SPARTA

# D7.5

# Final version of AI systems security mechanisms and tools

| | |
|---|---|
| **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 / D7.5 / V1.0 |
| **Work package contributing to the deliverable** | WP7 |
| **Due date** | July 2021 – M30 |
| **Actual submission date** | 13th August, 2021 |

| | |
|---|---|
| **Responsible organisation** | TCS |
| **Editor** | Boussad ADDAD |
| **Dissemination level** | PU |
| **Revision** | V1.0 |

| | |
|---|---|
| **Abstract** | This deliverable describes the second and last version of the defensive mechanisms against evasion attacks, explainability enhancing mechanisms, and fairness ensuring mechanisms. |
| **Keywords** | Adversarial Machine Learning, Secure AI, Robustness, Testing, Validation, Explainable AI, Fairness |

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

**Editor**

Boussad Addad (TCS)

**Contributors** (ordered according to beneficiary numbers)

Zakaria Chihani (CEA)

Manon Knockaert, Jean-Marc Van Gyseghem, Sophie Everarts de Velp (UNamur)

Mohammed Reza Norouzian (TUM)

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

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

Boussad Addad, Vincent Thouvenot, Jerome Kodjabachian (TCS)

Michal Choraś, Marek Pawlicki, Mateusz Szczepański, Krzysztof Samp (ITTI)

**Reviewers** (ordered according to beneficiary numbers)

Evaldas Bružė (L3CE)

Adam Kozakiewicz (NASK)

# Executive Summary

The current document D7.5[1] is a direct sequel of deliverable D7.4[2] of SPARTA project. It presents the final version of work done in the SAFAIR program (WP7). This effort seeks to address two major contemporary problems encountered in Artificial Intelligence (AI) systems based on Machine Learning (ML). The first goal is to ensure the security and robustness of AI/ML solutions and the second one deals with the issue of trustworthiness, transparency, and fairness of AI/ML. Finally, document D7.5 presents the results of an AI contest program that aims to develop robustification techniques against a popular type of attacks called evasion attacks.

To the first end, the SAFAIR program produces formalisms, methods and tools, which make current AI algorithms more robust, ensuring their security, reliability and privacy. While the threat analysis conducted by the SAFAIR program made its way to the D7.1[3], the deliverable D7.2 holds the information regarding the defensive and reactive mechanisms designed to ensure resilience against the new, complex cyber-threats identified in D7.1. The preliminary descriptions of those tools and methods are offered in Chapter 2 of D7.2. The mechanisms and tools proposed by the SAFAIR program seek to optimize resiliency without compromising the advantages provided by AI and aiming not to affect the performance of AI.

Secondly, the SAFAIR program seeks to extend the explainability of AI, which usually comes in the form of a black-box. This provides both the verification of functionality of AI and gives users relying on decisions made by AI the ability to trace back those decisions to the inputs, paving the way to better understanding of AI, increasing the transparency of AI and increasing the confidence in AI. In Chapter 3 of D7.2[4] some explainability mechanisms and tools can be found. The third aspect of the above mentioned objectives deals with the legal and societal aspects of trustworthiness and fairness of AI. The third task of SAFAIR seeks to provide mechanisms to reduce conscious and unconscious bias in AI decisions, ensuring that functionality does not introduce discrimination, and finding ways to grant both credibility and correct compliance with relevant legislative regulations. The current state of that work is contained in chapter 4 of D7.2.

D7.3[5] is dedicated to test and evaluate the adversarial machine learning solutions. To this end, D7.3 proposes two approaches: i) design an open AI contest in the adversarial environment to facilitate measurable progress towards robust machine learning models and, more generally, applicable adversarial attacks and ii) implement an open-source python tool that provides standardized reference implementations of adversarial example construction techniques and adversarial training.

D7.4 is the report due for Month 30 that presents the first demonstration of AI systems security mechanisms and tools. Compared to D7.2, which presented the state of art of such mechanisms, the objective of D7.4 is to give a detailed description of the approaches chosen in SAFAIR for security, trustworthiness and fairness of AI systems, which will be developed in the final demonstrator and applied in use cases to illustrate them.

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

D7.5 report, due for Month 30 (July 2021), presents the final version of a demonstrator of defence techniques against model evasion attacks (Chapter 2), AI explainability enhancing (Chapter 3) and fairness ensuring mechanisms (Chapter 4). The progress and results of the AI contest program are provided in Chapter 5. Chapter 6 explains the progress since Month 18 of the initially presented SAFAIR AI Threat model that is being updated to capture new results from ENISA and other relevant initiatives on AI threat landscape as well as from state-of-the-art literature on AML attacks. Then, some legal aspects about AI solutions are summarized in Chapter 7. Finally, Chapter 8 provides a short list of takeaways on security and robustness, and explainability and fairness, and concludes this document.

# Table of Content

# List of Figures

# List of Tables

# Chapter 1 Introduction

The document is the deliverable D7.5 of SPARTA and presents the work of the SAFAIR program (SPARTA WP7). It addresses two major contemporary problems inherent to the wide spread of Artificial Intelligence (AI) based on Machine Learning (ML) models. The first goal is to ensure the security of AI/ML solutions and the second to deal with the issue of trustworthiness and fairness of AI/ML.

To the first end, the SAFAIR program produces formalisms, methods and tools, which make current AI algorithms more robust, ensuring their security, reliability, and privacy. While the threat analysis conducted by the SAFAIR program made its way to the D7.1, the D7.2 work holds the information regarding the defensive and reactive mechanisms designed to ensure resilience against the new, complex cyber-threats identified in D7.1. The preliminary descriptions of those tools and methods were presented in D7.2. The mechanisms and tools proposed by the SAFAIR program seek to optimize resiliency without compromising the advantages provided by AI and aiming not to affect its performance. Secondly, the SAFAIR program seeks to extend the explainability of AI, which usually comes in the form of a blackbox. This provides both the verification of functionality of AI and gives personnel relying on decisions made by AI the ability to trace back those decisions to the inputs, paving the way to better understanding of AI, increasing the transparency of AI and increasing the confidence in it. In Chapter 3 of D7.2, some explainability mechanisms and tools were provided. One of the above mentioned objectives deals with the legal and societal aspects of trustworthiness and fairness of AI. The third task of SAFAIR seeks to provide mechanisms to reduce conscious and unconscious bias in AI decisions, ensuring functionality that does not introduce discrimination, and finding ways to grant both credibility and correct compliance with relevant legislative regulations. D7.4 presented the first demonstration of AI systems security mechanisms and tools. Compared to D7.2, which presented the state of art of such mechanisms, the objective of D7.4 was to give a detailed description of the approaches chosen in SAFAIR for security, trustworthiness and fairness of AI systems.

