classify them correctly [22].
- **Dimensionality reduction** (top and middle version): an autoencoder or encoder layer is added to the initial model (before de input data (top version) or between vgg16 and dense layer (middle version)) with the main idea that this dimensionality reduction happening in this new layer helps reducing the noise added to the original image to convert them into adversarials [22].
- **Prediction similarity:** an external layer is added to save the history of prediction. With the use of this layer's data, the idea is to detect that an adversarial example is being generated.
- **Activations' detector:** studying the behaviour of the activation in the original model for non-adversarial and adversarial images, a detector is trained in order to detect these different behaviours.

Prediction similarity and activations' detector has been modified in order to be able to calculate accuracies of this defences. As the dataset only contains two classes, in the case the external layer detects that an adversarial is being generated or an adversarial was introduced as input images, the prediction is changed to the contrary class as expected from the original model.

All the defences detailed above were validated with all the attacks provided in the benchmark tool. The results of their accuracy are summarised in the Table 2.

Table 2: Benchmark tool results against Vicomtech defence methods

Defence / Accuracy	Adversarial training	Dimensionality reduction top	Dimensionality reduction middle	Prediction Similarity	Activations' detector
Original	0.844	0.726	0.817	0.844	0.708

Defence / Accuracy	Adversarial training	Dimensionality reduction top	Dimensionality reduction middle	Prediction Similarity	Activations' detector
FGSM	0.837	0.716	0.812	0.837	0.708
L2 Basic Iter	0.844	0.725	0.817	0.842	0.708
BIM	0.837	0.716	0.812	0.837	0.708
Deep Fool	0.003	0	0.001	0.015	0.708
Additive noise	0.844	0.725	0.817	0.844	0.708
Newton Fool	0.464	0	0	0.46	0.708
PGD	0.837	0.717	0.813	0.837	0.708

The widely known adversarial training defence got a good result in several attacks. However, this countermeasure manages to avoid the adversarial examples that it already knows. In other words, this method defends the model from the corrupted examples, which are used to retrain the model. That is why the adversarial example would be the ideal defence in case of all these possible attacks are known, and different researches show the impossibility of that. In this case, the model was retrained by adversarials obtained with FGSM, PGD, and BIM algorithms. Therefore, it was expected that this defence was more accurate in those cases. Due to their resemblance in the attacks, this defence also works with Additive noise and L2 Basic Iter attack.

Concerning the dimensionality reduction defence, it was implemented in two ways. The first implementation was the injection of the autoencoder at the beginning of the model to be defended. That method reduces the initial accuracy significantly, and hence it is the defence with the worst results. The second implementation was the addition of the autoencoder in the middle of the original model. In that case, the accuracy was not lowered considerably, improving the previous implementation only changing the position of the developed autoencoder in the defended model. Thus, the second dimensionality reduction shows improved results, even with adversarials that it did not see previously.

The Table 2 shows the dominance of the prediction similarity defence. That proves its potential to avoid the majority of the presented attacks. Concretely, this countermeasure is focused on a history of injected images and the difference between those. Therefore, this method detects the tested algorithms which are constructing the adversarial example by using the gradient.

Finally, the Activations' detector method reduces the accuracy in clean images considerably. However, it is the unique defence that shows robustness against the Deep Fool and Newton Fool attacks. That makes this countermeasure interesting, even though it lowers the precision of the original data significantly.

To summarize, according to the accuracy in the original data, adversarial training and prediction similarity maintain the initial accuracy obtained by the model without any defence. Comparing both versions of dimensionality reduction defences, the middle one generates a more robust model with minimal loss in original accuracy. Finally, although the initial accuracy of the model is reduced drastically, the activations' detector defence is the only one that can be effective against Deep Fool and Newton Fool attacks.

3.1.4 Conclusion

A critical barrier delaying the search for robust ML models is the shortage of reliable evaluation tools. In our explorations and experiments during the SAFAIR program, we believe our tool can help the community evaluate their solutions in an easier and standardised way.

As the adversarial benchmark tool is modular and straightforward to extend, state-of-the-arts and more complicated attacks can be easily implemented, as well as countermeasures and robust models, hopefully, by helping the researchers and developers within the community.

Chapter 4 AI threat model testing and evaluation

The present chapter describes the evaluation of the SAFAIR AI Threat Model and Knowledge Base carried out as part of the task T7.5 and according to the plan detailed in D7.5. After the introductory section, where the aim and context of the evaluation is presented, the following sections summarise the evaluation objectives, evaluation methodology dimensions, evaluation means (online questionnaire), evaluation team, the process followed, and the results obtained. The Appendix A at the end of the report transcripts the questionnaire used in the process and the Appendix B shows the presentation made during the training workshop with the evaluators.

4.1 Introduction

One of the interests of SAFAIR researchers was to learn whether the knowledge gained in the Work Package 7 would be of interest and use of AI system developers. Therefore, as part of the validation of SAFAIR results in T7.5, the SAFAIR AI Threat model and accompanying Knowledge Base were evaluated.

The following sections detail the process and results of the evaluation of the final version of the SAFAIR AI Threat model and Knowledge Base.

The final version of the SAFAIR AI Threat KB evaluated was the version explained in D7.5, updated from that initial version of deliverable D7.1, and which includes the latest advances in trustworthy AI works, such as the “AI Cybersecurity Challenges” by ENISA (ENISA, 2020)[15], which beyond the challenges, offers taxonomies of AI assets and AI threats, as well as the Mitre's ATLAS - Adversarial Threat Landscape for Artificial-Intelligence Systems (MITRE ATLAS, 2021)[16], and the ETSI-SAI’s “Mitigation Strategy Report” (ETSI SAI Mitigation, 2021)[17] dedicated to countermeasures. Furthermore, the updated version of the SAFAIR AI Threat Knowledge Base enlarged the knowledge corpus with countermeasures, explainability and fairness solutions resulting from SAFAIR work as described in D7.5, as well as other state-of-the-art works on Adversarial Machine Learning (AML) attacks and safeguards published and analysed after D7.1 was issued.

4.2 SAFAIR AI Threat Knowledge Base evaluation

The evaluation of the SAFAIR AI Threat Knowledge Base tool (a.k.a. SAFAIR AI Threat KB) was the mean to evaluate the SAFAIR AI Threat Model and the analysed corpus knowledge structured following the Model. The initial version of the tool was part of D7.1 and was updated into the final version presented in D7.5 which was the subject of the evaluation.

The evaluation process followed the methodology and the work plan outlined in D7.5 and it is explained in more detail in the following sections. The last section summarizes the feedback and the evaluation results obtained.

4.2.1 Evaluation Objectives

The main objective of the SAFAIR AI Threat KB evaluation is to assess if its knowledge content and structure is correct and sufficient for the users when interacting with it. Therefore, the focus of the evaluation was the quality and completeness of the KB content.

[15] ENISA, AI Cybersecurity Challenges - Chapter 1. Threat Landscape for Artificial In-telligence. December 2020. Available at: https://www.enisa.europa.eu/publications/artificial-intelligence-cybersecurity-challenges/at_download/fullReport
[16] MITRE ATLAS, Adversarial Threat Landscape for Artificial-Intelligence Systems. Available at: https://atlas.mitre.org/
[17] ETSI GR SAI 005 V1.1.1 (2021-03), Securing Artificial Intelligence (SAI); Mitigation Strategy Report

4.2.2 Evaluation Dimensions

We considered five dimensions in the evaluation process, being the following, in order of importance:

- **Quality** – to check if the information provided by the Knowledge Base is good, correct, sufficient and useful.
- **Correctness** – to check if the description of techniques and countermeasures is appropriate (i.e. it reflects well the source) and if it is well understood.
- **Completeness** – to check if all the content that should be in the Knowledge Base has been included.
- **Usability** – to check whether the content is useful for the users in the AI use case under study, and whether there are enough instances of attack techniques and countermeasures that have been useful or previously unknown to them. Please, note that in this dimension, we are not evaluating the user experience of the tool because the tool has no GUI.
- **Re-usability** – to check whether the information provided by the Knowledge Base could be used in the future for models similar to the model under study.

4.2.3 Evaluation Questionnaire

A **questionnaire** with dedicated questions to assess the different evaluation dimensions above was designed to support the evaluation process (see Appendix A). The questionnaire was used to get the evaluators' feedback about different aspects of the tool after reading the documentation of the tool and after using the Knowledge Base.

In order to facilitate the feedback gathering and statistics, the questionnaire was an online questionnaire shared with the evaluators. Both the questionnaire editing and the questionnaire processing procedures followed GDPR principles with regards to protection of personal information of the responding data subjects, i.e. securely storing and processing this information; letting the responders know the purpose of the processing; and allowing them opting out at any time.

The questions included general questions about the evaluator profile and AI development practices and AML knowledge, as well as technical questions related to the quality of the content of the KB. The questions were grouped in seven sections:

1. **Profile.** In this section, we collect information about the evaluators' experience in AI development.
2. **Current Practice.** This section collects information about the current evaluators' AI trustworthiness practices when developing AI systems.
3. **Correctness.** This section asks the evaluator to assess the overall quality of the threat and countermeasure information provided by the Knowledge Base in terms of correctness.
4. **Completeness.** In this section, we ask the evaluator to assess the overall quality of the threat and countermeasure information provided by the Knowledge Base in terms of completeness.
5. **Usefulness.** In this section, the evaluator is required to assess the usefulness of the attack and defence information provided, i.e. whether s/he thinks the information is useful for improving the AI system design with respect to trustworthiness.
6. **Re-usability.** This section includes questions to the evaluators on the re-usability of the attack and defence information provided in the Threat KB, i.e. whether they believe the information can be used for improving the trustworthiness of AI systems in other contexts beyond the ones evaluated.

7. **Background knowledge and suggestions**. This section inquires the evaluators about their background on similar tools and initiatives around the “Trustworthy AI” principles. This information is relevant in order to learn on the knowledge and interest of the evaluators about the AI trustworthiness support subject.

4.2.4 Evaluators

The selected evaluation team was composed of **eleven** actual AI designers and developers in Tecnalia that are members of different Tecnalia divisions and are working in AI system development projects in different industry sectors and application domains (health, energy, cybersecurity, etc.).

From them, 9 out of 11 are experts coming from the field of AI with limited knowledge on cybersecurity and two of them are cybersecurity experts with less experience in AI development.

All of them took the role of AI designer or AI developer responsible for developing secure AI systems and were asked to learn the concept and use the SAFAIR AI Threat KB to get information and gain insights into threats against AI systems and possible countermeasures.

All the evaluators participated in the evaluation in a voluntary basis.

4.2.5 Evaluation process

Figure 7 depicts the overall evaluation process followed. As shown in the figure, first, a Training workshop was held organised by Tecnalia participants in SAFAIR with the evaluators of the SAFAIR AI Threat KB as trainees. During the session, the evaluators were enlightened on SPARTA project objectives and main research activities and outcomes, as well as on WP7 SAFAIR objectives, tasks and results among which the AI Threat model and KB were carefully explained and thoroughly detailed (see Appendix B). In the Training workshop, the evaluators were also instructed on the evaluation goals and the process to follow, including a detailed description of the questionnaire they had to fill.

In order to illustrate what they need to do to get information from the KB, two supporting videos were showcased, showing two use cases of the potential use of the KB for improving the trustworthiness of the ML models.

After that, the evaluators were aided in browsing the contents of the SAFAIR AI Threat KB so they could learn the threats and countermeasures that had more interest to them according to the type of algorithms they research and develop.

The evaluation concluded with all the evaluators voluntarily filling out the online questionnaire to provide their feedback, which is reported in the next section.

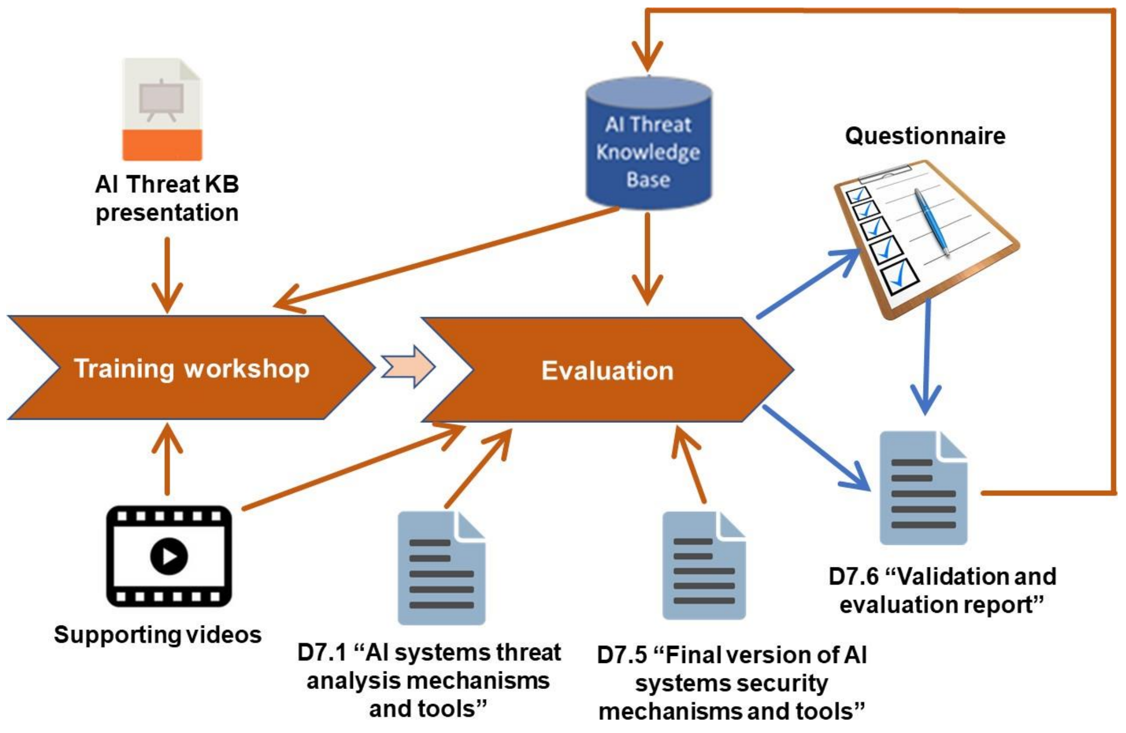

AI Threat KB presentation
AI Threat Knowledge Base
Questionnaire
Training workshop
Evaluation
Supporting videos
D7.1 "AI systems threat analysis mechanisms and tools"
D7.5 "Final version of AI systems security mechanisms and tools"
D7.6 "Validation and evaluation report"

Figure 7: SAFAIR AI Threat Knowledge Base Evaluation process

The supporting materials offered to the evaluators included the following:

- D7.1 "AI systems threat analysis mechanisms and tools" document, which describes the AI Threat model together with the KB structure and summarizes the literature review performed in SAFAIR.
- D7.5 "Final version of AI systems security mechanisms and tools" report, which describes the updates performed on the AI Threat model and KB, as well as details the mechanisms for defence, explainability and fairness developed in SAFAIR (which are also part of the final version of the KB).
- Supporting videos illustrating two example use cases of the SAFAIR AI Threat Knowledge Base:
  - UC1: Poisoning Threat Analysis on AI-based Healthcare system for disease detection.
  - UC2: Threat Analysis on AI-based network traffic classification system realized by Support Vector Machines (SVM).
- The final version of the SAFAIR AI Threat Knowledge Base tool.
- Presentation for the Training workshop (see Appendix B).
- Recording of the Training workshop.

The evaluators were asked to perform a **threat analysis** of the AI system of their interest using the SAFAIR AI Threat Knowledge Base. The information about the threats and associated potential countermeasures were consulted following the process explained to them:

- First: Identify the ML algorithm and family of your interest (i.e. the AI asset).
- Second: Identify the relevant potential threats and attacks, including all the attributes, e.g. attack tactic group, tactic, technique, threat agent knowledge, etc.
- Third: Identify the countermeasures to adopt for protecting your ML algorithm.

4.2.6 *Evaluation results*

In this section, we provide the statistics of questionnaire responses gathered and the analysis of the results. The results have provided relevant feedback to improve the contents of the SAFAIR AI Threat KB and the design of the questionnaire so that it can be extended to an external audience to further improve the KB.

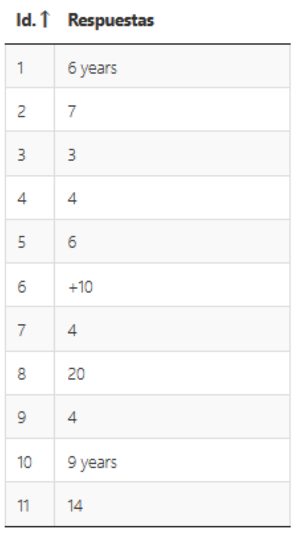
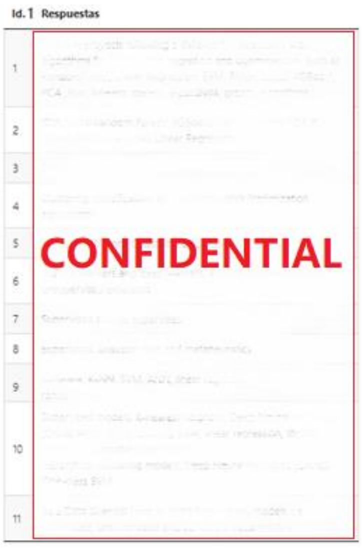
4.2.6.1 Section 1: Profile

As shown by the responses of the evaluators, all of them have more than three years of experience in ML system development, and 36% of them have more than seven years, which reflects the deep knowledge of the evaluators on the types of systems to be protected.