The current document provides a description of the second and last version of the demonstrators. Chapter 2 is dedicated to demonstration of evasion attacks and some defence mechanisms to counter them. Many approaches are compared while adopting different attack techniques.

In Chapter 3, we present a component based on ShapKit, a Python module dedicated to local explanation of machine learning model presented in D7.4. We apply this component on a case dedicated to Denial of Service attack detection. In the second part of the Chapter, we present a component named hybrid oracle explainer, based on decision trees, which has been applied to Intrusion Detection Systems, and we present some supplemental explorations of surrogate-type methods for explainable artificial intelligence. We apply it in the context of cybersecurity.

In Chapter 4, we describe a tool that is dedicated both to interpretability and to fairness inspection and presents its usage on the adult-income dataset. All the functions we use and the resulting plots in this part are implemented in *ethik*, a Python module dedicated to AI fairness and interpretability.

Chapter 5 is dedicated to a contest organized to evaluate evasion attacks and the defence strategies adopted by the participants with respect to a baseline of standard approaches.

Chapter 6 describes the updates performed on the SAFAIR AI Threat model and Knowledge Base of SPARTA deliverable D7.1 where the initially presented approach has been extended and improved to capture in the model new results from ENISA and other relevant initiatives on AI threat landscape.

Chapter 7 is dedicated to legal aspects. It aims firstly at establishing a practical checklist for AI software developers in order to respect the equity criteria throughout the development process. The

second part of it will link the different elements of the fairness principle with the algorithms proposed by the partners in the current deliverable.

The last chapter derives a conclusion about the contents of the deliverable.

# Chapter 2 Defensive mechanisms against model evasion attacks

Machine learning models proved to be successful for many tasks, from object detection on images to natural language question answering task, and intrusion detection on hosts or networks. Unfortunately, these models are also vulnerable and subject to some inherent attacks, either during the training process (data poisoning) or after the algorithm deployment (model evasion).

The current chapter addresses evasion attacks and above all how to counter them using different approaches. To assess the effectiveness of these defence methods, we built a demonstrator on a cybersecurity use case, malware detection in PDF files. A good AI/ML solution must obviously be robust against adversarial examples but also keep unchanged its basic performance (e.g. accuracy).

In D7.4, we have already started the implementation of a demonstrator showing both attacks and defence techniques on some use cases like health image classification, PDF malware detection, and network intrusion detection. In the current report, we provide the results of the final demonstrator in Section 2.8, while we extend the list of defensive mechanisms and consider another use case in Section 2.7 (face identification).

## 2.1 Implemented attacks

Four model evasion attacks are implemented in the demonstrator:

- FGSM (Fast gradient sign method)
- iter-FGSM (iterative Fast gradient sign method)
- C&W (Carlini and Wagner)
- CIA (Centered Initial Attack)

For more details about these techniques, please see deliverable D7.4 “First demonstration of AI systems security mechanisms and tools”.

## 2.2 Implemented defences

Two approaches are implemented in the demonstrator: adversarial training and feature scattering. The first one was already available in the first version of the demonstrator (deliverable D7.4) but not the second one. They are compared in the current document.

## 2.3 Adversarial training

The primary objective of the adversarial training is to increase model robustness by injecting adversarial examples into the training dataset [61], [57]. Adversarial training [60] is a standard brute force approach where the defender simply generates a lot of adversarial examples, using one or more attack strategies, and uses them for retraining the target model (whitebox attack) or the proxy model (blackbox or no-box). The augmentation can be made either by feeding the model with both the original data and the crafted data, or by replacing the original data (or a portion of it) with their adversarial counterpart.

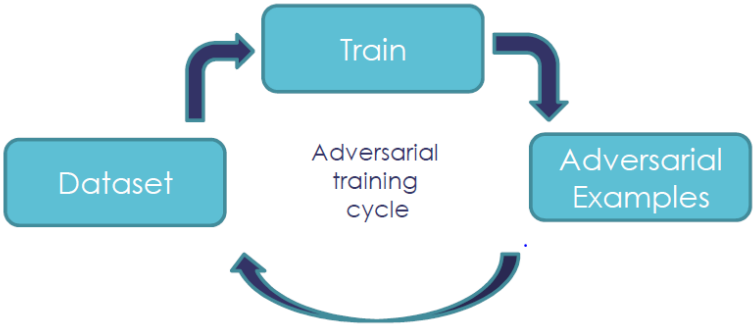



Figure 1: Iterative adversarial training principle

Adversarial training can be applied iteratively but generating new adversarial examples using the already adversarially robustified model. So, the number of iterations (cycles in figure above) is a hyper parameter that can be adjusted to get a better robustness.

## 2.4 Feature scattering

The work published in [63] is an interesting approach for neural networks robustification. We provide here an overview of the principle behind it.

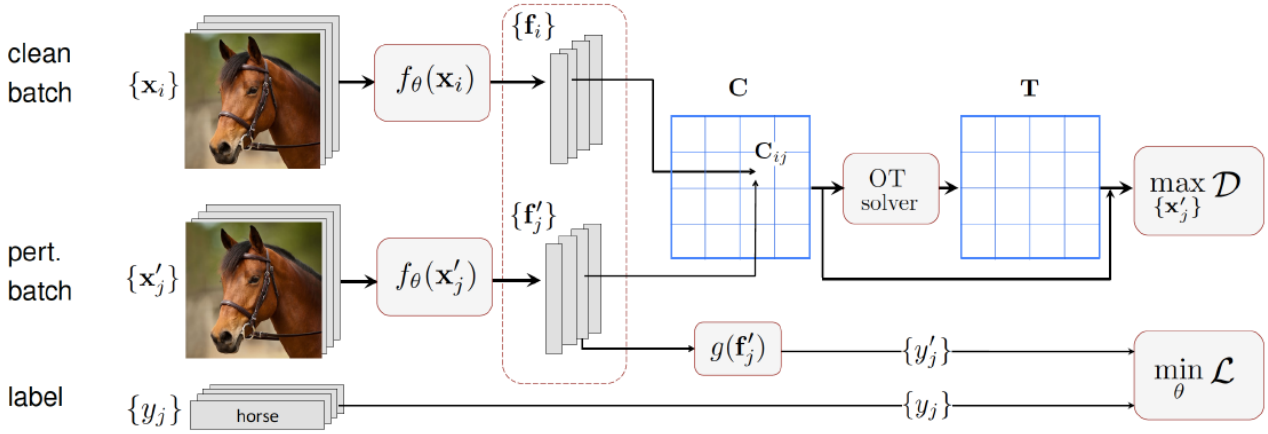



Figure 2: Feature Scattering-based Adversarial Training Pipeline

The technique of feature scattering is quite similar to the procedure of crafting adversarial examples in adversarial training. It aims to maximize the distance between the outputs relative to clean and perturbed images while respecting a norm constraint during the training of the neural network, while obviously keeping the good labels on the perturbed data.

The main difference is that the distance *D* is not measured between the outputs of two single images, a clean and a perturbed one, but the distributions of the ***sets*** of the extracted features (encodings obtained using a deep neural network up to the softmax layer) of the clean and perturbed sets of images. For this purpose, the optimal transport (OT) distance is considered. The resolution of the OT optimization problem is carried out with solvers to make it tractable (iteratively for a length *T*). The transport cost is defined as the cosine distance between the image features.

**Algorithm 1** Feature Scattering-based Adversarial Training

**Input:** dataset $S$, training epochs $K$, batch size $n$, learning rate $\gamma$, budget $\epsilon$, attack iterations $T$
**for** $k = 1$ **to** $K$ **do**
  **for** random batch $\{\mathbf{x}_i, y_i\}_{i=1}^{n} \sim S$ **do**
    **initialization**: $\boldsymbol{\mu} = \sum_i u_i \delta_{\mathbf{x}_i}, \quad \boldsymbol{\nu} = \sum_i v_i \delta_{\mathbf{x}'_i}, \ \mathbf{x}'_i \sim B(\mathbf{x}_i, \epsilon)$
    **feature scattering** (maximizing feature matching distance $\mathcal{D}$ *w.r.t.* $\boldsymbol{\nu}$):
    **for** $t = 1$ **to** $T$ **do**
      $\cdot\ \mathbf{x}'_i \leftarrow \mathcal{P}_{\mathcal{S}_\mathbf{x}}\left(\mathbf{x}'_i + \epsilon \cdot \mathrm{sign}\left(\nabla_{\mathbf{x}'_i} \mathcal{D}(\boldsymbol{\mu}, \boldsymbol{\nu})\right)\right) \quad \forall i = 1, \cdots, n, \quad \boldsymbol{\nu} = \sum_i v_i \delta_{\mathbf{x}'_i}$
    **end for**
    **adversarial training** (updating model parameters):
    $\cdot\ \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \gamma \cdot \frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{x}'_i, y_i; \boldsymbol{\theta})$
  **end for**
**end for**
**Output:** model parameter $\boldsymbol{\theta}$.

Figure 3: Feature scattering algorithm pseudo code.

The proposed approach is equivalent to the minimization of a loss consisting of the conventional loss on the original data (features + labels), and a regularization term coupled over ***all*** the inputs.

Feature scattering has the advantage of solving the problem of label leaking. Label leaking occurs when the additive perturbation (difference between clean and perturbed data) is highly correlated with the ground-truth label. Therefore, when it is added to the image, the network can directly tell the class label by decoding the additive perturbation without relying on the real content of the image, leading to higher adversarial accuracy than the clean image during training. In feature scattering, this problem is not faced since the perturbation is calculated using the batches of the images samples all at once.

In our demonstrator, we suppose that we do not have access to the AI/ML solution we want to attack (no-box type). We therefore train a proxy neural network as usual and use it to craft adversarial examples to be tested on the defence robustified using feature scattering.

## 2.5 Hybrid approach

We also investigated the effect of feature scattering on the robustification of another model through adversarial examples generated through a proxy trained using feature scattering. In other words, we train a neural network proxy model M1 with feature scattering, then use it to generate many adversarial examples. These examples are injected in the training dataset (with or without replacement of the clean data) of another model, a random forest for instance.

## 2.6 Neuron's Behaviour

This defence is focused on the deep learning models. The objective is to analyse the behaviour of each neuron that compounds the Area of Interest of the model.

First, the behaviour of a neuron is defined. An example of behaviour could be the impact of each neuron in the output prediction. In this case, the behaviour of each neuron changes for each input sample. Figure 4 shows each neuron's impact in the prediction for the neurons part of dense layer of the model for a specific sample.

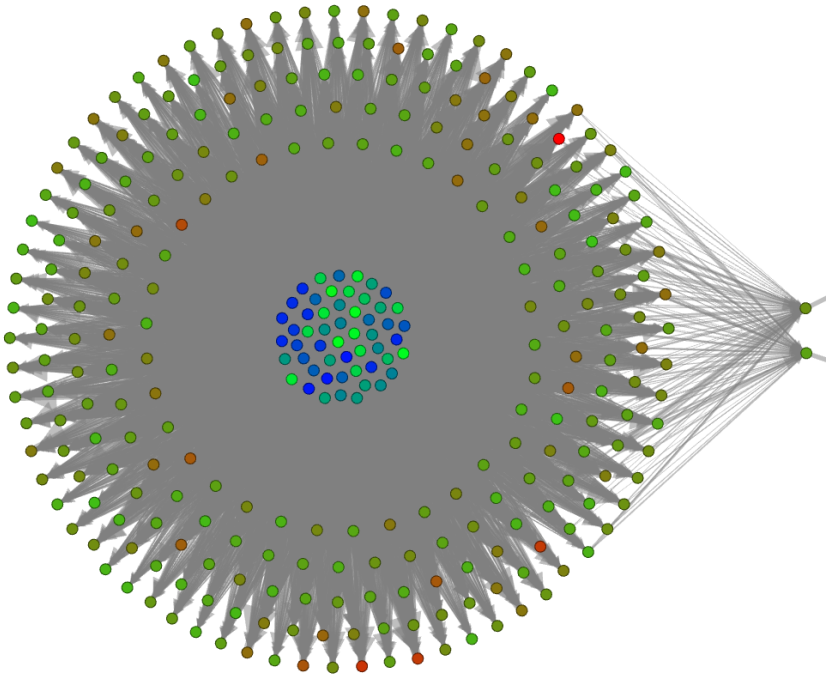


Figure 4: Behaviour map of the desired section of a model in the prediction of a sample. The colours indicate if the impact value of the neuron, which represents the behaviour in our study case, is big or small, compared to the rest of neurons. In this case, blue, green, and red represent high, medium, and low values, respectively.