Furthermore, the types of ML algorithms most used by the evaluators, that is, those used at least by the 60% of the evaluators, were: K-means clustering, PCA, Linear Regression, Logistic Regression, SVM, Genetic Algorithm and Neural Networks.

The target domains of the ML models developed were multiple, including health applications, energy efficiency, cybersecurity, etc.

While only 27% of the responders have a background in cybersecurity, 45% of them did already have experience in AML practices, which makes their answer even more valuable for the purpose.

1. Please indicate how many years of experience you have in AI development	**2.** Please specify the types of ML algorithms or AI models you develop
	

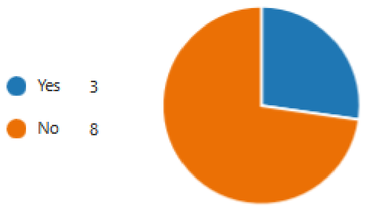
Id. ↑	Respuestas
1	6 years
2	7
3	3
4	4
5	6
6	+10
7	4
8	20
9	4
10	9 years
11	14

Id. ↑	Respuestas
1	CONFIDENTIAL
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	

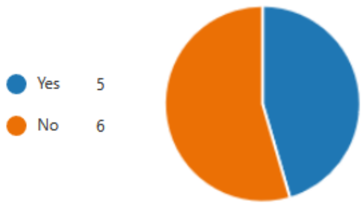
3. Please specify the types of AI applications you develop (i.e. the mission and domain of application).

Id. ↑	Respuestas
1	[redacted]
2	[redacted]
3	[redacted]
4	[redacted]
5	[redacted]
6	CONFIDENTIAL
7	[redacted]
8	[redacted]
9	[redacted]
10	[redacted]
11	[redacted]

4. Do you have any experience in cyber security in general?



5. Do you have any experience in Adversarial Machine Learning?

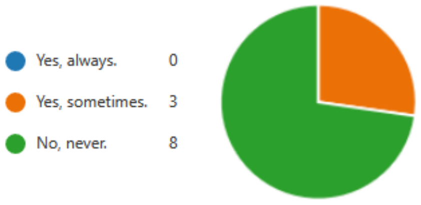


4.2.6.2 Section 2: Current Practice

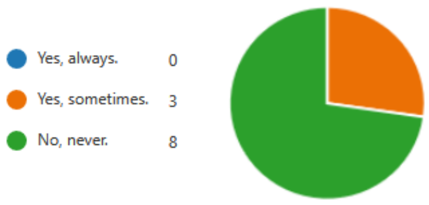
As reported by the evaluators, the current practice on threat analysis and protection of AI systems is limited. Only 27% of the evaluators consider the exposure of the AI system to cyber threats and the possible attacks on them. Even more, only 18% try to adopt measures to counter such risks. However, 18% also test their AI systems for integrity, robustness and security.

When asked about the potential attacks of their interest, all the main four families (poisoning, evasion, oracle and data access) were mentioned by the evaluators, with a greater proportion of poisoning and evasion threats.

6. When developing an AI system, do you think on how exposed the AI system is to cyber-attacks?



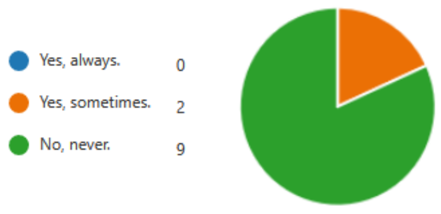
7. Do you assess potential forms of attacks against your AI systems?



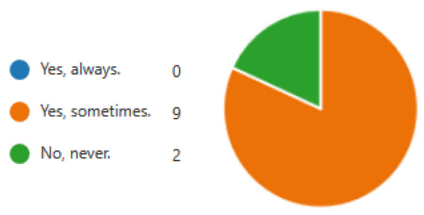
8. If yes, what types of potential attacks do you consider?

Id. ↑	Respuestas
1	Unintended (data pollution, obsolete..) or intended (maliciuos feeding, etc..)
2	Depending on the mathematics behind each model, their vulnerabilities differ. Some are sensible to data normalization or scalation. Others are sensible to data domain, balance among classes, etc.
3	I have recently read about how to extract training data from trained networks
4	Evasion attacks

9. Do you put measures in place to counter such potential attacks over the lifecycle of the AI system?



10. Do you test the AI system to check its integrity, robustness and security?



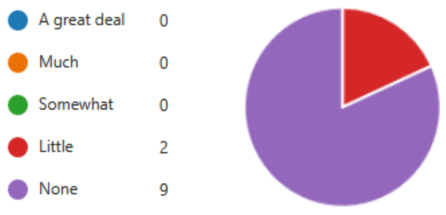
11. What types of potential attacks are more relevant to you?

1	For me, the most relevant is the Evasion Attack.
2	Unreliable training dataset already infected) Unknown attacks->not detected in AI->other non-AI mechanisms to stop or recover.
3	Direct poisoning - Data Manipulation - Input Manipulation
4	NA
5	data poisoning and model substitution
6	Attacks that differentiate between model attacks and system attacks. Most of the attacks are suitable for any model given the right circumnstances and are important for the implementation of the system not so much for the development of the model.
7	Data Access
8	Evasion and Oracle attacks
9	For supervised models the ones which try to infer statistical characteristics of data, because maybe the attack can infer some things about the bussiness.

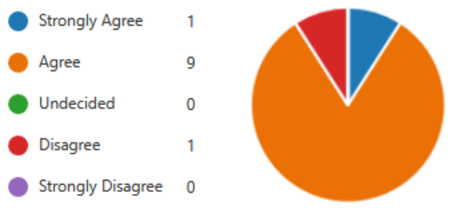
4.2.6.3 Section 3: Correctness

The great majority of the evaluators agree on the quality of the KB with respect to the correctness and easiness of the information contained, while none of them indicated any mistake in the content or source references provided in the threat or countermeasure information.

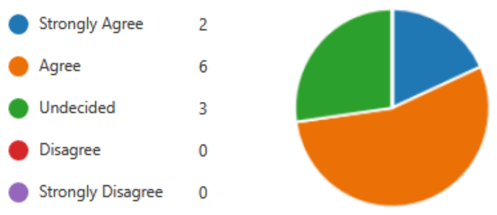
12. Did you find any mistake or erroneous information?



13. Do you think the information is easy to understand?



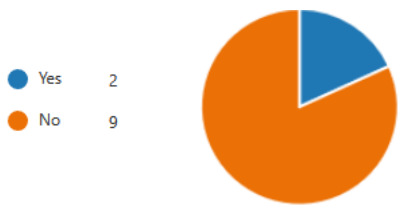
14. Do you think the information reflects or summarises well the corresponding reference source?



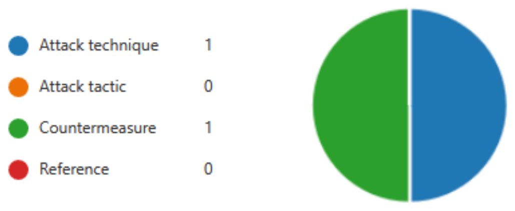
4.2.6.4 Section 4: Completeness

The completeness of the KB was evaluated very positively by 82% of the evaluators, even though one evaluator indicated that some attack technique and countermeasure information could be improved, and 64% of the evaluators would have appreciated having additional complementary information and materials.

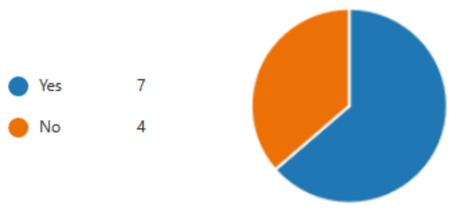
15. Did you miss any information?



16. If yes, please specify which type of information



17. Did you miss any complementary tool, background documentation or other material/resource?



4.2.6.5 Section 5: Usefulness

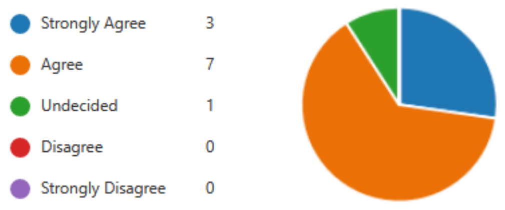
Most of the evaluators, a total of 91% of them, deemed relevant the information about threats and countermeasures offered by the KB. Even more, 82% of them acknowledge that the KB offered new information not previously known.

The utility of the content was clearly evaluated positively, particularly for aiding AI system developers in their work. A total of 82% of the evaluators considered it useful for developers, while 55% agreed that it is also useful for researchers. When asked for clarifications, the disagreeing evaluator explained that there was a misunderstanding on the roles of "AI developer" and "AI researcher". The negative evaluation reflects the opinion of this evaluator about the fact that AML threats are usually an issue of AI system implementation rather than an issue of the mathematical algorithm itself. Therefore, from the perspective of the ML model creator, the information in the KB may not impact his work.

The evaluators replied that the knowledge compiled is clearly most useful for improving security-by-design and privacy-by-design of AI systems, while AI fairness and interpretability are less improved, which is normal considering the current proportion of threats gathered for each of the aspects.

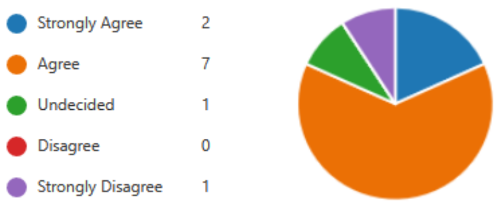
The confidence of the evaluators about being able to use the KB to actually improve the AI systems by applying the countermeasures information provided is in a ratio of 73% of the evaluators who agree they have sufficient skills.

18. Do you think that the information given has been relevant?



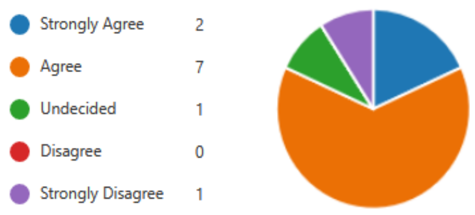
● Strongly Agree	3
● Agree	7
● Undecided	1
● Disagree	0
● Strongly Disagree	0

19. Was the information given new and not previously known?



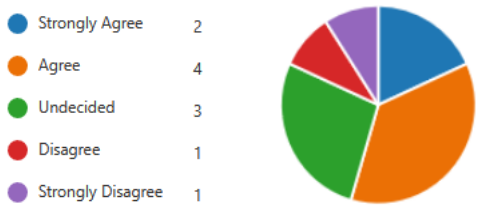
● Strongly Agree	2
● Agree	7
● Undecided	1
● Disagree	0
● Strongly Disagree	1

20. Do you think the information given is useful for a developer of the AI use cases you studied?



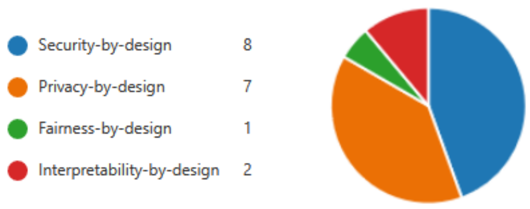
● Strongly Agree	2
● Agree	7
● Undecided	1
● Disagree	0
● Strongly Disagree	1

21. Do you think the information given is useful for a researcher of the AI use cases you studied?



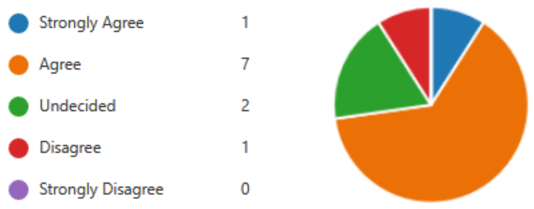
● Strongly Agree	2
● Agree	4
● Undecided	3
● Disagree	1
● Strongly Disagree	1

22. If yes, which aspects of the AI system do you think the information given helps to improve?



● Security-by-design	8
● Privacy-by-design	7
● Fairness-by-design	1
● Interpretability-by-design	2

23. Do you think that you have the skills to implement the recommended countermeasures?

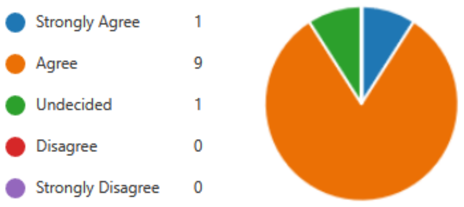


● Strongly Agree	1
● Agree	7
● Undecided	2
● Disagree	1
● Strongly Disagree	0

4.2.6.6 Section 6: Re-usability

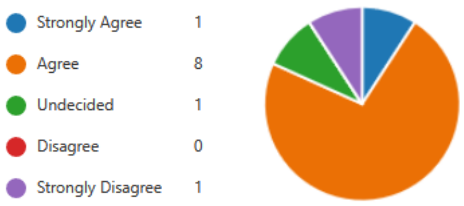
The re-usability quality of the KB has also been evaluated very positively as 91% of the evaluators agree with the statement that the KB could be used in similar use cases beyond the studied ones. Furthermore, 82% agree that the knowledge gained could be useful for improving the trustworthiness of other ML models in the future.

24. Do you think that the information provided could be used in other use cases similar to the ones you studied?



● Strongly Agree	1
● Agree	9
● Undecided	1
● Disagree	0
● Strongly Disagree	0

25. Do you think that the knowledge you gained could be useful in the future for improving the trustworthiness of other models you will develop?



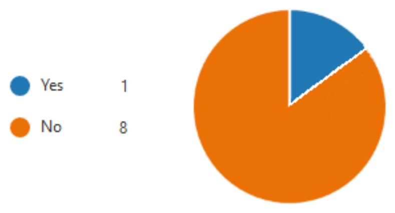
● Strongly Agree	1
● Agree	8
● Undecided	1
● Disagree	0
● Strongly Disagree	1

4.2.6.7 Section 7: Background knowledge and suggestions

Being great experts in AI but not cybersecurity, the majority of the evaluators did not have an extensive background in AML. 82% stated they did not know further frameworks of AML testing or similar KBs, and they did not even know about the current work on AI certification by ENISA.

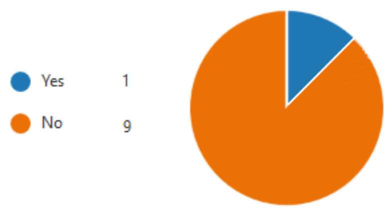
82% also think that it would be interesting to automate the AML testing/assessment techniques of ML/AI models, which gives the idea of similar tools such as the SAFAIR AI Threat KB.

26. Do you know any tool or framework to perform the AML testing? Please provide the reference(s)



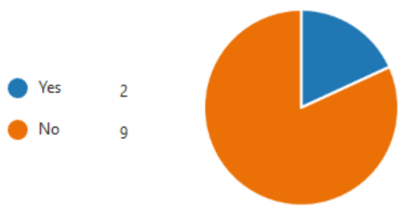
"I know python AML libraries where you can apply well-known AML countermeasures to evaluate the robustness of a model."

27. Do you know any other similar public or private Knowledge Base? Please provide the reference(s).

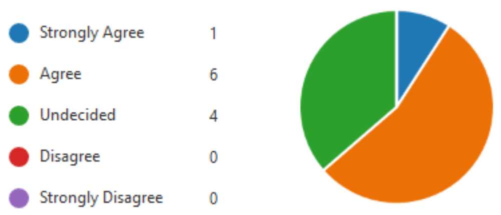


"The MITRE ML Threat Matrix."

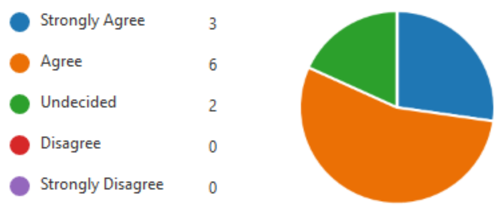
28. Did you already know that ENISA is working on an AI system certification scheme?



29. Do you think it would be interesting to have a European certification of AI system trustworthiness?



30. Do you think it would be interesting to automate the AML testing/assessment techniques of ML/AI models?



4.3 Conclusions

The overall conclusion of the SAFAIR AI Threat KB evaluation results is that the great majority of the evaluators agreed on the high quality, correctness, completeness, usefulness and re-usability of the contents of the Knowledge Base.

According to their feedback, in general, the KB does not present any error or gap of information, providing easy to understand descriptions as well as complete references. Most of the evaluators considered the information about threats and countermeasures therein as relevant and useful for developers to improve the security and privacy of ML models, with limited usefulness to improve AI fairness and explainability, which is aligned with the amount and proportion of the types of mechanisms collected.

Moreover, the evaluators believe that the knowledge gained from the techniques and countermeasures studied could aid in improving multiple types of ML models of diverse use cases.

The evaluation also showed the need of spreading the Trustworthy AI concepts and works among AI developers, including SAFAIR results on AML, since most of the evaluators showed limited awareness of the landscape of works and other initiatives around Trustworthy AI, and only a minority of the evaluators already knew about current efforts in these aspects.

Most of the evaluators also think that it would be interesting to automate the AML testing and assessment, which gives a positive view of the future of the SAFAIR AI Threat KB in support of such automation.

Chapter 5 External validation through peer review

One of the only widely accepted methods for the validation of scientific research is the peer-review process. The procedure has a successful tradition of over 350 years and plays a critical role in the scientific publishing process. Major scientific publishers maintain the peer review process as a way to guarantee the validity and quality of published research pieces.

Some of the work completed in the Sparta SAFAIR program have successfully undergone validation through the peer review process in top-tier journals. The following chapter presents the key concepts and findings of the published works.

5.1 Defending network intrusion detection systems against adversarial evasion attacks

The work completed on the detection of Evasion Attacks on the CICIDS2017 dataset has been described in d7.2. The results of this work have been evaluated and published in a top tier journal - Future Generation Computer System (Impact Factor = 7.187).