Once the behaviour of interest is defined, these values are computed for each neuron and each sample selected for the study, generating a dataset that summarises the behaviour of the model depending on the input distance. Once the data is normalized, the values are categorized generating labels. In the previous example, the impact of a neuron can be positive, neutral, or negative.

Finally, the neurons' behaviours are grouped according to the sample that has generated them. Hence, each sample is associated with a group of labels that describe the behaviour of the model in the area of interest. These descriptions can be used to construct a detector allowing to distinguish if an input sample is an adversarial example or not.

## 2.7 Application of preprocessing adversarial defences to robustify face reidentification systems

The algorithms and technologies presented in this section are used to form a submission to the Reidentification defences track of the SAFAIR contest. The task was formulated around the face recognition dataset [1] [2]. The dataset, as used in the task, featured 5304 classes, with 85612 samples in the training subset and 28523 samples in the testing set. The objective of the defensive track was to propose ways of preventing adversarial samples from lowering the accuracy of the face recognition model. The following sections describe the specific technologies used for defining the submission of the contest, the rationale behind those choices, the formulated defences, and provide the results of the experiments.

### 2.7.1 *Face Reidentification (reID)*

In general, reidentification refers to the process of re-attaching publicly available data to an anonymised record in order to discover the identity of an individual. In the context of computer vision, the phrase refers to the ability of an image recognition system to spot an individual across different cameras, and different angles. The following paragraphs will describe the construction of such a

mechanism, which leverages the state-of-the-art findings in the deep learning domain, and implementations of defences against adversarial attacks geared towards disrupting such a system.

### 2.7.2 *The methods used*

Convolutional Neural Networks are widely used for computer vision (CV) tasks [3], some of the best-performing ImageNet contest architectures were based on the premise of utilising convolutional layers. The network architectures tend to be very deep - Inception features over 6 million trainable parameters [4][5], ResNet18 over 11million [6], AlexNet over 60million [7], VGG16 over 138million [8], etc. Training a top-tier deep neural network is therefore a huge computational endeavour [9]. In order not to repeat this effort for each task, transfer learning can be employed [10]. Transfer learning leverages pre-trained networks, essentially using them as feature extractors with frozen weights, feeding samples to the network and only training added dense layers at the output end of the topology. In this work, the VGG-face network was used [11] with the pre-trained 'resnet-50' [12] architecture. VGGFace is trained on a dataset containing 2.6 million face images of over 2.6k people.

The final layer of the pre-trained network is AveragePooling2D with the shape of (None, 1, 1, 2048). To perform transfer learning, a dense layer of 2048 neurons is added to the model, followed by a dropout layer, and wrapped up by the softmax layer set with the number of neurons equal to the number of classes, that is 5304. The weights between the AveragePooling layer and the Dense layer along with the weights between the Dense layer and the output layer constitute the part of the network that is trained on the CelebA dataset, with the weights of the remainder of the network frozen.

The following code is responsible for compiling the network:

```
num_classes = len(reid_dataset_train['label'].value_counts())


baseModel = VGGFace(model='resnet50', include_top=False, input_tensor=Input(shape=(224, 224, 3)))
    print('initialising VGG-face')
    # freezing VGG-Face layers so only the headModel is trained for transfer learning
    for layer in baseModel.layers:
        layer.trainable = False


    headModel = baseModel.output
    headModel = Flatten(name="flatten")(headModel)
    headModel = Dense(2048, activation="relu")(headModel)
    headModel = Dropout(0.5)(headModel)
    headModel = Dense(num_classes, activation="softmax")(headModel) #output classes
    print('defining headModel for transfer learning')
    from tensorflow.keras import Model
    model = Model(inputs=baseModel.input, outputs=headModel)


    model.compile(
      optimizer='adam',
      loss=tf.losses.SparseCategoricalCrossentropy(from_logits=True),
      metrics=['accuracy'])
    model.summary()
    print('compiling model')
```

```
log_dir = "logs/fit/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
checkpoint_filepath = '/tmp/checkpoint'
model_checkpoint_callback = tf.keras.callbacks.ModelCheckpoint(
    filepath=checkpoint_filepath,
    save_weights_only=True,
    monitor='val_accuracy',
    mode='max',
    save_best_only=True)
callback = tf.keras.callbacks.EarlyStopping(monitor='loss', patience=3)

history = model.fit(train_generator, epochs=20,
                    validation_data=validation_generator, callbacks=[callback, model_checkpoint_callback])
```

The trainable part of the model contains 15,064,248 parameters. To allow fast prototyping, a toy model was built on fourteen most populated classes in the CelebA dataset. Changing just the number of classes allowed to reduce the number of trainable parameters to just over 4 million; a reduction of over 70%.

Multi-Task Cascaded Convolutional Neural Networks (MTCNN) is capable of spotting faces and extracting them for later processing by other networks. A state-of-the-art face recognition processing pipeline consists of MTCNN for face detection and landmark placement, and a CNN used for placing the extracted face in adequate categories [13][14][15]. In this work, MTCNN is used for preprocessing the CelebA images for both training and testing. The CelebA subset selected for the formulation of the model was further split into the training set and the testing set.

```
from mtcnn.mtcnn import MTCNN
im = cv2.cvtColor(reshaped, cv2.COLOR_BGR2RGB) detector = MTCNN()
results = detector.detect_faces(im)
x1, y1, width, height = results[0]['box']
x2, y2 = x1 + width, y1 + height
face = im[y1:y2, x1:x2]
```

The classifier performance on the test set containing the 14 most populated classes are found in Table 1.