- Pawlicki, Marek, Michał Choraś, and Rafał Kozik. "Defending network intrusion detection systems against adversarial evasion attacks." Future Generation Computer Systems 110 (2020): 148-154

The algorithms used for the creation of evasion attacks were:

- Carlini and Wagner attack (CW)
- Fast Gradient Sign Method (FGM)
- Basic Iterative Method (BIM)
- Projected Gradient Descent (PGD)

The diagram in Figure 8 shows the Training/Testing Pipeline of the adversarial detector. In essence, a secondary ML-based model trained on the neuron activation values from the network intrusion detection neural network, which allows spotting odd behaviour of the network, which might indicate the occurrence of an attack.

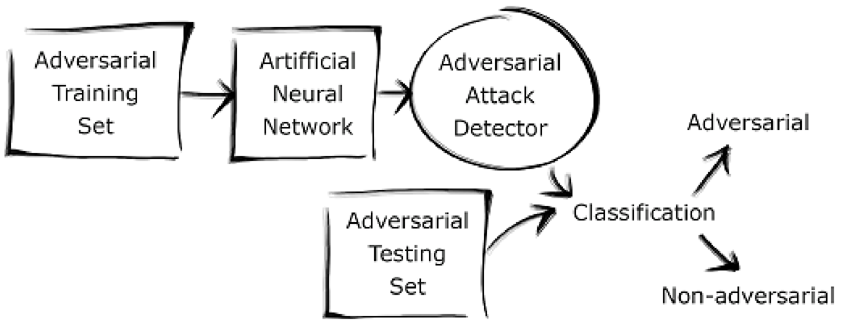
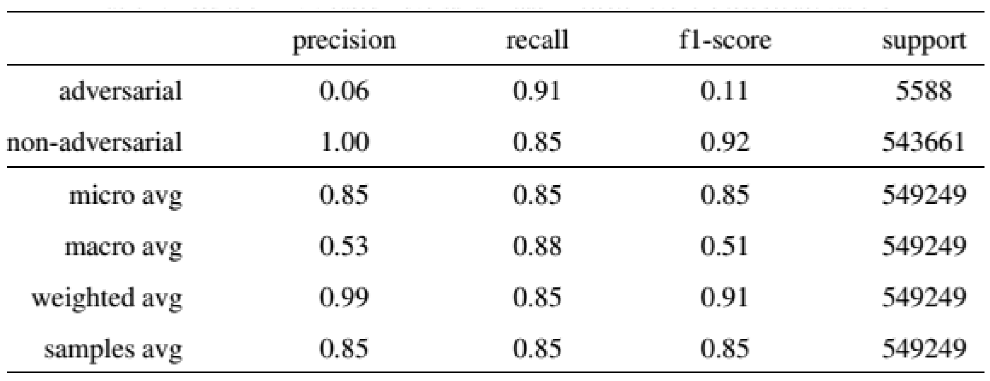


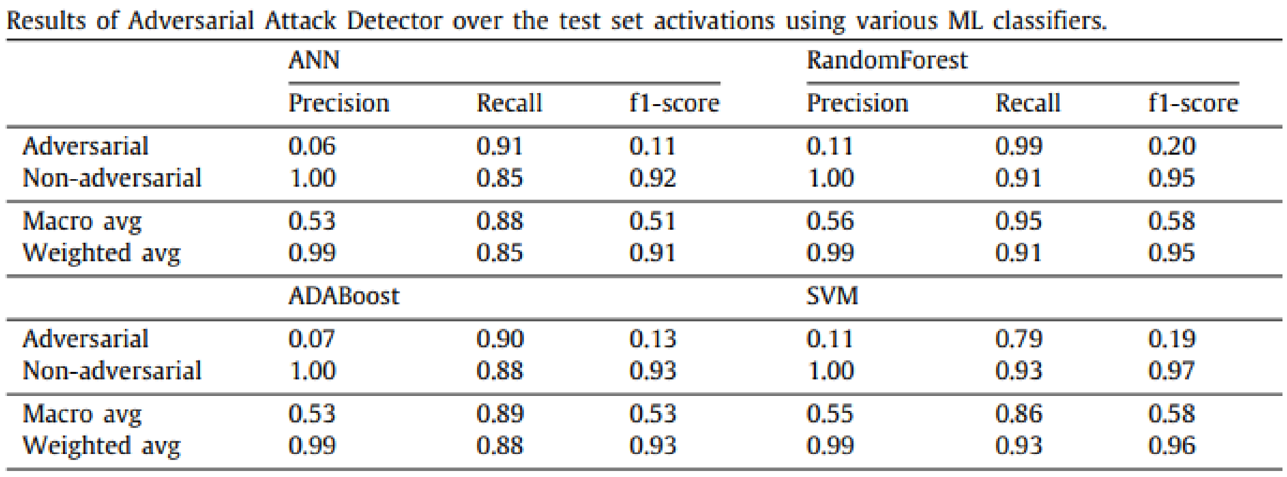
Figure 8: The Adversarial Detector Training/Testing Pipeline

The detector achieved an accuracy of 0.8506 on the testing set. The detailed results containing the precision and recall metrics are assembled in Figure 9 and Figure 10.



	precision	recall	f1-score	support
adversarial	0.06	0.91	0.11	5588
non-adversarial	1.00	0.85	0.92	543661
micro avg	0.85	0.85	0.85	549249
macro avg	0.53	0.88	0.51	549249
weighted avg	0.99	0.85	0.91	549249
samples avg	0.85	0.85	0.85	549249

Figure 9: Results of ANN-based Adversarial Attack Detector over the test set activations



Results of Adversarial Attack Detector over the test set activations using various ML classifiers.

	ANN			RandomForest		
	Precision	Recall	f1-score	Precision	Recall	f1-score
Adversarial	0.06	0.91	0.11	0.11	0.99	0.20
Non-adversarial	1.00	0.85	0.92	1.00	0.91	0.95
Macro avg	0.53	0.88	0.51	0.56	0.95	0.58
Weighted avg	0.99	0.85	0.91	0.99	0.91	0.95
	ADABoost			SVM		
Adversarial	0.07	0.90	0.13	0.11	0.79	0.19
Non-adversarial	1.00	0.88	0.93	1.00	0.93	0.97
Macro avg	0.53	0.89	0.53	0.55	0.86	0.58
Weighted avg	0.99	0.88	0.93	0.99	0.93	0.96

Figure 10: Results of Adversarial Attack Detection with other ML methods

5.2 The application of preprocessing adversarial defences to robustify face reidentification systems

The work on preprocessing pipelines to protect computer vision algorithms in face re-identification tasks against adversarial evasion attacks has been described in d7.5 and also published in the Entropy Journal (Impact Factor = 3.012)

- Pawlicki, Marek, and Ryszard S. Choraś. "Preprocessing Pipelines including Block-Matching Convolutional Neural Network for Image Denoising to Robustify Deep Reidentification against Evasion Attacks." Entropy 23, no. 10 (2021): 1304.

The classifier performance on the test set containing the 14 most populated classes is found in Table 3.

Table 3: Classifier performance on the test set containing the 14 most populated classes.

label	precision	recall	f1-score
1757.0	1.00	1.00	1.00
2114.0	1.00	1.00	1.00
2820.0	0.88	1.00	0.93

label	precision	recall	f1-score
3227.0	1.00	0.86	0.92
3699.0	0.88	1.00	0.93
3745.0	1.00	1.00	1.00
3782.0	1.00	1.00	1.00
4262.0	0.88	1.00	0.93
4740.0	1.00	1.00	1.00
4978.0	1.00	1.00	1.00
6568.0	1.00	1.00	1.00
8968.0	1.00	1.00	1.00
9152.0	1.00	1.00	1.00
9256.0	1.00	0.71	0.83
macro avg	0.97	0.97	0.97
weighted avg	0.97	0.97	0.97
accuracy	0.9693877551020408		
balanced accuracy	0.9693877551020408		

The effects of PGD eps=4 on the performance of the classifier can be seen in Table 4.

Table 4: The effects of PGD eps=4 on the performance of the classifier.

label	precision	recall	f1-score
1757.0	1.00	0.14	0.25
2114.0	0.33	0.14	0.20
2820.0	0.00	0.00	0.00
3227.0	1.00	0.17	0.29
3699.0	0.32	1.00	0.48
3745.0	0.00	0.00	0.00
3782.0	0.00	0.00	0.00
4262.0	0.33	0.71	0.45

label	precision	recall	f1-score
4740.0	0.08	0.14	0.11
4978.0	0.00	0.00	0.00
6568.0	1.00	0.14	0.25
8968.0	0.00	0.00	0.00
9152.0	0.50	0.14	0.22
9256.0	1.00	0.40	0.57
macro avg	0.40	0.21	0.20
weighted avg	0.38	0.21	0.19
accuracy	0.21052631578947367		
balanced accuracy	0.2139455782312925		

The results of the classifier using JPEG compression with quality set to 20 on PGD attacks with epsilon=4 can be found in Table 5.

Table 5: The results of the classifier using JPEG compression with quality set to 20 on PGD attacks with epsilon=4.

label	precision	recall	f1-score
1757.0	1.00	1.00	1.00
2114.0	1.00	1.00	1.00
2820.0	1.00	1.00	1.00
3227.0	1.00	0.83	0.91
3699.0	0.88	1.00	0.93
3745.0	0.86	0.86	0.86
3782.0	0.86	0.86	0.86
4262.0	0.78	1.00	0.88
4740.0	1.00	1.00	1.00
4978.0	0.86	0.86	0.86
6568.0	1.00	1.00	1.00
8968.0	1.00	0.86	0.92

label	precision	recall	f1-score
9152.0	1.00	0.86	0.92
9256.0	0.80	0.80	0.80
macro avg	0.93	0.92	0.92
weighted avg	0.93	0.93	0.93
accuracy	0.9263157894736842		
balanced accuracy	0.9227891156462587		

Table 6: The results of the classifier using BMCNN with sigma set to 20 used on adversarial samples created with PGD using with epsilon set to four.

label	precision	recall	f1-score
1757.0	1.00	1.00	1.00
2114.0	1.00	1.00	1.00
2820.0	1.00	1.00	1.00
3227.0	0.83	0.83	0.83
3699.0	0.70	1.00	0.82
3745.0	1.00	0.71	0.83
3782.0	0.88	1.00	0.93
4262.0	0.78	1.00	0.88
4740.0	1.00	1.00	1.00
4978.0	0.88	1.00	0.93
6568.0	1.00	0.86	0.92
8968.0	1.00	0.86	0.92
9152.0	0.80	0.57	0.67
9256.0	1.00	0.8	0.89
macro avg	0.92	0.90	0.90
weighted avg	0.92	0.91	0.90
accuracy	0.9052631578947369		
balanced accuracy	0.9023809523809525		

Table 7: The results of the classifier using spatial smoothing with JPEG compression, gaussian augmentation, total variance minimisation and BMCNN with sigma set to 20 on PGD images with epsilon set to four.

label	precision	recall	f1-score
1757.0	0.50	0.71	0.59
2114.0	0.50	0.43	0.46
2820.0	0.00	0.00	0.00
3227.0	0.40	0.33	0.36
3699.0	0.37	1.00	0.54
3745.0	0.25	0.14	0.18
3782.0	0.25	0.86	0.39
4262.0	0.25	0.14	0.18
4740.0	0.50	0.57	0.53
4978.0	0.67	0.29	0.40
6568.0	1.00	0.14	0.25
8968.0	0.50	0.14	0.22
9152.0	0.67	0.29	0.40
9256.0	0.00	0.00	0.00
macro avg	0.42	0.36	0.32
weighted avg	0.43	0.37	0.33
accuracy	0.3684210526315789		
balanced accuracy	0.36054421768707484		

Table 8: The results of the classifier using spatial smoothing with JPEG compression, gaussian augmentation and BMCNN with sigma set to 20 on PGD images with epsilon set to four, without total variance minimisation.

label	precision	recall	f1-score
1757.0	1.00	1.00	1.00
2114.0	1.00	1.00	1.00
2820.0	1.00	1.00	1.00
3227.0	0.83	0.83	0.83

label	precision	recall	f1-score
3699.0	0.78	1.00	0.88
3745.0	1.00	0.86	0.92
3782.0	0.75	0.86	0.80
4262.0	0.78	1.00	0.88
4740.0	1.00	1.00	1.00
4978.0	0.86	0.86	0.86
6568.0	1.00	0.86	0.92
8968.0	1.00	0.86	0.92
9152.0	1.00	0.57	0.73
9256.0	0.83	1.00	0.91
macro avg	0.92	0.91	0.90
weighted avg	0.92	0.91	0.90
accuracy	0.9052631578947369		
balanced accuracy	0.9064625850340137		

Table 9: The results of the classifier using spatial smoothing with JPEG compression on PGD images with epsilon set to four.

label	precision	recall	f1-score
1757.0	1.00	1.00	1.00
2114.0	1.00	1.00	1.00
2820.0	1.00	1.00	1.00
3227.0	1.00	0.83	0.91
3699.0	0.78	1.00	0.88
3745.0	0.86	0.86	0.86
3782.0	0.86	0.86	0.86
4262.0	0.78	1.00	0.88
4740.0	1.00	1.00	1.00
4978.0	0.86	0.86	0.86

label	precision	recall	f1-score
6568.0	1.00	1.00	1.00
8968.0	1.00	0.86	0.92
9152.0	1.00	0.71	0.83
9256.0	0.80	0.80	0.80
macro avg	0.92	0.91	0.91
weighted avg	0.93	0.92	0.92
accuracy	0.9157894736842105		
balanced accuracy	0.9125850340136055		

Table 10: The results of the classifier using JPEG compression, gaussian augmentation and BMCNN on PGD images with epsilon set to four.

label	precision	recall	f1-score
1757.0	0.88	1.00	0.93
2114.0	1.00	1.00	1.00
2820.0	1.00	1.00	1.00
3227.0	1.00	0.83	0.91
3699.0	0.78	1.00	0.88
3745.0	0.86	0.86	0.86
3782.0	0.86	0.86	0.86
4262.0	0.88	1.00	0.93
4740.0	1.00	1.00	1.00
4978.0	0.86	0.86	0.86
6568.0	1.00	1.00	1.00
8968.0	1.00	0.86	0.92
9152.0	1.00	0.71	0.83
9256.0	1.00	1.00	1.00
macro avg	0.94	0.93	0.93
weighted avg	0.93	0.93	0.93

label	precision	recall	f1-score
accuracy	0.9263157894736842		
balanced accuracy	0.9268707482993197		

To assess the results of the preprocessing defences, the best performing preprocessing pipeline was tested on a clean, unperturbed set. The results of this experiment can be found in Table 9.

Table 11: Results of classification with preprocessing defences on a clean dataset.

label	precision	recall	f1-score
1757.0	1.00	1.00	1.00
2114.0	1.00	1.00	1.00
2820.0	1.00	1.00	1.00
3227.0	1.00	0.83	0.91
3699.0	0.88	1.00	0.93
3745.0	0.83	0.71	0.77
3782.0	0.75	0.86	0.80
4262.0	0.78	1.00	0.88
4740.0	1.00	1.00	1.00
4978.0	1.00	1.00	1.00
6568.0	1.00	1.00	1.00
8968.0	1.00	1.00	1.00
9152.0	1.00	1.00	1.00
9256.0	1.00	0.60	0.75
macro avg	0.95	0.93	0.93
weighted avg	0.94	0.94	0.94
accuracy	0.9368421052631579		
balanced accuracy	0.9289115646258503		

The classifier performance indicates that using preprocessing defences causes a drop in the measured metrics, at the same time, the achieved robustness is considerable. The results of the experiments prove that input transformations are an effective weapon against adversarial attacks, though the robustness comes at a cost. The utility of the proposed preprocessing pipeline solution comes in the fact that it can be used as a plug-and-play quick-fix, granting a measure of robustness against adversarial attacks without having to incur the costs of re-training the classifier.

Chapter 6 Summary and Conclusion

This document proposed solutions to evaluate the adversarial machine learning methods within the SPARTA WP7 SAFAIR program. One is designing an adversarial machine learning contest to test participants' attacks and defences solutions which is an intermediate solution. The design of the contest proved to be working and can be recommended.

The next one is implementing an adversarial ML benchmark tool that helps researchers and developers to design and implement more robust ML models and present standardised benchmarks of proposed solutions in the area of adversarial machine learning. The adversarial ML benchmark tool is functional and can be recommended by SPARTA to use in the wider community. However, the AI contest and the benchmark tool solutions are generic and adaptive, which can be used for lots of different adversarial ML scenarios (not just SAFAIR ML scenarios) for evaluations.

Last but not least, the SAFAIR AI Threat KB evaluation results demonstrates that the great majority of the evaluators agreed on the high quality, correctness, completeness, usefulness and re-usability of the contents of the Knowledge Base and the KB does not present any error or gap of information, providing easy to understand descriptions as well as complete references. Most of the evaluators considered the information about threats and countermeasures to be relevant and useful for developers to improve the security and privacy of ML models.

To this end, the verification of ML models robustness is at the beginning of the pathway because the algorithms and techniques have presumptions that avoid them presenting full guarantees of not having any adversarial examples. Consequently, we hope our readers will be inspired to solve some of the challenges mentioned in this document.

Chapter 7 List of Abbreviations

Abbreviation	Translation
AI	Artificial Intelligence
ML	Machine Learning
AML	Adversarial Machine Learning
FGSM	Fast Gradient Sign Method attack
Iter-FGSM	iterative Fast Gradient Sign Method attack (same as BIM – Basic Iterative Method)
C&W	Carlini and Wagner attack
NN	Neural Network
DNN	Deep Neural Network
CNN	Convolutional Neural Network
ROC	Receiver Operating Characteristic

Chapter 8 Bibliography

[1] L. Huang, A. D. Joseph, B. Nelson, B. Rubinstein, and J. D. Tygar. “Adversarial machine learning.” In 4th ACM Workshop AISec, pages 43–57, Chicago, IL, USA, 2011.