Table 1: Classifier performance on the test set containing the 14 most populated classes.

| Label | Precision | Recall | f1-score | Support |
|---|---|---|---|---|
| **1757.0** | 1.00 | 1.00 | 1.00 | 7 |
| **2114.0** | 1.00 | 1.00 | 1.00 | 7 |
| **2820.0** | 0.88 | 1.00 | 0.93 | 7 |
| **3227.0** | 1.00 | 0.86 | 0.92 | 7 |

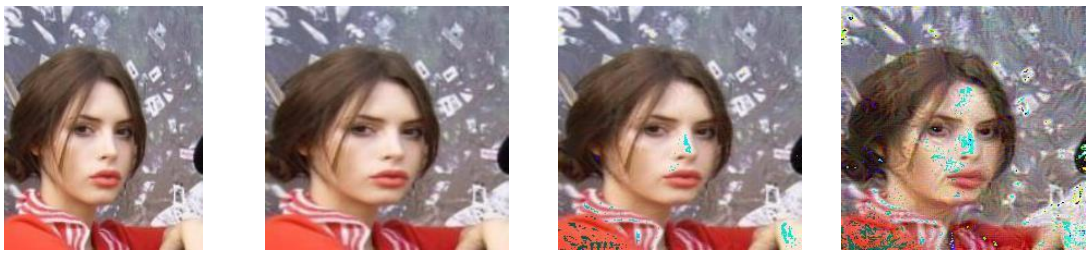
| Label | Precision | Recall | f1-score | Support |
|---|---|---|---|---|
| **3699.0** | 0.88 | 1.00 | 0.93 | 7 |
| **3745.0** | 1.00 | 1.00 | 1.00 | 7 |
| **3782.0** | 1.00 | 1.00 | 1.00 | 7 |
| **4262.0** | 0.88 | 1.00 | 0.93 | 7 |
| **4740.0** | 1.00 | 1.00 | 1.00 | 7 |
| **4978.0** | 1.00 | 1.00 | 1.00 | 7 |
| **6568.0** | 1.00 | 1.00 | 1.00 | 7 |
| **8968.0** | 1.00 | 1.00 | 1.00 | 7 |
| **9152.0** | 1.00 | 1.00 | 1.00 | 7 |
| **9256.0** | 1.00 | 0.71 | 0.83 | 7 |
| **macro avg** | 0.97 | 0.97 | 0.97 | 98 |
| **weighted avg** | 0.97 | 0.97 | 0.97 | 98 |
| **accuracy** | 0.9693877551020408 | | | |
| **balanced accuracy** | 0.9693877551020408 | | | |

For better evaluation of the effects of adversarial perturbations and adversarial defences, the misclassified samples were removed from the set, manually pushing the performance to 100% accuracy. That way, any adversarial perturbations are registered as drops in performance, avoiding a situation where an attack pushes the misclassified sample to the correct class. Furthermore, the way the defences affect the classifier performance is more distinct.

### 2.7.3 *Adversarial attacks*

The testing set was then subjected to the procedure of creating the adversarial samples.

To produce the adversarial attacks, the Projected Gradient Descent method was used, considering PGD as the universal first-order adversary, following [16]. The maximum number of iterations was set to 100, the epsilon step to 0.1. Multiple values of *epsilon* were tested to simulate different strengths of attack. The effect different strengths of the attacks have on the image can be seen in Figure 5. The three attacked pictures are reformatted to fit the vgg-face input shape.



Clear test sample | PGD epsilon = 8/255 | PGD epsilon = 4. | PGD epsilon = 25.5.

Figure 5: The effects of different strengths of the attacks on the image.

The effects of PGD eps=4 on the performance of the classifier can be seen in Table 2.

Table 2: The effects of PGD eps=4 on the performance of the classifier.

| Label | Precision | Recall | f1-score | Support |
|---|---|---|---|---|
| 1757.0 | 1.00 | 0.14 | 0.25 | 7 |
| 2114.0 | 0.33 | 0.14 | 0.20 | 7 |
| 2820.0 | 0.00 | 0.00 | 0.00 | 7 |
| 3227.0 | 1.00 | 0.17 | 0.29 | 6 |
| 3699.0 | 0.32 | 1.00 | 0.48 | 7 |
| 3745.0 | 0.00 | 0.00 | 0.00 | 7 |
| 3782.0 | 0.00 | 0.00 | 0.00 | 7 |
| 4262.0 | 0.33 | 0.71 | 0.45 | 7 |
| 4740.0 | 0.08 | 0.14 | 0.11 | 7 |
| 4978.0 | 0.00 | 0.00 | 0.00 | 7 |
| 6568.0 | 1.00 | 0.14 | 0.25 | 7 |
| 8968.0 | 0.00 | 0.00 | 0.00 | 7 |
| 9152.0 | 0.50 | 0.14 | 0.22 | 7 |
| 9256.0 | 1.00 | 0.40 | 0.57 | 5 |
| **macro avg** | 0.40 | 0.21 | 0.20 | 95 |
| **weighted avg** | 0.38 | 0.21 | 0.19 | 95 |
| **accuracy** | 0.21052631578947367 | | | |
| **balanced accuracy** | 0.2139455782312925 | | | |

### 2.7.4 *Defences*

There have been a number of defences proposed by the research community [17].The task is to design robust AI tools that are resilient to adversarial attacks. Some methods rely on retraining the entire classifier using attacks generated with the known attack methods [18]. This method, called adversarial training, not only impacts the effectiveness of the classifier, but also requires an immense computational effort.

The proposition contained in this section utilises the idea of using pre-processing methods to robustify existing AI-based classifiers, so as the users do not need to re-train their models. The proposed methods are accompanied by an assessment of how the defensive measures affect the classifier performance, which helps optimise the resiliency of AI against the loss of performance some defences introduce.

### 2.7.5 *JPEG Compression*

The JPEG compression used as adversarial defence relies on the fact that JPEG-compressed images are very prevalent in contemporary usage. Following the authors of [19], who noted that JPEG compression often has the ability to reverse the effects of small adversarial perturbations, the technique is evaluated here for the use as a purely pre-processing defence against adversarial attacks. The compression has the effect of removing additive artefacts in square blocks of an image, effectively working as a filter removing adversarial perturbations [20].