[2] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndic, P. Laskov, G. Giacinto, and F. Roli. “Evasion attacks against machine learning at test time.” In ECML PKDD, Part III, volume 8190 of LNCS, pages 387–402. Springer Berlin Heidelberg, 2013.

[3] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. “Intriguing properties of neural networks.” In ICLR, 2014.

[4] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. “The limitations of deep learning in adversarial settings.” In 1st IEEE Euro SP, pages 372–387. IEEE, 2016.

[5] B. Biggio and F. Roli. Wild patterns: “Ten years after the rise of adversarial machine learning. Pattern Recognition”, 84:317–33, 2018.

[6] A. D. Joseph, B. Nelson, B. I. P. Rubinstein, and J. Tygar. “Adversarial Machine Learning.” Cambridge University Press, 2018.

[7] Brendel, W., Rauber, J., and Bethge, M. (2017). “Decision-based adversarial attacks: Reliable attacks against black-box machine learning models.” 2017.

[8] Xiao, C., Zhu, J.-Y., Li, B., He, W., Liu, M., and Song, D. “Spatially transformed adversarial examples.” 2018.

[9] Metzen, Jan Hendrik, et al. "On detecting adversarial perturbations." arXiv preprint arXiv:1702.04267 (2017).

[10] Guo, Chuan, et al. "Countering adversarial images using input transformations." arXiv preprint arXiv:1711.00117 (2017).

[11] Kurakin, Alexey, Ian Goodfellow, and Samy Bengio. "Adversarial machine learning at scale." arXiv preprint arXiv:1611.01236 (2016).

[12] Shafahi, Ali, et al. "Are adversarial examples inevitable?." arXiv preprint arXiv:1809.02104 (2018).

[13] Ding, Gavin Weiguang, et al. "On the Sensitivity of Adversarial Robustness to Input Data Distributions." ICLR (Poster). 2019.

[14] Huang, Gary B., and Erik Learned-Miller. "Labeled faces in the wild: Updates and new reporting procedures." Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep 14.003 (2014).

[15] N. Ateqah, B. Mat, N. Hidayah, B. Abd, and Z. Ibrahim, “Celebrity Face Recognition using Deep Learning,” vol. 12, no. 2, pp. 476–481, 2018, doi: 10.11591/ijeecs.v12.i2.pp476-481.

[16] https://git.sec.in.tum.de/Norouzian/safair-ai-contest/-/tree/master/dev_toolkit

[17] Sahay, Rajeev, Rehana Mahfuz, and Aly El Gamal. "Combatting adversarial attacks through denoising and dimensionality reduction: A cascaded autoencoder approach." 2019 53rd Annual conference on information sciences and systems (CISS). IEEE, 2019.

[18] https://www.sec.in.tum.de/i20/projects/sparta-safair-ai-contest

[19] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." International Conference on Machine Learning. PMLR, 2019.

[20] Wang, Qilong, et al. "ECA-Net: efficient channel attention for deep convolutional neural networks, 2020 IEEE." CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE. 2020.

[21] Zheng, Haizhong, et al. "Efficient adversarial training with transferable adversarial examples." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

[22] Echeberria-Barrio, Xabier, et al. "Deep Learning Defenses Against Adversarial Examples for Dynamic Risk Assessment." Conference on Complex, Intelligent, and Software Intensive Systems. Springer, Cham, 2020.

[23] Sahay, Rajeev, Rehana Mahfuz, and Aly El Gamal. "Combatting adversarial attacks through denoising and dimensionality reduction: A cascaded autoencoder approach." 2019 53rd Annual conference on information sciences and systems (CISS). IEEE, 2019.

[24] Howard, Andrew, et al. "Searching for mobilenetv3." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.

[25] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. “Towards deep learning models resistant to adversarial attacks.” arXiv preprint arXiv:1706.06083, 2017.

[26] Jérôme Rony, Luiz G Hafemann, Luiz S Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. “Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses.” arXiv preprint arXiv:1811.09600, 2018.

[27] Pawlicki, Marek, and Ryszard S. Choraś. "Preprocessing Pipelines including Block-Matching Convolutional Neural Network for Image Denoising to Robustify Deep Reidentification against Evasion Attacks." Entropy 23, no. 10 (2021): 1304.

[28] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. “Intriguing properties of neural networks.” In International Conference on Learning Representations, 2014. URL http://arxiv.org/abs/1312.6199

[29] Anish Athalye, Nicholas Carlini, and David A. Wagner. “Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples.” CoRR, abs/1802.00420, 2018b. URL http://arxiv.org/abs/1802.00420.

[30] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. “Towards deep learning models resistant to adversarial attacks.” In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, 2018.

[31] Nicholas Carlini and David A. Wagner. “Towards evaluating the robustness of neural networks.” CoRR, abs/1608.04644, 2016.

[32] Papernot, Nicolas, et al. "Practical black-box attacks against machine learning." Proceedings of the 2017 ACM on Asia conference on computer and communications security. 2017.

[33] J. Rauber, W. Brendel, and M. Bethge, “Foolbox: A Python toolbox to benchmark the robustness of machine learning models,” arXiv. 2017.

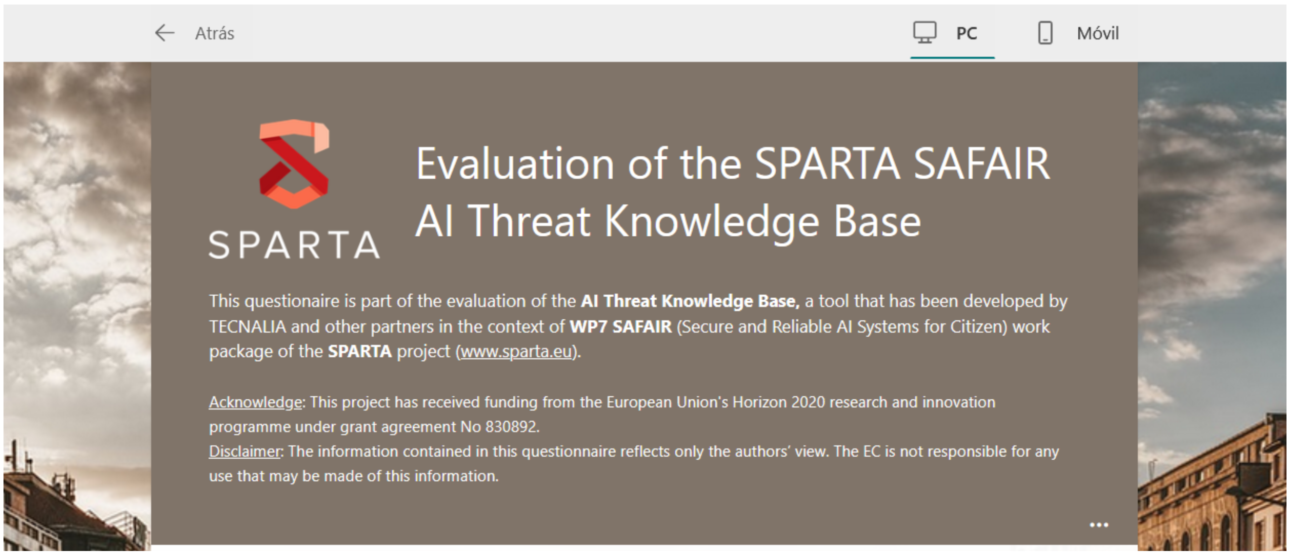
[34] Kurakin, Alexey, et al. "Adversarial attacks and defences competition." The NIPS'17 Competition: Building Intelligent Systems. Springer, Cham, 2018. 195-231.

[35] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).

[36] Papernot, Nicolas, Patrick McDaniel, and Ian Goodfellow. "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples." arXiv preprint arXiv:1605.07277 (2016).

Appendix A AI Threat Knowledge Base evaluation questionnaire

This section transcripts the online questionnaire that was designed for getting the feedback of the evaluators of the SAFAIR AI Threat Knowledge Base. The figures below represent the different sections in which the questions were grouped.

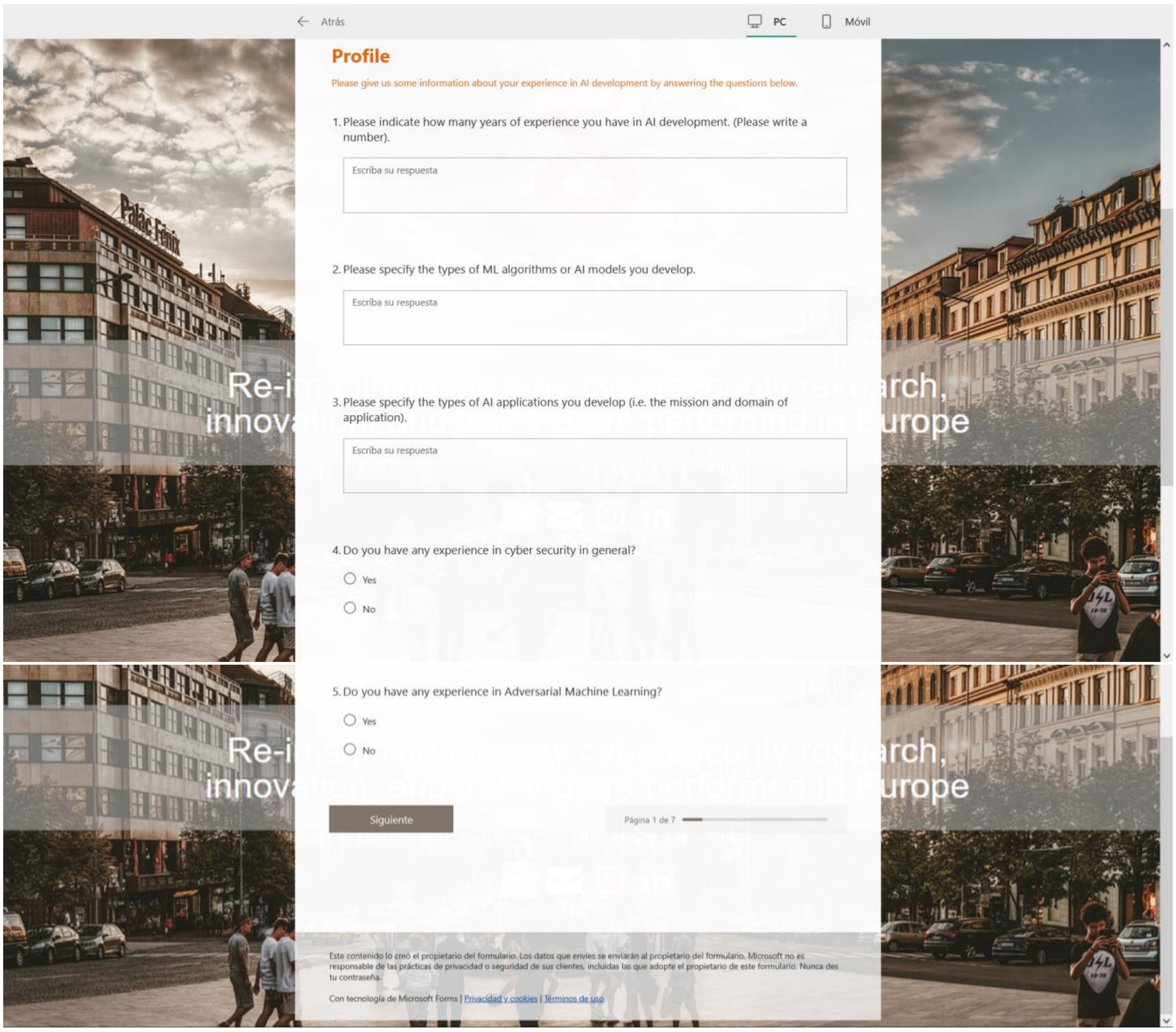


Evaluation of the SPARTA SAFAIR AI Threat Knowledge Base

This questionaire is part of the evaluation of the **AI Threat Knowledge Base,** a tool that has been developed by TECNALIA and other partners in the context of **WP7 SAFAIR** (Secure and Reliable AI Systems for Citizen) work package of the **SPARTA** project (www.sparta.eu).

Acknowledge: This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 830892.
Disclaimer: The information contained in this questionnaire reflects only the authors' view. The EC is not responsible for any use that may be made of this information.

Figure 11: Introduction to the Questionnaire


Atrás
PC
Móvil
Re-imagining the way cybersecurity research, innovation, and training are performed in Europe
Siguiente
Página 1 de 7

Profile

Please give us some information about your experience in AI development by answering the questions below.

1. Please indicate how many years of experience you have in AI development. (Please write a number).

Escriba su respuesta

2. Please specify the types of ML algorithms or AI models you develop.

Escriba su respuesta

3. Please specify the types of AI applications you develop (i.e. the mission and domain of application).

Escriba su respuesta

4. Do you have any experience in cyber security in general?

Yes
No

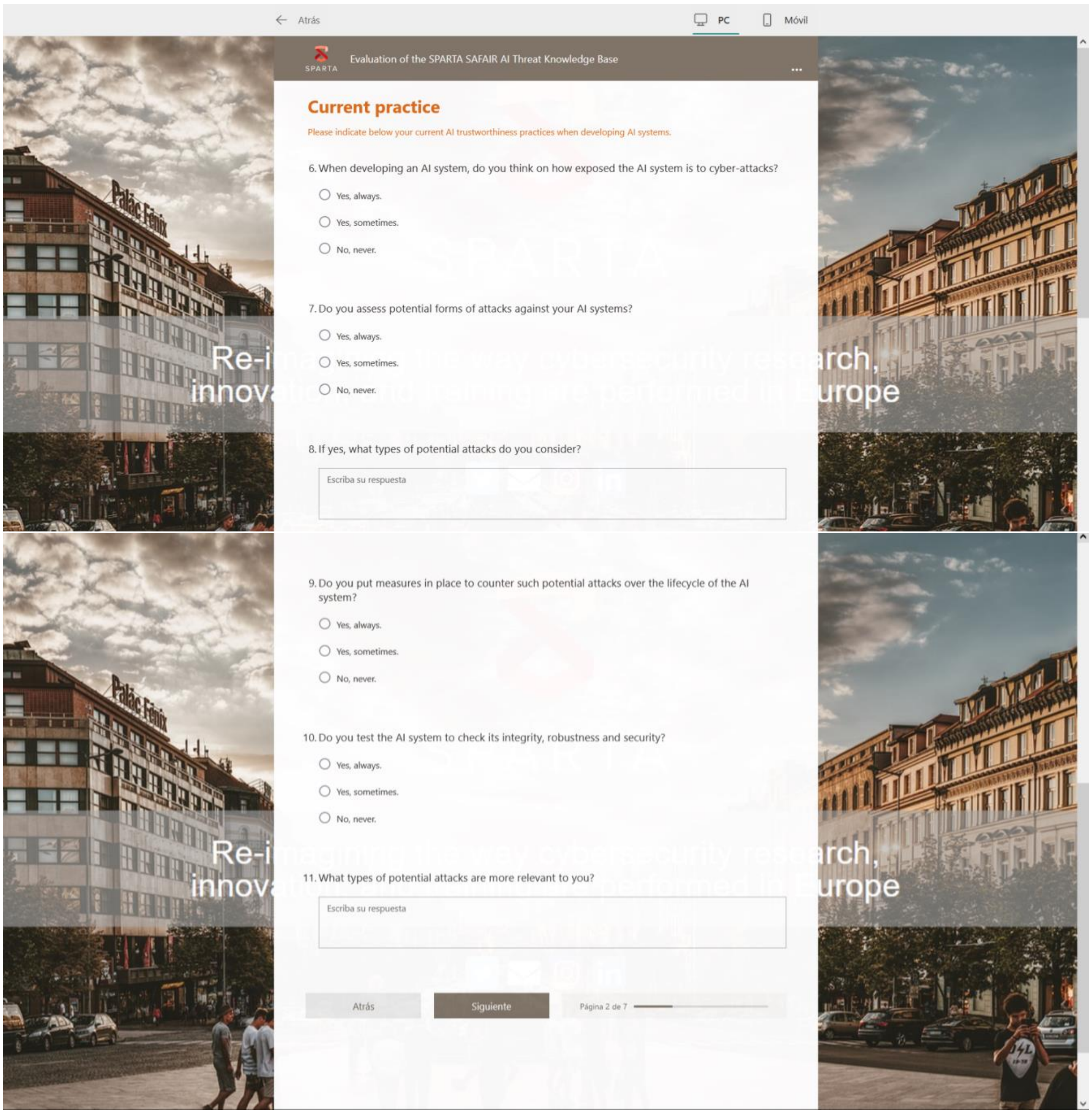
5. Do you have any experience in Adversarial Machine Learning?

Yes
No

Este contenido lo creó el propietario del formulario. Los datos que envíes se enviarán al propietario del formulario. Microsoft no es responsable de las prácticas de privacidad o seguridad de sus clientes, incluidas las que adopte el propietario de este formulario. Nunca des tu contraseña.

Con tecnología de Microsoft Forms | Privacidad y cookies | Términos de uso

Figure 12: Section 1 - Profile


Atrás
PC
Móvil
SPARTA
Re-imagining the way cybersecurity research, innovation and training are performed in Europe

Evaluation of the SPARTA SAFAIR AI Threat Knowledge Base

Current practice

Please indicate below your current AI trustworthiness practices when developing AI systems.

6. When developing an AI system, do you think on how exposed the AI system is to cyber-attacks?

- ○ Yes, always.
- ○ Yes, sometimes.
- ○ No, never.

7. Do you assess potential forms of attacks against your AI systems?

- ○ Yes, always.
- ○ Yes, sometimes.
- ○ No, never.

8. If yes, what types of potential attacks do you consider?

Escriba su respuesta

9. Do you put measures in place to counter such potential attacks over the lifecycle of the AI system?

- ○ Yes, always.
- ○ Yes, sometimes.
- ○ No, never.

10. Do you test the AI system to check its integrity, robustness and security?

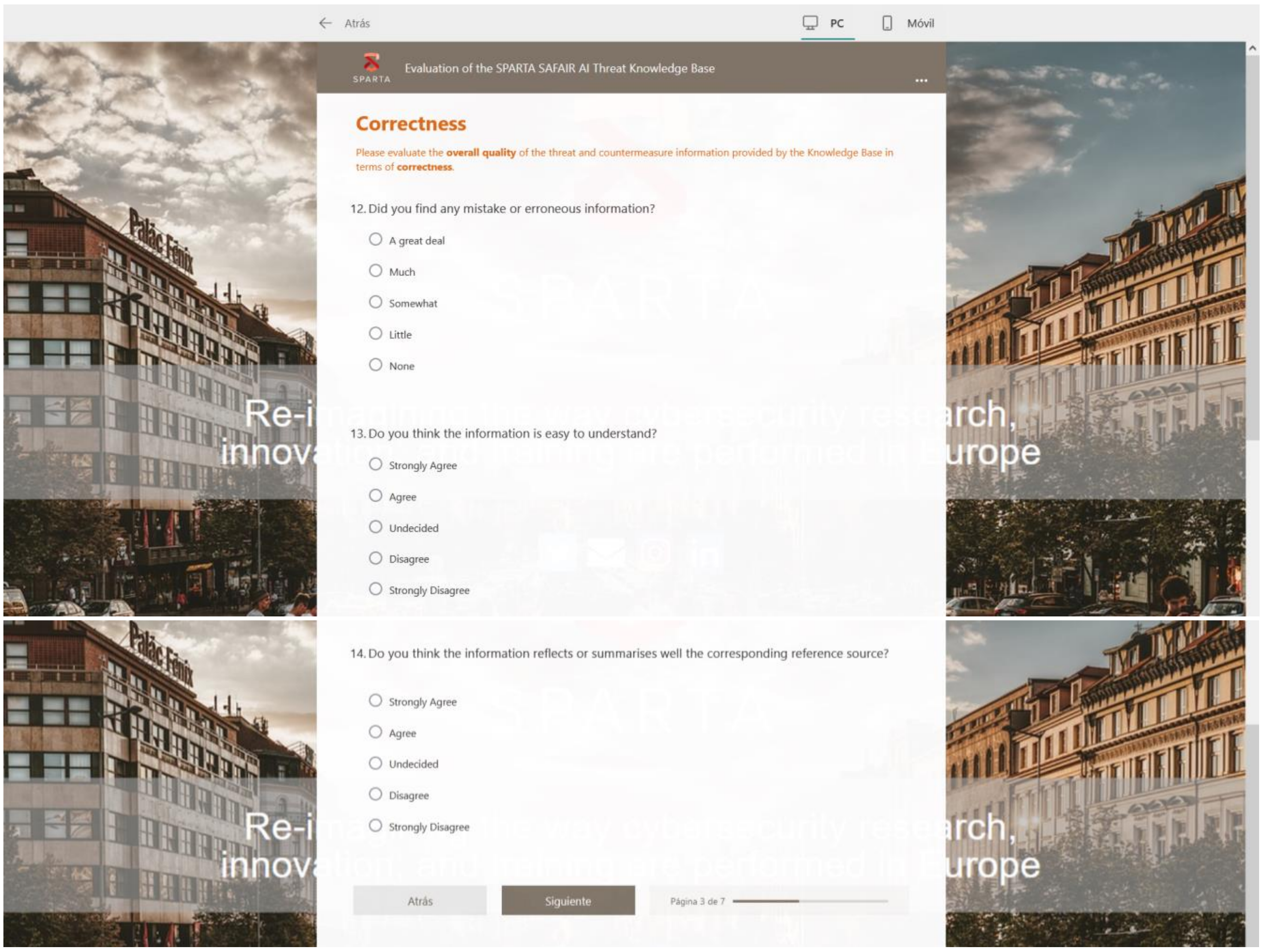
- ○ Yes, always.
- ○ Yes, sometimes.
- ○ No, never.

11. What types of potential attacks are more relevant to you?

Escriba su respuesta

Atrás | Siguiente | Página 2 de 7

Figure 13: Section 2 – Current Practice


Atrás
PC
Móvil
SPARTA
Re-imagining the way cybersecurity research, innovation and training are performed in Europe

Evaluation of the SPARTA SAFAIR AI Threat Knowledge Base

Correctness

Please evaluate the **overall quality** of the threat and countermeasure information provided by the Knowledge Base in terms of **correctness**.

12. Did you find any mistake or erroneous information?

- ○ A great deal
- ○ Much
- ○ Somewhat
- ○ Little
- ○ None

13. Do you think the information is easy to understand?

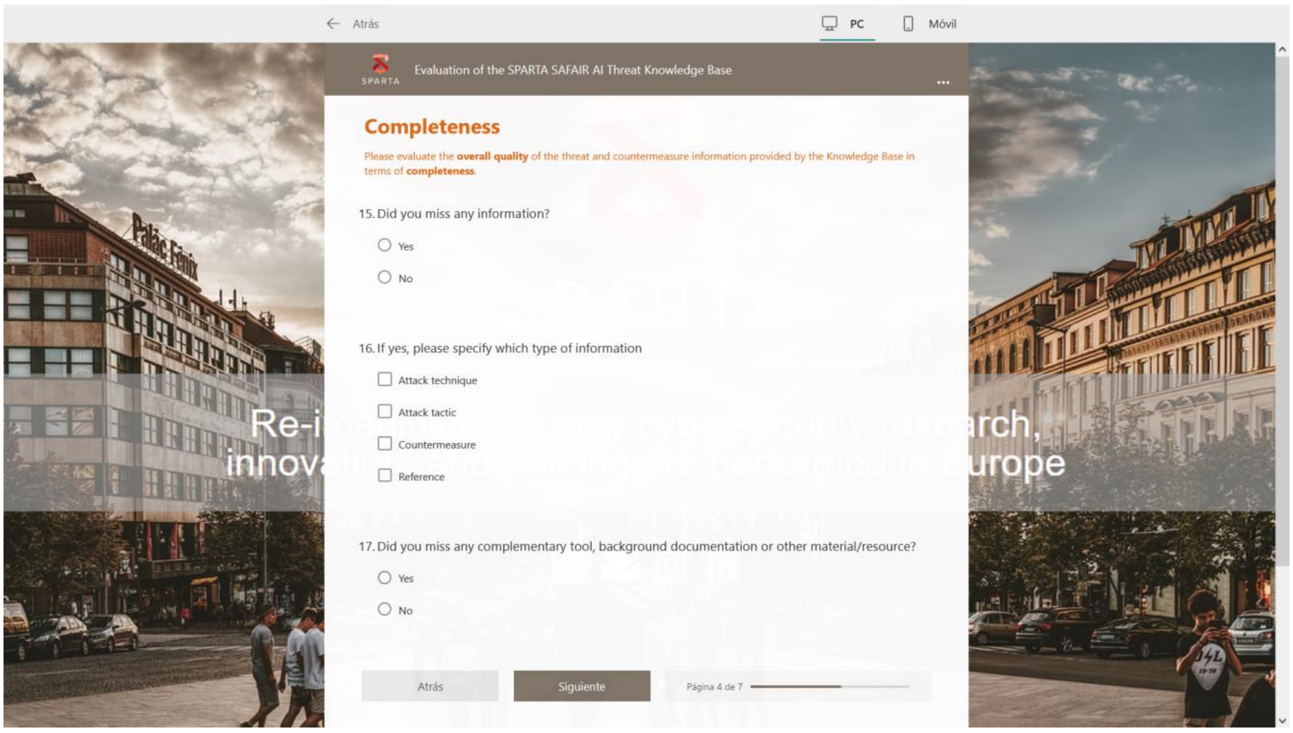
- ○ Strongly Agree
- ○ Agree
- ○ Undecided
- ○ Disagree
- ○ Strongly Disagree

14. Do you think the information reflects or summarises well the corresponding reference source?

- ○ Strongly Agree
- ○ Agree
- ○ Undecided
- ○ Disagree
- ○ Strongly Disagree

Atrás | Siguiente | Página 3 de 7

Figure 14: Section 3 – Correctness


Atrás
PC
Móvil
SPARTA
Re-imagining the way cybersecurity research, innovation and training are performed in Europe

Evaluation of the SPARTA SAFAIR AI Threat Knowledge Base

Completeness

Please evaluate the **overall quality** of the threat and countermeasure information provided by the Knowledge Base in terms of **completeness**.

15. Did you miss any information?

- ○ Yes
- ○ No

16. If yes, please specify which type of information

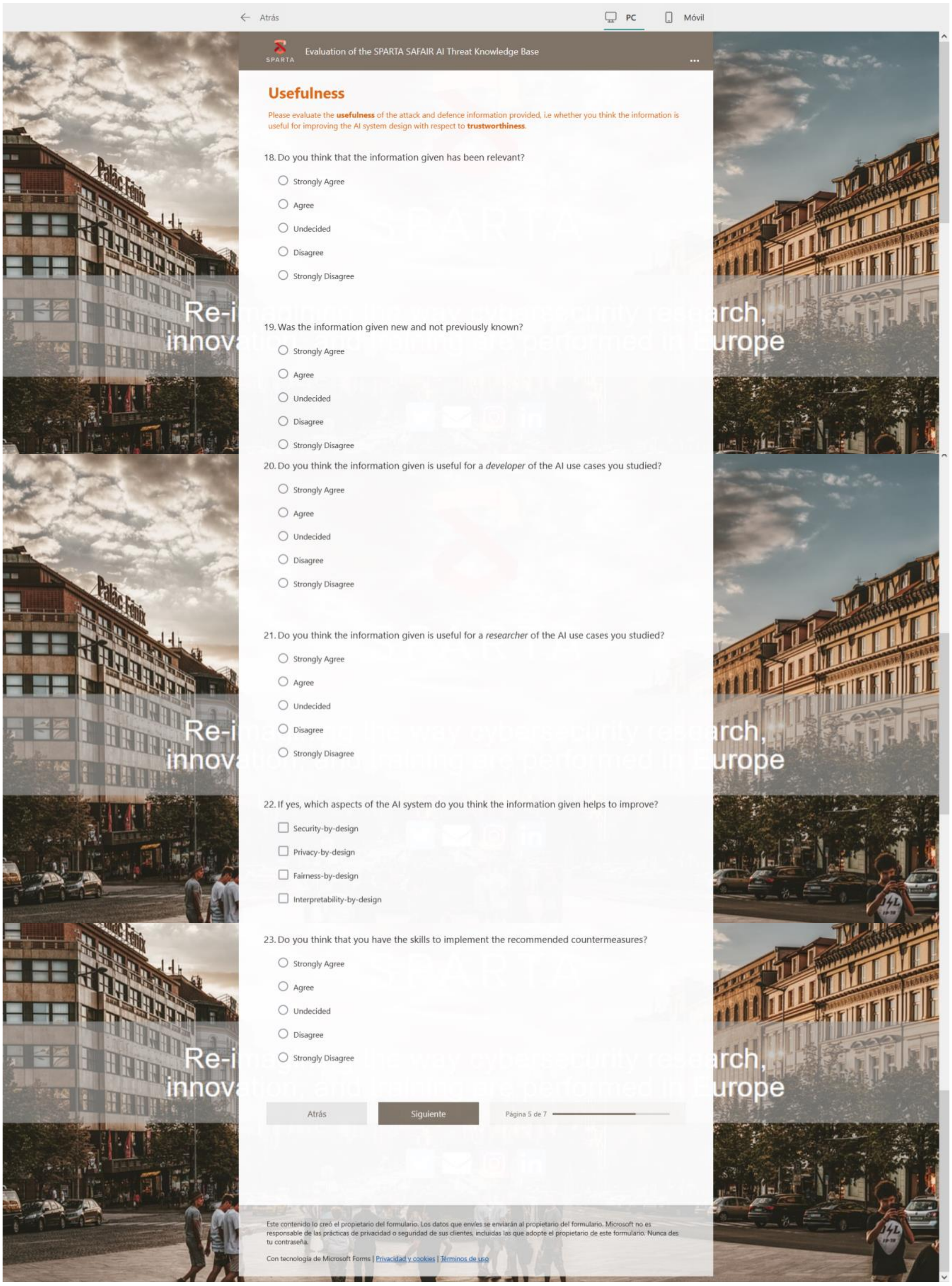
- ☐ Attack technique
- ☐ Attack tactic
- ☐ Countermeasure
- ☐ Reference

17. Did you miss any complementary tool, background documentation or other material/resource?

- ○ Yes
- ○ No

Atrás | Siguiente | Página 4 de 7

Figure 15: Section 4 – Completeness



Evaluation of the SPARTA SAFAIR AI Threat Knowledge Base

Usefulness

Please evaluate the **usefulness** of the attack and defence information provided, i.e whether you think the information is useful for improving the AI system design with respect to **trustworthiness**.

18. Do you think that the information given has been relevant?

- ○ Strongly Agree
- ○ Agree
- ○ Undecided
- ○ Disagree
- ○ Strongly Disagree

19. Was the information given new and not previously known?

- ○ Strongly Agree
- ○ Agree
- ○ Undecided
- ○ Disagree
- ○ Strongly Disagree

20. Do you think the information given is useful for a *developer* of the AI use cases you studied?

- ○ Strongly Agree
- ○ Agree
- ○ Undecided
- ○ Disagree
- ○ Strongly Disagree

21. Do you think the information given is useful for a *researcher* of the AI use cases you studied?

- ○ Strongly Agree
- ○ Agree
- ○ Undecided
- ○ Disagree
- ○ Strongly Disagree

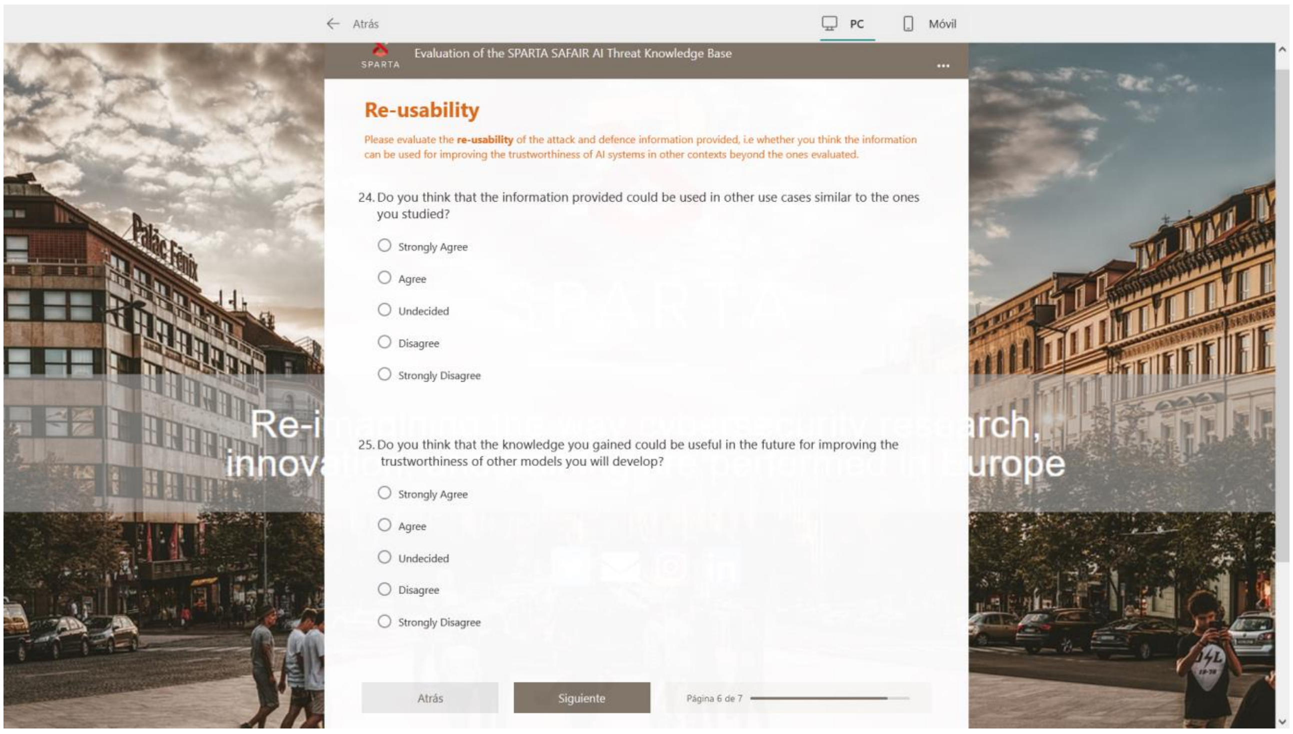
22. If yes, which aspects of the AI system do you think the information given helps to improve?

- ☐ Security-by-design
- ☐ Privacy-by-design
- ☐ Fairness-by-design
- ☐ Interpretability-by-design

23. Do you think that you have the skills to implement the recommended countermeasures?

- ○ Strongly Agree
- ○ Agree
- ○ Undecided
- ○ Disagree
- ○ Strongly Disagree

Figure 16: Section 5 – Usefulness


Atrás
PC
Móvil
SPARTA
Re-imagining the way cybersecurity research, innovation ... Europe

Evaluation of the SPARTA SAFAIR AI Threat Knowledge Base

Re-usability

Please evaluate the **re-usability** of the attack and defence information provided, i.e whether you think the information can be used for improving the trustworthiness of AI systems in other contexts beyond the ones evaluated.