The effect of different magnitudes of compression (20, 40, 80) can be seen in Figure 6.

Figure 6: JPEG compression.

The results of the classifier using JPEG compression with quality set to 20 on PGD attacks with epsilon=4 can be found in Table 3.

Table 3: The results of the classifier using JPEG compression with quality set to 20 on PGD attacks with epsilon=4.

| Label | Precision | Recall | f1-score | Support |
|---|---|---|---|---|
| 1757.0 | 1.00 | 1.00 | 1.00 | 7 |
| 2114.0 | 1.00 | 1.00 | 1.00 | 7 |
| 2820.0 | 1.00 | 1.00 | 1.00 | 7 |
| 3227.0 | 1.00 | 0.83 | 0.91 | 6 |

| Label | Precision | Recall | f1-score | Support |
|---|---|---|---|---|
| 3699.0 | 0.88 | 1.00 | 0.93 | 7 |
| 3745.0 | 0.86 | 0.86 | 0.86 | 7 |
| 3782.0 | 0.86 | 0.86 | 0.86 | 7 |
| 4262.0 | 0.78 | 1.00 | 0.88 | 7 |
| 4740.0 | 1.00 | 1.00 | 1.00 | 7 |
| 4978.0 | 0.86 | 0.86 | 0.86 | 7 |
| 6568.0 | 1.00 | 1.00 | 1.00 | 7 |
| 8968.0 | 1.00 | 0.86 | 0.92 | 7 |
| 9152.0 | 1.00 | 0.86 | 0.92 | 7 |
| 9256.0 | 0.80 | 0.80 | 0.80 | 5 |
| **macro avg** | 0.93 | 0.92 | 0.92 | 95 |
| **weighted avg** | 0.93 | 0.93 | 0.93 | 95 |
| **accuracy** | 0.9263157894736842 | | | |
| **balanced accuracy** | 0.9227891156462587 | | | |

### 2.7.6 *Gaussian Data Augmentation*

Gaussian Data Augmentation [21] is a process of adding gaussian noise to a sample. This method is proven not to produce adversarial samples and can reverse the effects of known adversarial attacks. Image samples with different sigma settings can be seen in Figure 7.

Figure 7: Gaussian Augmentation - sigma 255.0/5, 255.0/17, 255.0/3.

### 2.7.7 *Local Spatial Smoothing*

Following the research of [22], spatial smoothing can be used to reduce the effects of added adversarial noise. The algorithm uses local blurring filters to remove the effects of adversarial noise. The approach is one of the feature squeezing methods and can be effectively applied as a pre-processor-based defence.

The image before and after Spatial Smoothing can be seen in Figure 8.

Figure 8: The image before and after spatial smoothing.

### 2.7.8 *Total variance minimisation*

Total variance minimisation is a model-agnostic preprocessor approach. In the original paper [23] the defence is used for retraining the model and then the inputs are also pre-processed at test time. The method reassembles the image by rebuilding a randomly chosen set of pixels with the plainest depiction of these pixels.

The Image before and after total variance minimisation can be seen in Figure 9.

Figure 9: The Image before and after total variance minimisation.

### 2.7.9 *Block-Matching Convolutional Neural Network (BMCNN) for Image Denoising as an adversarial defence*

Following the work in image denoising presented in [24], and extending the idea of applying autoencoders as adversarial defences [25] the BMCNN is proposed for the a method of robustifying the image recognition system against adversarial attacks. BMCNN is an attempt to merge two leading approaches to image denoising: nonlocal self-similarity prior based methods [26] and feed-forward denoising with the use of Convolutional Neural Networks [27]. The method is applied as a pre-processor to remove adversarial noise before the sample is fed to the classifier.

The results of the BMCNN with sigma set to 20 used on adversarial samples created with PGD using with epsilon set to four can be seen in Table 4.

Table 4: 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 | Support |
|---|---|---|---|---|
| 1757.0 | 1.00 | 1.00 | 1.00 | 7 |
| 2114.0 | 1.00 | 1.00 | 1.00 | 7 |
| 2820.0 | 1.00 | 1.00 | 1.00 | 7 |
| 3227.0 | 0.83 | 0.83 | 0.83 | 6 |
| 3699.0 | 0.70 | 1.00 | 0.82 | 7 |
| 3745.0 | 1.00 | 0.71 | 0.83 | 7 |
| 3782.0 | 0.88 | 1.00 | 0.93 | 7 |
| 4262.0 | 0.78 | 1.00 | 0.88 | 7 |
| 4740.0 | 1.00 | 1.00 | 1.00 | 7 |
| 4978.0 | 0.88 | 1.00 | 0.93 | 7 |
| 6568.0 | 1.00 | 0.86 | 0.92 | 7 |
| 8968.0 | 1.00 | 0.86 | 0.92 | 7 |
| 9152.0 | 0.80 | 0.57 | 0.67 | 7 |
| 9256.0 | 1.00 | 0.8 | 0.89 | 5 |
| **macro avg** | 0.92 | 0.90 | 0.90 | 95 |
| **weighted avg** | 0.92 | 0.91 | 0.90 | 95 |
| **accuracy** | 0.9052631578947369 | | | |
| **balanced accuracy** | 0.9023809523809525 | | | |

### 2.7.10 *Combining the preprocessing methods*

The low computational cost of the preprocessors in comparison with re-training the classifier allows to mix and match the defences. The experiments show that some pipelines are more effective than others. An example of a defensive pipeline which utilises all the researched defences is displayed in Figure 10. The pipeline makes intuitive sense, as blurring the image should remove some of the artefacts added by PGD, same for JPEG compression, then adding gaussian noise and removing it with BMCNN denoising has the potential of removing both the gaussian and the adversarial noise at the same time. The results of this particular pipeline are shown in Table 5.


Figure 10: A defensive pipeline, which utilises all the researched defences.

Table 5: 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 | Support |
|---|---|---|---|---|
| 1757.0 | 0.50 | 0.71 | 0.59 | 7 |
| 2114.0 | 0.50 | 0.43 | 0.46 | 7 |
| 2820.0 | 0.00 | 0.00 | 0.00 | 7 |
| 3227.0 | 0.40 | 0.33 | 0.36 | 6 |
| 3699.0 | 0.37 | 1.00 | 0.54 | 7 |
| 3745.0 | 0.25 | 0.14 | 0.18 | 7 |
| 3782.0 | 0.25 | 0.86 | 0.39 | 7 |
| 4262.0 | 0.25 | 0.14 | 0.18 | 7 |
| 4740.0 | 0.50 | 0.57 | 0.53 | 7 |
| 4978.0 | 0.67 | 0.29 | 0.40 | 7 |
| 6568.0 | 1.00 | 0.14 | 0.25 | 7 |
| 8968.0 | 0.50 | 0.14 | 0.22 | 7 |
| 9152.0 | 0.67 | 0.29 | 0.40 | 7 |
| 9256.0 | 0.00 | 0.00 | 0.00 | 5 |
| **macro avg** | 0.42 | 0.36 | 0.32 | 95 |
| **weighted avg** | 0.43 | 0.37 | 0.33 | 95 |
| **accuracy** | 0.3684210526315789 | | | |
| **balanced accuracy** | 0.36054421768707484 | | | |

As showcased by the results of the experiment in Table 5, the mix of defences improved the detection metrics as compared to the undefended model, however it did not perform as well as, for example BMCNN denoising alone (Table 4). For the next experiment, the total variance minimisation preprocessor was removed, as it has a similar filtering effect as localised spatial smoothing. The pipeline is shown in Figure 11. The results of the experiment are contained in Table 6.


Figure 11: A defensive, which utilises all the researched defences, except total variance minimisation.

Table 6: 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 | Support |
|---|---|---|---|---|
| 1757.0 | 1.00 | 1.00 | 1.00 | 7 |
| 2114.0 | 1.00 | 1.00 | 1.00 | 7 |
| 2820.0 | 1.00 | 1.00 | 1.00 | 7 |
| 3227.0 | 0.83 | 0.83 | 0.83 | 6 |
| 3699.0 | 0.78 | 1.00 | 0.88 | 7 |
| 3745.0 | 1.00 | 0.86 | 0.92 | 7 |
| 3782.0 | 0.75 | 0.86 | 0.80 | 7 |
| 4262.0 | 0.78 | 1.00 | 0.88 | 7 |
| 4740.0 | 1.00 | 1.00 | 1.00 | 7 |
| 4978.0 | 0.86 | 0.86 | 0.86 | 7 |
| 6568.0 | 1.00 | 0.86 | 0.92 | 7 |
| 8968.0 | 1.00 | 0.86 | 0.92 | 7 |
| 9152.0 | 1.00 | 0.57 | 0.73 | 7 |
| 9256.0 | 0.83 | 1.00 | 0.91 | 5 |
| **macro avg** | 0.92 | 0.91 | 0.90 | 95 |
| **weighted avg** | 0.92 | 0.91 | 0.90 | 95 |
| **accuracy** | 0.9052631578947369 | | | |
| **balanced accuracy** | 0.9064625850340137 | | | |

To find the optimal mix of preprocessors that would minimise or eliminate the effect of adversarial perturbations without significantly deteriorating the classifier results a range of experiments was performed. The results of some of those tests are contained in Table 7 and Table 8.

Table 7: The results of the classifier using spatial smoothing with JPEG compression on PGD images with epsilon set to four.

| Label | Precision | Recall | f1-score | Support |
|---|---|---|---|---|
| 1757.0 | 1.00 | 1.00 | 1.00 | 7 |
| 2114.0 | 1.00 | 1.00 | 1.00 | 7 |
| 2820.0 | 1.00 | 1.00 | 1.00 | 7 |
| 3227.0 | 1.00 | 0.83 | 0.91 | 6 |
| 3699.0 | 0.78 | 1.00 | 0.88 | 7 |
| 3745.0 | 0.86 | 0.86 | 0.86 | 7 |
| 3782.0 | 0.86 | 0.86 | 0.86 | 7 |
| 4262.0 | 0.78 | 1.00 | 0.88 | 7 |
| 4740.0 | 1.00 | 1.00 | 1.00 | 7 |
| 4978.0 | 0.86 | 0.86 | 0.86 | 7 |
| 6568.0 | 1.00 | 1.00 | 1.00 | 7 |
| 8968.0 | 1.00 | 0.86 | 0.92 | 7 |
| 9152.0 | 1.00 | 0.71 | 0.83 | 7 |
| 9256.0 | 0.80 | 0.80 | 0.80 | 5 |
| **macro avg** | 0.92 | 0.91 | 0.91 | 95 |
| **weighted avg** | 0.93 | 0.92 | 0.92 | 95 |
| **accuracy** | 0.9157894736842105 | | | |
| **balanced accuracy** | 0.9125850340136055 | | | |


Figure 12: A defensive pipeline with JPEG compression, gaussian augmentation and BMCNN.

Table 8: 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 | Support |
|---|---|---|---|---|
| 1757.0 | 0.88 | 1.00 | 0.93 | 7 |
| 2114.0 | 1.00 | 1.00 | 1.00 | 7 |
| 2820.0 | 1.00 | 1.00 | 1.00 | 7 |
| 3227.0 | 1.00 | 0.83 | 0.91 | 6 |
| 3699.0 | 0.78 | 1.00 | 0.88 | 7 |
| 3745.0 | 0.86 | 0.86 | 0.86 | 7 |
| 3782.0 | 0.86 | 0.86 | 0.86 | 7 |
| 4262.0 | 0.88 | 1.00 | 0.93 | 7 |
| 4740.0 | 1.00 | 1.00 | 1.00 | 7 |
| 4978.0 | 0.86 | 0.86 | 0.86 | 7 |
| 6568.0 | 1.00 | 1.00 | 1.00 | 7 |
| 8968.0 | 1.00 | 0.86 | 0.92 | 7 |
| 9152.0 | 1.00 | 0.71 | 0.83 | 7 |
| 9256.0 | 1.00 | 1.00 | 1.00 | 5 |
| **macro avg** | 0.94 | 0.93 | 0.93 | 95 |
| **weighted avg** | 0.93 | 0.93 | 0.93 | 95 |
| **accuracy** | 0.9263157894736842 | | | |
| **balanced accuracy** | 0.9268707482993197 | | | |