24. Do you think that the information provided could be used in other use cases similar to the ones you studied?

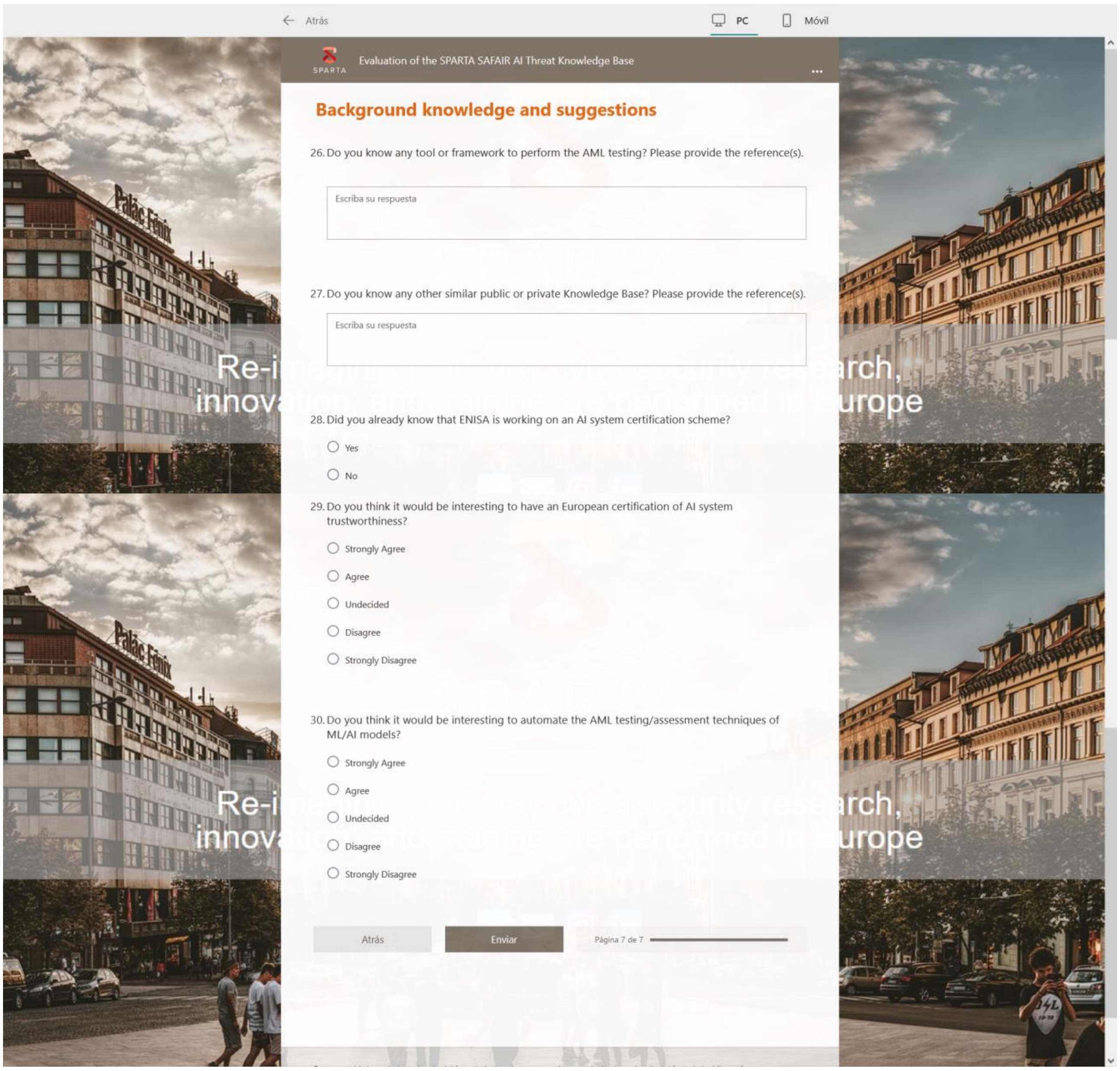
- Strongly Agree
- Agree
- Undecided
- Disagree
- Strongly Disagree

25. Do you think that the knowledge you gained could be useful in the future for improving the trustworthiness of other models you will develop?

- Strongly Agree
- Agree
- Undecided
- Disagree
- Strongly Disagree

Atrás | Siguiente | Página 6 de 7

Figure 17: Section 6 – Re-usability


Atrás
PC
Móvil
SPARTA
Re-imagining the way cybersecurity research, innovation, and training are performed in Europe

Evaluation of the SPARTA SAFAIR AI Threat Knowledge Base

Background knowledge and suggestions

26. Do you know any tool or framework to perform the AML testing? Please provide the reference(s).

Escriba su respuesta

27. Do you know any other similar public or private Knowledge Base? Please provide the reference(s).

Escriba su respuesta

28. Did you already know that ENISA is working on an AI system certification scheme?

○ Yes
○ No

29. Do you think it would be interesting to have an European certification of AI system trustworthiness?

○ Strongly Agree
○ Agree
○ Undecided
○ Disagree
○ Strongly Disagree

30. Do you think it would be interesting to automate the AML testing/assessment techniques of ML/AI models?

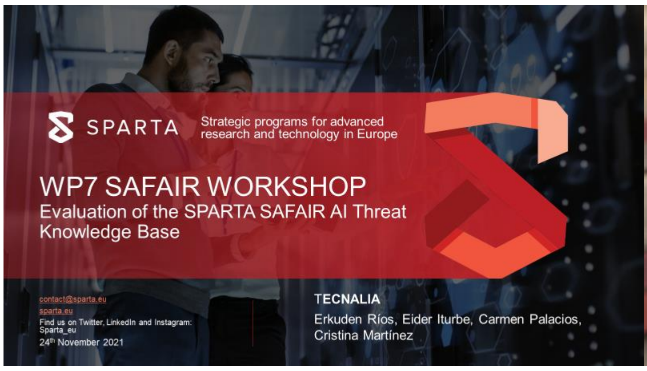
○ Strongly Agree
○ Agree
○ Undecided
○ Disagree
○ Strongly Disagree

Atrás | Enviar | Página 7 de 7

Figure 18: Section 7 – Background knowledge and suggestions

Appendix B AI Threat Knowledge Base evaluation presentation

This section transcripts the contents of the presentation that was used in the SAFAIR Training workshop. The figures below represent the different slides of the presentation.

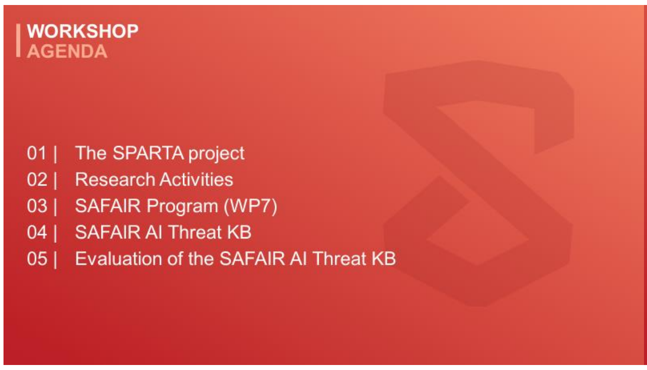


SPARTA Strategic programs for advanced research and technology in Europe

WP7 SAFAIR WORKSHOP
Evaluation of the SPARTA SAFAIR AI Threat Knowledge Base

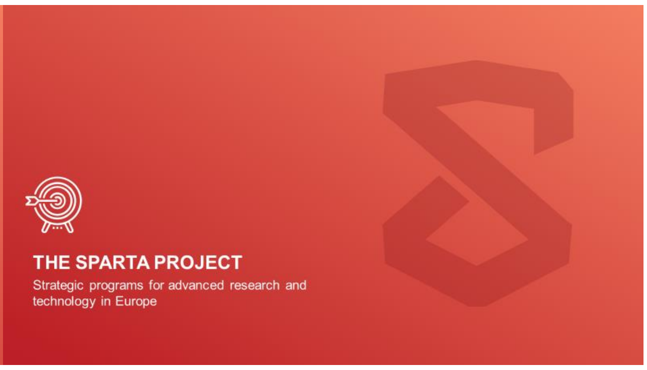
contact@sparta.eu
sparta.eu
Find us on Twitter, LinkedIn and Instagram: Sparta_eu
24th November 2021

TECNALIA
Erkuden Rios, Eider Iturbe, Carmen Palacios, Cristina Martinez

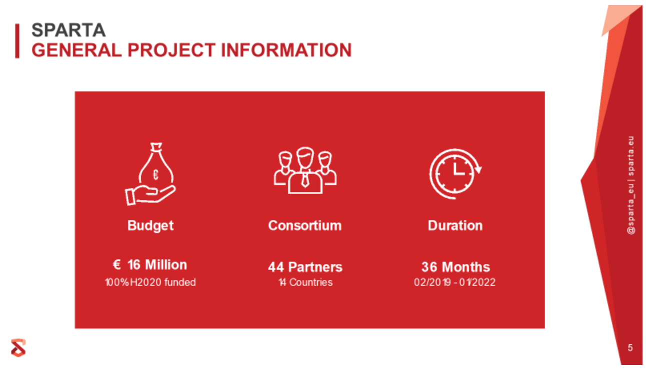


WORKSHOP
AGENDA

01 | The SPARTA project
02 | Research Activities
03 | SAFAIR Program (WP7)
04 | SAFAIR AI Threat KB
05 | Evaluation of the SAFAIR AI Threat KB

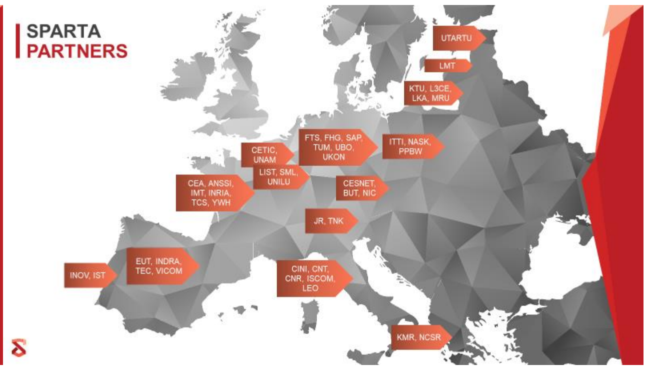


THE SPARTA PROJECT
Strategic programs for advanced research and technology in Europe



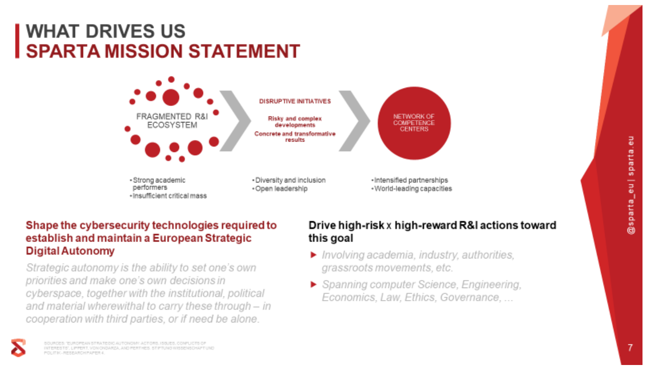
SPARTA
GENERAL PROJECT INFORMATION

Budget	Consortium	Duration
€ 16 Million	44 Partners	36 Months
100% H2020 funded	14 Countries	02/2019 - 01/2022



SPARTA
PARTNERS

5



WHAT DRIVES US
SPARTA MISSION STATEMENT

FRAGMENTED R&I ECOSYSTEM
DISRUPTIVE INITIATIVES
Risky and complex developments
Concrete and transformative results
NETWORK OF COMPETENCE CENTERS

- Strong academic performers
- Insufficient critical mass

- Diversity and inclusion
- Open leadership

- Intensified partnerships
- World-leading capacities

Shape the cybersecurity technologies required to establish and maintain a European Strategic Digital Autonomy

Strategic autonomy is the ability to set one's own priorities and make one's own decisions in cyberspace, together with the institutional, political and material wherewithal to carry these through – in cooperation with third parties, or if need be alone.

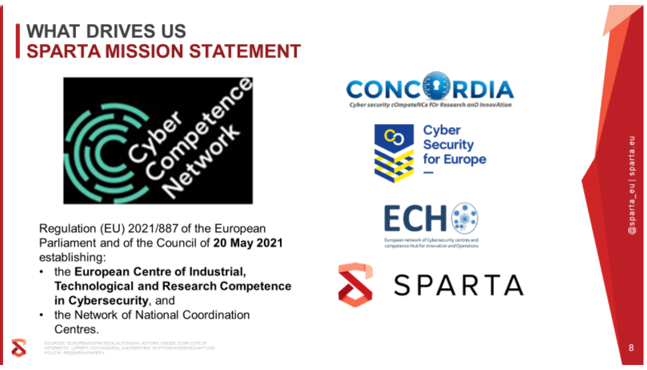
Drive high-risk x high-reward R&I actions toward this goal

- *Involving academia, industry, authorities, grassroots movements, etc.*
- *Spanning computer Science, Engineering, Economics, Law, Ethics, Governance, …*

SOURCES: "EUROPEAN STRATEGIC AUTONOMY: ACTORS, ISSUES, CONFLICTS OF INTERESTS", LIPPERT, VON ONDARZA, AND PERTHES. STIFTUNG WISSENSCHAFT UND POLITIK - RESEARCH PAPER 4.

@sparta_eu | sparta.eu

7



WHAT DRIVES US
SPARTA MISSION STATEMENT

Cyber Competence Network
CONCORDIA
Cyber security cOmpeteNCe fOr Research anD InnovAtion
Cyber Security for Europe
ECHO
European network of Cybersecurity centres and competence Hub for innovation and Operations
SPARTA

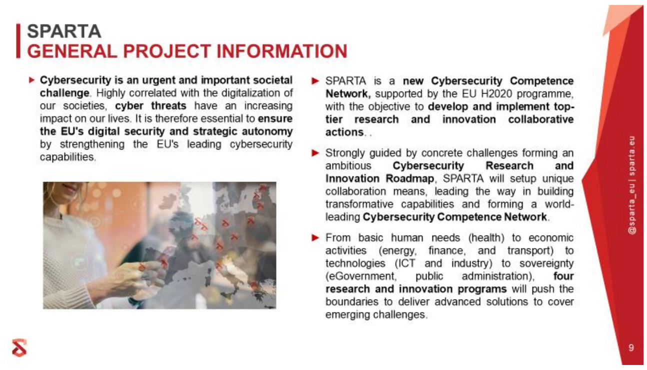
Regulation (EU) 2021/887 of the European Parliament and of the Council of **20 May 2021** establishing:

- the **European Centre of Industrial, Technological and Research Competence in Cybersecurity**, and
- the Network of National Coordination Centres.

SOURCES: "EUROPEAN STRATEGIC AUTONOMY: ACTORS, ISSUES, CONFLICTS OF INTERESTS", LIPPERT, VON ONDARZA, AND PERTHES. STIFTUNG WISSENSCHAFT UND POLITIK - RESEARCH PAPER 4.

@sparta_eu | sparta.eu

8

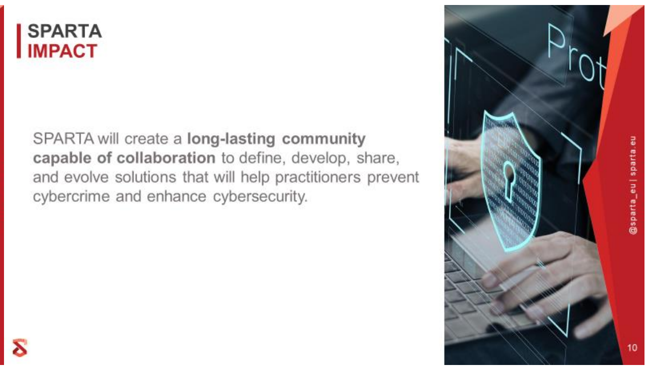


SPARTA GENERAL PROJECT INFORMATION

- **Cybersecurity is an urgent and important societal challenge**. Highly correlated with the digitalization of our societies, **cyber threats** have an increasing impact on our lives. It is therefore essential to **ensure the EU's digital security and strategic autonomy** by strengthening the EU's leading cybersecurity capabilities.
- SPARTA is a **new Cybersecurity Competence Network,** supported by the EU H2020 programme, with the objective to **develop and implement top-tier research and innovation collaborative actions**..
- Strongly guided by concrete challenges forming an ambitious **Cybersecurity Research and Innovation Roadmap**, SPARTA will setup unique collaboration means, leading the way in building transformative capabilities and forming a world-leading **Cybersecurity Competence Network**.
- From basic human needs (health) to economic activities (energy, finance, and transport) to technologies (ICT and industry) to sovereignty (eGovernment, public administration), **four research and innovation programs** will push the boundaries to deliver advanced solutions to cover emerging challenges.

@sparta_eu | sparta.eu

9

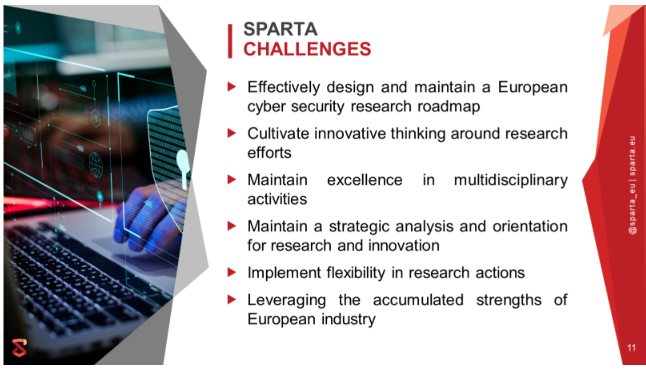


SPARTA IMPACT

SPARTA will create a **long-lasting community capable of collaboration** to define, develop, share, and evolve solutions that will help practitioners prevent cybercrime and enhance cybersecurity.

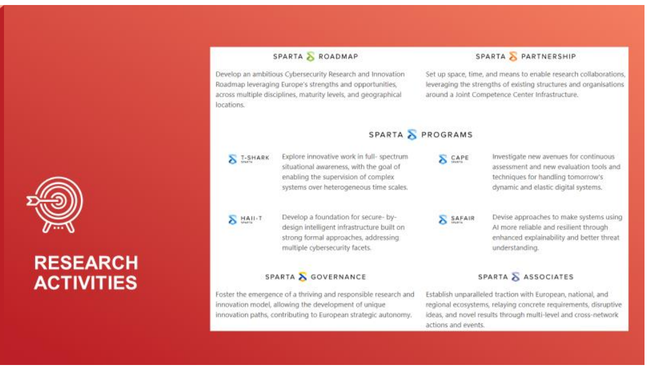
@sparta_eu | sparta.eu

10



SPARTA CHALLENGES

- Effectively design and maintain a European cyber security research roadmap
- Cultivate innovative thinking around research efforts
- Maintain excellence in multidisciplinary activities
- Maintain a strategic analysis and orientation for research and innovation
- Implement flexibility in research actions
- Leveraging the accumulated strengths of European industry



RESEARCH ACTIVITIES

SPARTA ROADMAP

Develop an ambitious Cybersecurity Research and Innovation Roadmap leveraging Europe's strengths and opportunities, across multiple disciplines, maturity levels, and geographical locations.

SPARTA PARTNERSHIP

Set up space, time, and means to enable research collaborations, leveraging the strengths of existing structures and organisations around a Joint Competence Center Infrastructure.

SPARTA PROGRAMS

T-SHARK	Explore innovative work in full- spectrum situational awareness, with the goal of enabling the supervision of complex systems over heterogeneous time scales.
CAPE	Investigate new avenues for continuous assessment and new evaluation tools and techniques for handling tomorrow's dynamic and elastic digital systems.
HAII-T	Develop a foundation for secure- by-design intelligent infrastructure built on strong formal approaches, addressing multiple cybersecurity facets.
SAFAIR	Devise approaches to make systems using AI more reliable and resilient through enhanced explainability and better threat understanding.

SPARTA GOVERNANCE

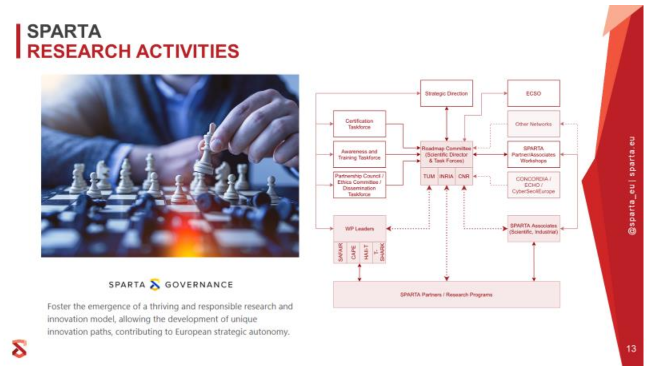
Foster the emergence of a thriving and responsible research and innovation model, allowing the development of unique innovation paths, contributing to European strategic autonomy.

SPARTA ASSOCIATES

Establish unparalleled traction with European, national, and regional ecosystems, relaying concrete requirements, disruptive ideas, and novel results through multi-level and cross-network actions and events.

@sparta_eu | sparta.eu

11



SPARTA RESEARCH ACTIVITIES

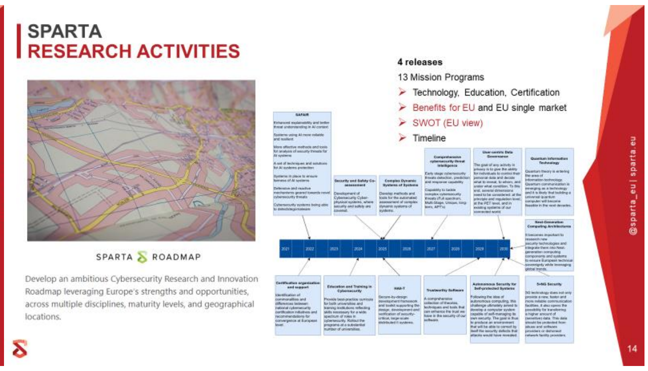
SPARTA GOVERNANCE

Foster the emergence of a thriving and responsible research and innovation model, allowing the development of unique innovation paths, contributing to European strategic autonomy.

Strategic Direction
ECSO
Certification Taskforce
Other Networks
Awareness and Training Taskforce
Roadmap Committee (Scientific Director & Task Forces)
SPARTA Partner/Associates Workshops
Partnership Council / Ethics Committee / Dissemination Taskforce
TUM
INRIA
CNR
CONCORDIA / ECHO / CyberSec4Europe
WP Leaders
SAFAIR
CAPE
HAII-T
T-SHARK
SPARTA Associates (Scientific, Industrial)
SPARTA Partners / Research Programs

@sparta_eu | sparta.eu

13



SPARTA RESEARCH ACTIVITIES

SPARTA ROADMAP

Develop an ambitious Cybersecurity Research and Innovation Roadmap leveraging Europe's strengths and opportunities, across multiple disciplines, maturity levels, and geographical locations.

4 releases

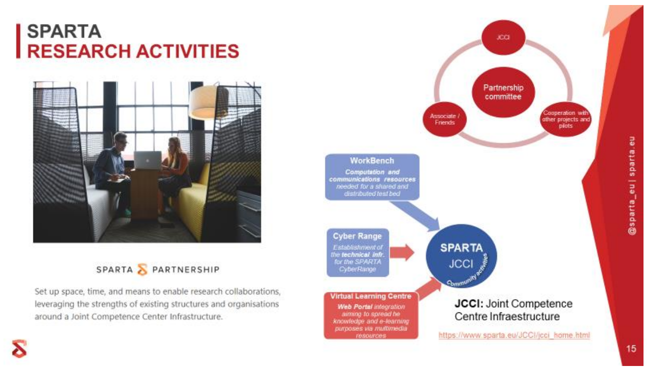
13 Mission Programs

- Technology, Education, Certification
- Benefits for EU and EU single market
- SWOT (EU view)
- Timeline

SAFAIR
Security and Safety Co-assessment
Complex Dynamic Systems of Systems
Comprehensive cybersecurity threat intelligence
User-centric Data Governance
Quantum Information Technology
Next-Generation Computing Architectures
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
Certification organization and support
Education and Training in Cybersecurity
HAII-T
Trustworthy Software
Autonomous Security for Self-protected Systems
5+NG Security

@sparta_eu | sparta.eu

14



SPARTA RESEARCH ACTIVITIES

SPARTA PARTNERSHIP

Set up space, time, and means to enable research collaborations, leveraging the strengths of existing structures and organisations around a Joint Competence Center Infrastructure.

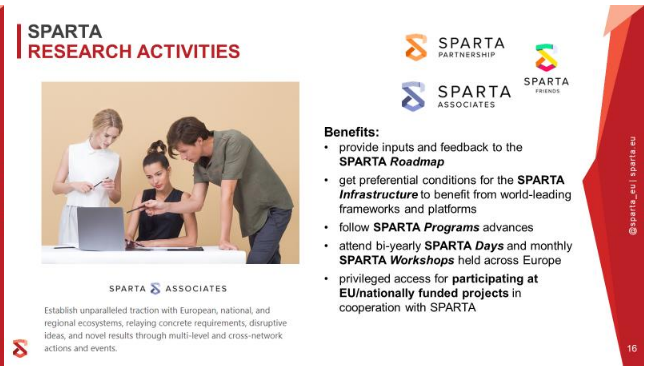
JCCI
Partnership committee
Associate / Friends
Cooperation with other projects and pilots
WorkBench
Computation and communications resources needed for a shared and distributed test bed
Cyber Range
Establishment of the technical infr. for the SPARTA CyberRange
Virtual Learning Centre
Web Portal integration aiming to spread he knowledge and e-learning purposes via multimedia resources
SPARTA
JCCI
Community activities

JCCI: Joint Competence Centre Infraestructure

https://www.sparta.eu/JCCI/jcci_home.html

@sparta_eu | sparta.eu

15



SPARTA RESEARCH ACTIVITIES

SPARTA ASSOCIATES

Establish unparalleled traction with European, national, and regional ecosystems, relaying concrete requirements, disruptive ideas, and novel results through multi-level and cross-network actions and events.

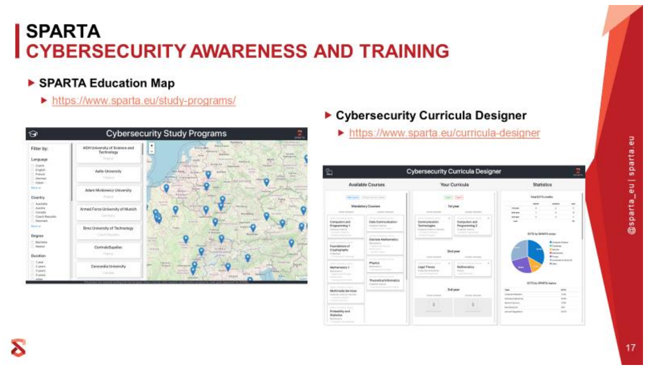
SPARTA PARTNERSHIP
SPARTA FRIENDS
SPARTA ASSOCIATES

Benefits:

- provide inputs and feedback to the **SPARTA *Roadmap***
- get preferential conditions for the **SPARTA *Infrastructure*** to benefit from world-leading frameworks and platforms
- follow **SPARTA *Programs*** advances
- attend bi-yearly **SPARTA *Days*** and monthly **SPARTA *Workshops*** held across Europe
- privileged access for **participating at EU/nationally funded projects** in cooperation with SPARTA

@sparta_eu | sparta.eu

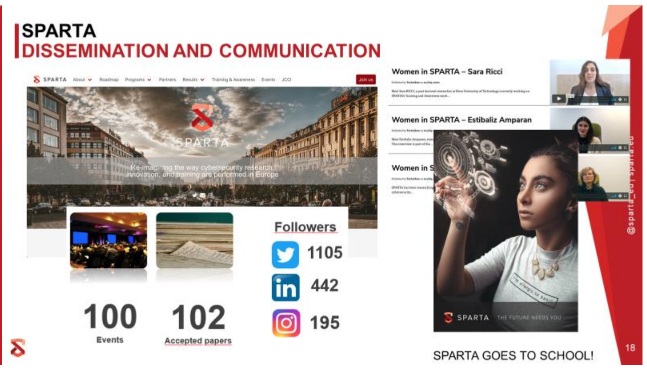
16



SPARTA CYBERSECURITY AWARENESS AND TRAINING

- SPARTA Education Map
  - https://www.sparta.eu/study-programs/
- Cybersecurity Curricula Designer
  - https://www.sparta.eu/curricula-designer

Cybersecurity Study Programs
Filter by:
Language
Country
Degree
Duration
AGH University of Science and Technology
Aalto University
Adam Mickiewicz University
Armed Forces University of Munich
Brno University of Technology
CentraleSupélec
Concordia University
Cybersecurity Curricula Designer
Available Courses
Your Curricula
Statistics
@sparta_eu | sparta.eu
17



SPARTA DISSEMINATION AND COMMUNICATION

100 Events

102 Accepted papers

Followers

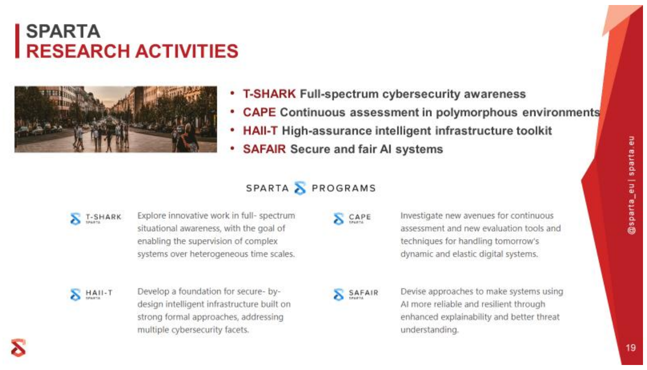
Twitter	1105
LinkedIn	442
Instagram	195

Women in SPARTA – Sara Ricci

Women in SPARTA – Estibaliz Amparan

SPARTA GOES TO SCHOOL!

SPARTA
About
Roadmap
Programs
Partners
Results
Training & Awareness
Events
JCCI
Join us
Re-imagining the way cybersecurity research, innovation, and training are performed in Europe
SPARTA THE FUTURE NEEDS YOU
@sparta_eu | sparta.eu
18



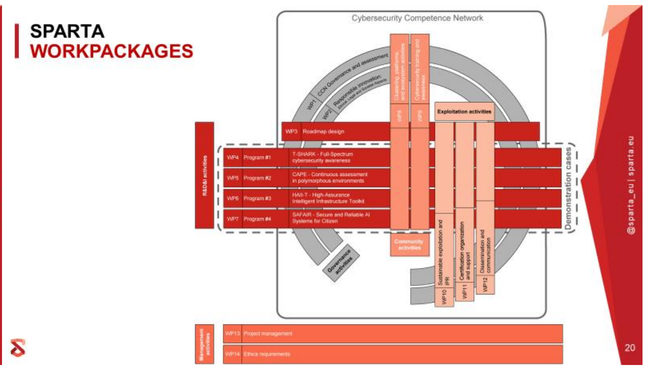
SPARTA RESEARCH ACTIVITIES

- T-SHARK Full-spectrum cybersecurity awareness
- CAPE Continuous assessment in polymorphous environments
- HAII-T High-assurance intelligent infrastructure toolkit
- SAFAIR Secure and fair AI systems

SPARTA PROGRAMS

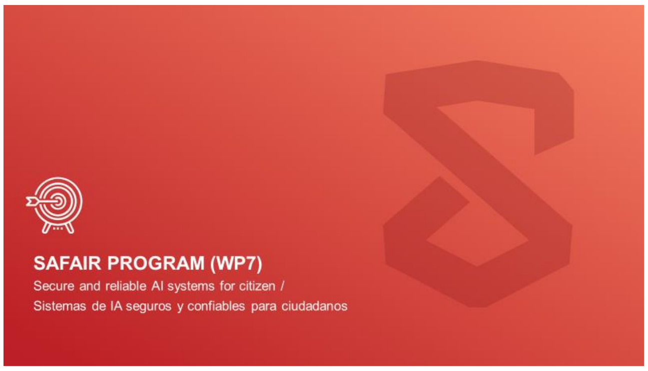
T-SHARK	Explore innovative work in full- spectrum situational awareness, with the goal of enabling the supervision of complex systems over heterogeneous time scales.
CAPE	Investigate new avenues for continuous assessment and new evaluation tools and techniques for handling tomorrow's dynamic and elastic digital systems.
HAII-T	Develop a foundation for secure- by-design intelligent infrastructure built on strong formal approaches, addressing multiple cybersecurity facets.
SAFAIR	Devise approaches to make systems using AI more reliable and resilient through enhanced explainability and better threat understanding.

@sparta_eu | sparta.eu
19



SPARTA WORKPACKAGES

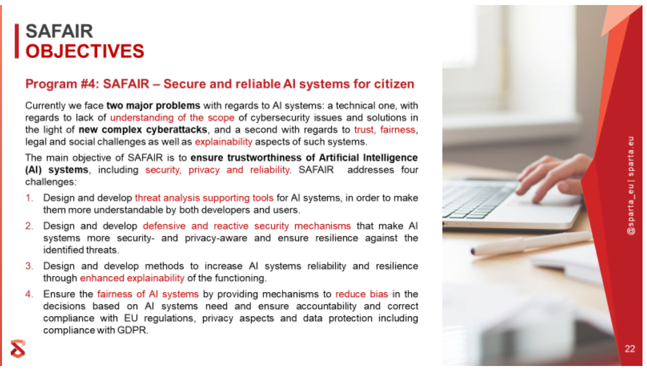
Cybersecurity Competence Network
WP1 CCN Governance and assessment
WP2 Responsible innovation: Ethical, Legal and Societal Aspects
WP8 Clustering, platforms, and ecosystem activities
WP9 Cybersecurity training and awareness
Exploitation activities
WP3 Roadmap design
R&D&I activities
WP4 Program #1 T-SHARK - Full-Spectrum cybersecurity awareness
WP5 Program #2 CAPE - Continuous assessment in polymorphous environments
WP6 Program #3 HAII-T - High-Assurance Intelligent Infrastructure Toolkit
WP7 Program #4 SAFAIR - Secure and Reliable AI Systems for Citizen
Demonstration cases
Community activities
Governance activities
WP10 Sustainable exploitation and IPR
WP11 Certification organization and support
WP12 Dissemination and communication
Management activities
WP13 Project management
WP14 Ethics requirements
@sparta_eu | sparta.eu
20



SAFAIR PROGRAM (WP7)

Secure and reliable AI systems for citizen /

Sistemas de IA seguros y confiables para ciudadanos



SAFAIR OBJECTIVES

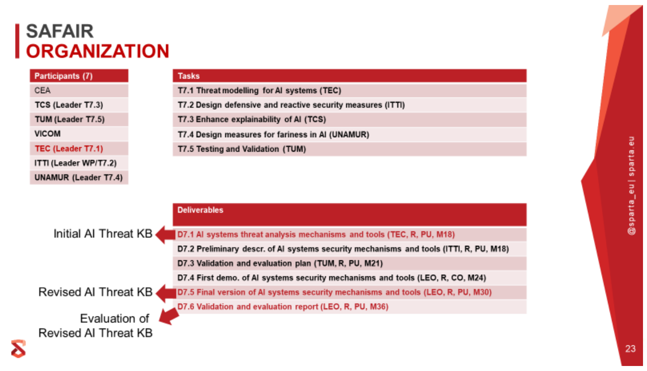
Program #4: SAFAIR – Secure and reliable AI systems for citizen

Currently we face **two major problems** with regards to AI systems: a technical one, with regards to lack of understanding of the scope of cybersecurity issues and solutions in the light of **new complex cyberattacks**, and a second with regards to trust, fairness, legal and social challenges as well as explainability aspects of such systems.

The main objective of SAFAIR is to **ensure trustworthiness of Artificial Intelligence (AI) systems**, including security, privacy and reliability. SAFAIR addresses four challenges:

1. Design and develop threat analysis supporting tools for AI systems, in order to make them more understandable by both developers and users.
2. Design and develop defensive and reactive security mechanisms that make AI systems more security- and privacy-aware and ensure resilience against the identified threats.
3. Design and develop methods to increase AI systems reliability and resilience through enhanced explainability of the functioning.
4. Ensure the fairness of AI systems by providing mechanisms to reduce bias in the decisions based on AI systems need and ensure accountability and correct compliance with EU regulations, privacy aspects and data protection including compliance with GDPR.

@sparta_eu | sparta.eu
22



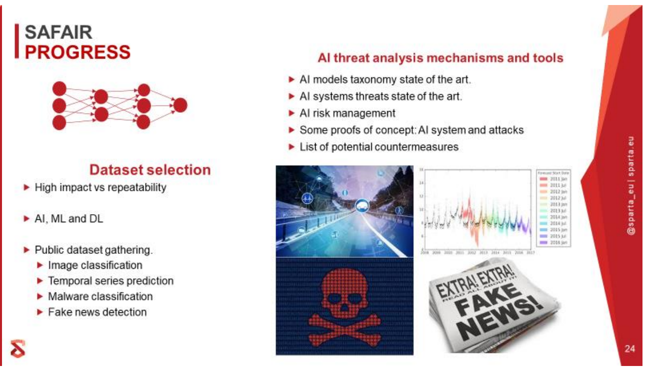
SAFAIR ORGANIZATION

Participants (7)
CEA
TCS (Leader T7.3)
TUM (Leader T7.5)
VICOM
TEC (Leader T7.1)
ITTI (Leader WP/T7.2)
UNAMUR (Leader T7.4)

Tasks
T7.1 Threat modelling for AI systems (TEC)
T7.2 Design defensive and reactive security measures (ITTI)
T7.3 Enhance explainability of AI (TCS)
T7.4 Design measures for fariness in AI (UNAMUR)
T7.5 Testing and Validation (TUM)

	Deliverables
Initial AI Threat KB	D7.1 AI systems threat analysis mechanisms and tools (TEC, R, PU, M18)
	D7.2 Preliminary descr. of AI systems security mechanisms and tools (ITTI, R, PU, M18)
	D7.3 Validation and evaluation plan (TUM, R, PU, M21)
	D7.4 First demo. of AI systems security mechanisms and tools (LEO, R, CO, M24)
Revised AI Threat KB	D7.5 Final version of AI systems security mechanisms and tools (LEO, R, PU, M30)
Evaluation of Revised AI Threat KB	D7.6 Validation and evaluation report (LEO, R, PU, M36)

@sparta_eu | sparta.eu
23



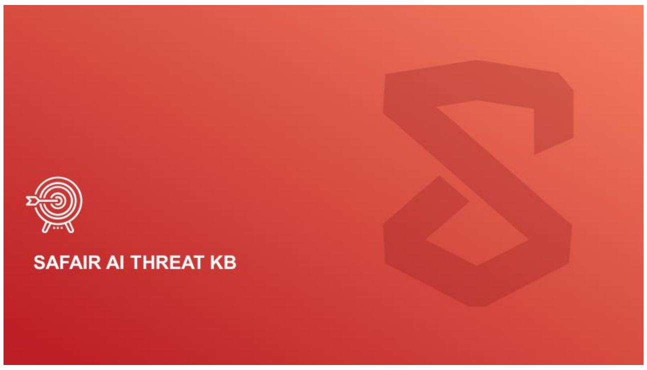
SAFAIR PROGRESS

Dataset selection

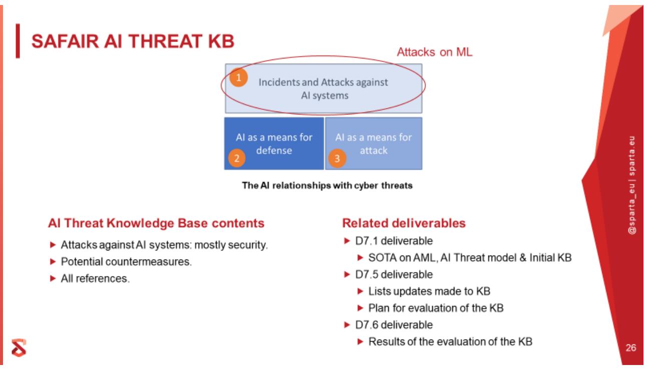
- High impact vs repeatability
- AI, ML and DL
- Public dataset gathering.
  - Image classification
  - Temporal series prediction
  - Malware classification
  - Fake news detection

AI threat analysis mechanisms and tools

- AI models taxonomy state of the art.
- AI systems threats state of the art.
- AI risk management
- Some proofs of concept: AI system and attacks
- List of potential countermeasures



SAFAIR AI THREAT KB



SAFAIR AI THREAT KB

Attacks on ML
1 Incidents and Attacks against AI systems
AI as a means for defense 2
AI as a means for attack 3

The AI relationships with cyber threats

AI Threat Knowledge Base contents

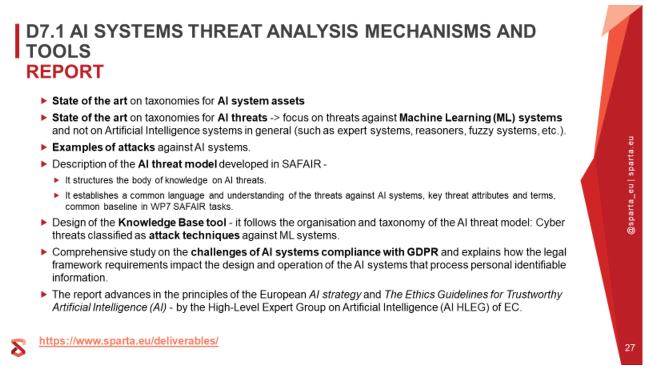
- Attacks against AI systems: mostly security.
- Potential countermeasures.
- All references.

Related deliverables

- D7.1 deliverable
  - SOTA on AML, AI Threat model & Initial KB
- D7.5 deliverable
  - Lists updates made to KB
  - Plan for evaluation of the KB
- D7.6 deliverable
  - Results of the evaluation of the KB

@sparta_eu | sparta.eu

26



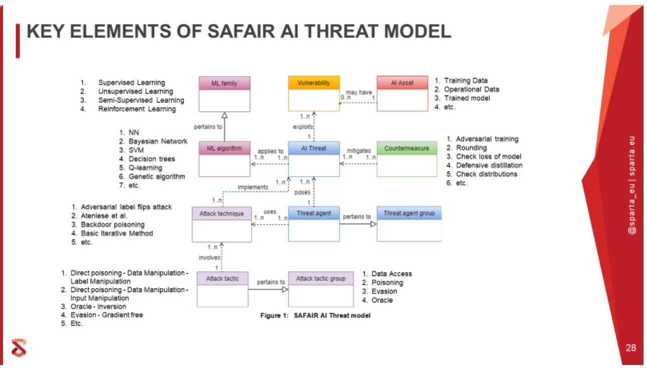
D7.1 AI SYSTEMS THREAT ANALYSIS MECHANISMS AND TOOLS REPORT

- **State of the art** on taxonomies for **AI system assets**
- **State of the art** on taxonomies for **AI threats** -> focus on threats against **Machine Learning (ML) systems** and not on Artificial Intelligence systems in general (such as expert systems, reasoners, fuzzy systems, etc.).
- **Examples of attacks** against AI systems.
- Description of the **AI threat model** developed in SAFAIR -
  - It structures the body of knowledge on AI threats.
  - It establishes a common language and understanding of the threats against AI systems, key threat attributes and terms, common baseline in WP7 SAFAIR tasks.
- Design of the **Knowledge Base tool** - it follows the organisation and taxonomy of the AI threat model: Cyber threats classified as **attack techniques** against ML systems.
- Comprehensive study on the **challenges of AI systems compliance with GDPR** and explains how the legal framework requirements impact the design and operation of the AI systems that process personal identifiable information.
- The report advances in the principles of the European *AI strategy* and *The Ethics Guidelines for Trustworthy Artificial Intelligence (AI)* - by the High-Level Expert Group on Artificial Intelligence (AI HLEG) of EC.

https://www.sparta.eu/deliverables/

@sparta_eu | sparta.eu

27

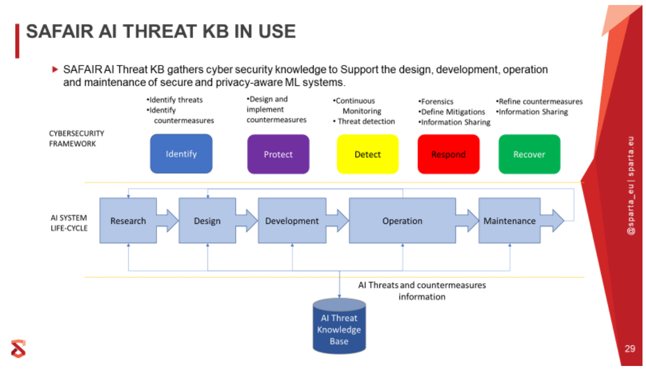


KEY ELEMENTS OF SAFAIR AI THREAT MODEL

Figure 1: SAFAIR AI Threat model

@sparta_eu | sparta.eu

28

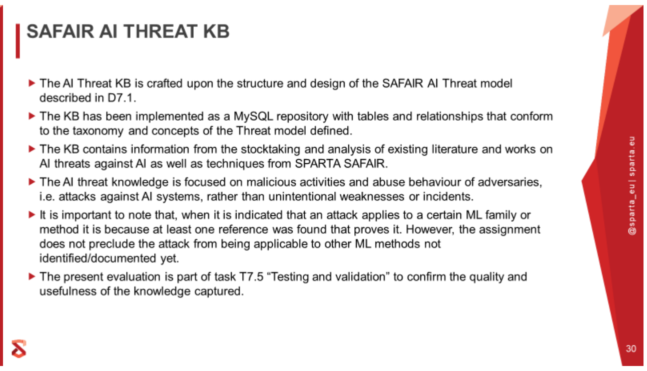


SAFAIR AI THREAT KB IN USE

- SAFAIR AI Threat KB gathers cyber security knowledge to Support the design, development, operation and maintenance of secure and privacy-aware ML systems.

@sparta_eu | sparta.eu

29

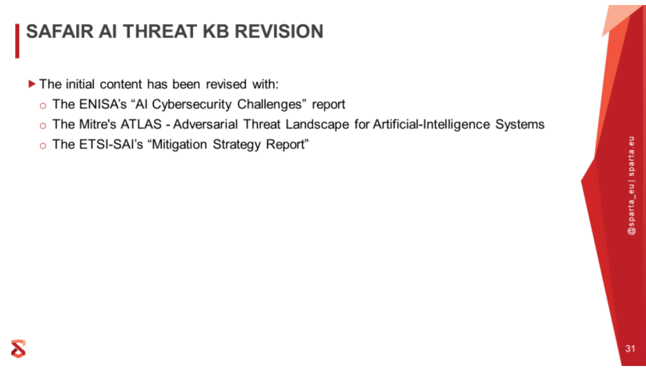


SAFAIR AI THREAT KB

- The AI Threat KB is crafted upon the structure and design of the SAFAIR AI Threat model described in D7.1.
- The KB has been implemented as a MySQL repository with tables and relationships that conform to the taxonomy and concepts of the Threat model defined.
- The KB contains information from the stocktaking and analysis of existing literature and works on AI threats against AI as well as techniques from SPARTA SAFAIR.
- The AI threat knowledge is focused on malicious activities and abuse behaviour of adversaries, i.e. attacks against AI systems, rather than unintentional weaknesses or incidents.
- It is important to note that, when it is indicated that an attack applies to a certain ML family or method it is because at least one reference was found that proves it. However, the assignment does not preclude the attack from being applicable to other ML methods not identified/documented yet.
- The present evaluation is part of task T7.5 "Testing and validation" to confirm the quality and usefulness of the knowledge captured.

@sparta_eu | sparta.eu

30

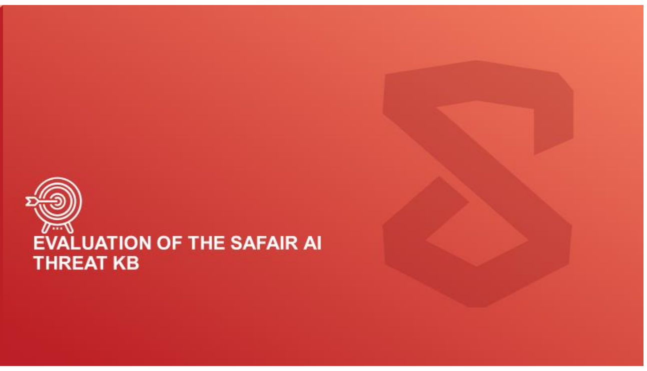


SAFAIR AI THREAT KB REVISION

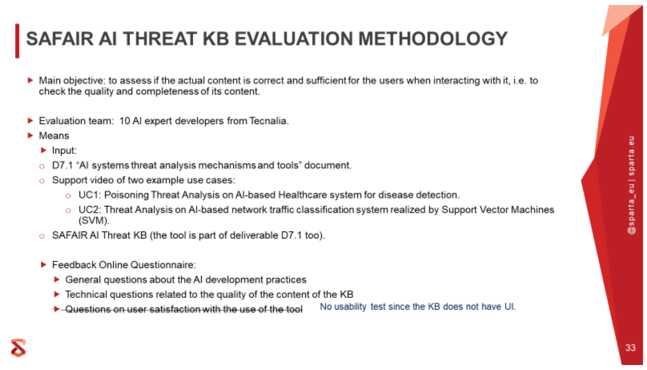
- The initial content has been revised with:
  - The ENISA's "AI Cybersecurity Challenges" report
  - The Mitre's ATLAS - Adversarial Threat Landscape for Artificial-Intelligence Systems
  - The ETSI-SAI's "Mitigation Strategy Report"

@sparta_eu | sparta.eu

31



EVALUATION OF THE SAFAIR AI THREAT KB

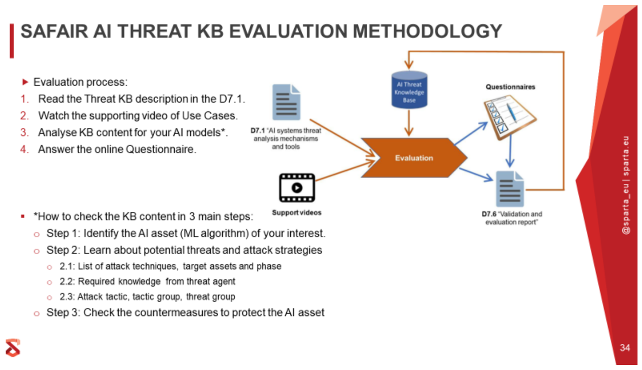


SAFAIR AI THREAT KB EVALUATION METHODOLOGY

- Main objective: to assess if the actual content is correct and sufficient for the users when interacting with it, i.e. to check the quality and completeness of its content.
- Evaluation team: 10 AI expert developers from Tecnalia.
- Means
  - Input:
  - D7.1 "AI systems threat analysis mechanisms and tools" document.
  - Support video of two example use cases:
    - UC1: Poisoning Threat Analysis on AI-based Healthcare system for disease detection.
    - UC2: Threat Analysis on AI-based network traffic classification system realized by Support Vector Machines (SVM).
  - SAFAIR AI Threat KB (the tool is part of deliverable D7.1 too).
  - Feedback Online Questionnaire:
    - General questions about the AI development practices
    - Technical questions related to the quality of the content of the KB
    - ~~Questions on user satisfaction with the use of the tool~~ No usability test since the KB does not have UI.

@sparta_eu | sparta.eu

33



SAFAIR AI THREAT KB EVALUATION METHODOLOGY

- Evaluation process:
1. Read the Threat KB description in the D7.1.
2. Watch the supporting video of Use Cases.
3. Analyse KB content for your AI models*.
4. Answer the online Questionnaire.

AI Threat Knowledge Base
Questionnaires
D7.1 "AI systems threat analysis mechanisms and tools
Evaluation
Support videos
D7.6 "Validation and evaluation report"

- *How to check the KB content in 3 main steps:
  - Step 1: Identify the AI asset (ML algorithm) of your interest.
  - Step 2: Learn about potential threats and attack strategies
    - 2.1: List of attack techniques, target assets and phase
    - 2.2: Required knowledge from threat agent
    - 2.3: Attack tactic, tactic group, threat group
  - Step 3: Check the countermeasures to protect the AI asset

@sparta_eu | sparta.eu

34

SPARTA

THANK YOU!

@sparta_eu | sparta.eu
24th November 2021

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 830892