### 2.7.11 *Conclusion*

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

| Label | Precision | Recall | f1-score | Support |
|---|---|---|---|---|
| 1757.0 | 1.00 | 1.00 | 1.00 | 7 |
| 2114.0 | 1.00 | 1.00 | 1.00 | 7 |

| Label | Precision | Recall | f1-score | Support |
|---|---|---|---|---|
| 2820.0 | 1.00 | 1.00 | 1.00 | 7 |
| 3227.0 | 1.00 | 0.83 | 0.91 | 6 |
| 3699.0 | 0.88 | 1.00 | 0.93 | 7 |
| 3745.0 | 0.83 | 0.71 | 0.77 | 7 |
| 3782.0 | 0.75 | 0.86 | 0.80 | 7 |
| 4262.0 | 0.78 | 1.00 | 0.88 | 7 |
| 4740.0 | 1.00 | 1.00 | 1.00 | 7 |
| 4978.0 | 1.00 | 1.00 | 1.00 | 7 |
| 6568.0 | 1.00 | 1.00 | 1.00 | 7 |
| 8968.0 | 1.00 | 1.00 | 1.00 | 7 |
| 9152.0 | 1.00 | 1.00 | 1.00 | 7 |
| 9256.0 | 1.00 | 0.60 | 0.75 | 5 |
| **macro avg** | 0.95 | 0.93 | 0.93 | 95 |
| **weighted avg** | 0.94 | 0.94 | 0.94 | 95 |
| **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.

## 2.8 Application to PDF malware detection

PDF files are all over the Web and may be an important vector for harming machines. A malware embedded in a PDF may try to exploit a flaw in the reader in order to infect the machine [70], [71]. Therefore, several works investigate solutions among which the use of machine learning to detect malicious files, (e.g. [72], [69], [67], [74]). Most of the proposed ML-based models display high detection rates and low false alarms.

However, it is still possible to fool such algorithms, as always the case with machine learning-based solutions. Indeed, several evasion attacks have been proposed in the literature [65], [66]. In [66] for instance, the authors used gradient descent algorithms to evade successfully SVM and Neural Network detectors.

In the current work, we implemented some defence techniques to counter a bunch of attacks we implemented in a demonstrator.

### 2.8.1 *Data*

A PDF file is a tree like structure composed of objects identified by one or several tags. These tags characterise the file and can be used as features for ML model. There are several tools made to analyze PDF files to extract these features. In this work we used the PDFiD Python script made available by Didier Stevens [73]. Among the thousands possible types of tags, Stevens provides a short list of 21 features that are commonly found in common PDFs and in malicious files. For instance the feature /JS indicates that a PDF file contains JavaScript and /OpenAction indicates that an automatic action is to be performed. It is quite suspicious to find these features in a file, and sometimes, it can be a cue of malicious behavior. PDFiD essentially scans through a PDF file, and counts the number of occurrences of each of these features. It can also be used to count the number of occurrences of every features (not only the 21) that characterize a file. Figure below shows an output example of PDFiD script. For each feature, the corresponding tag is given in the first column, and the number of occurrences in the second one.

```
PDFiD 0.2.1 CLEAN_PDF_9000_files/rr-07-58.pdf
 PDF Header: %PDF-1.4
 obj                    23
 endobj                 23
 stream                  6
 endstream               6
 xref                    2
 trailer                 2
 startxref               2
 /Page                   4
 /Encrypt                0
 /ObjStm                 0
 /JS                     0
 /JavaScript             0
 /AA                     0
 /OpenAction             0
 /AcroForm               0
 /JBIG2Decode            0
 /RichMedia              0
 /Launch                 0
 /EmbeddedFile           0
 /XFA                    0
 /Colors > 2^24          0
```

Figure 13: PDFiD output format.

We trained three types of classifiers (see in the following section) with a dataset of 10,000 clean and 10,000 malicious PDF files from the Contagio database [64].

To be in a realistic scenario, we suppose that the dataset of the attacker is different from the one used by the defender. We also suppose that the adversarial examples were not in the dataset used by the attacker to train the proxy model. Therefore, we split the Contagio dataset into three sub-datasets:

- A dataset used to train the defence model (45%)
- A dataset used to train the proxy model (45%)
- A dataset used to craft adversarial examples (10%)

### 2.8.2 *Malware detection algorithms*

To assess the malware detection models robustness, even in cross-technique settings (craft a sample on a NN and attack an SVM), we implemented three different algorithms with the following hyper parameters:

- RBF SVM (Support Vector Machine)
- Random Forest (50 estimators, max_depth = 4)
- Neural network (4 layer network 221→ 32→ 32 → 2)

### 2.8.3 *Evasion attack modus operandi*

#### 2.8.3.1 Features handling

As stated above, the feature vector is constituted of the number of occurrences of each tag in the PDF file. The vector is then standardized before feeding the ML model to stabilize the ML model training.

There are two important constraints to respect while generating the adversarial example in the current use case:

- **Positive Constraint:** adversarial examples are crafted by modifying the input, i.e. the features (number of occurrence of each of the 21 tags). The problem with PDF files is the fact that we cannot remove tags (decrease the value of a feature). So, the change is to be made only in the positive direction. As consequence, a clipping is applied to all generated examples to respect this constraint (except for Centered Initial Attack that takes into account the positive constraint by design).
- **Rounding constraint:** the features, before being standardized, are counters and therefore positive natural numbers. So, after crafting an example, we have to go the opposite way and transform the real number into a natural one. This step implies rounding the features to nearest number. This operation is like a clipping one and may degrade the effectiveness of an attack. It is taken into account in all our experimentations.

#### 2.8.3.2 Transferability of attacks

All the experiments we carried out were made in no-box setting (even if we display also the success rate of attacks on the proxy model). In other words, the attacker does not have access to the targeted model (not whitebox) and cannot request it to get feedback (not blackbox). The attacker uses a neural network as proxy to generate adversarial examples. Then, they are sent to the three different target models (SVM, Random Forest, Neural Network). From the literature, e.g. [75], we know that transferability property holds in such a scenario and we will verify it on the current PDF malware detection use case.

### 2.8.4 *Experimentations*

#### 2.8.4.1 Experimenting platform

All the code is written in python while relying upon popular libraries and frameworks like Tensorflow, scikit-learn. All the attacks and defences were implemented from scratch to have more flexibility and take into account the specific constraints of the considered use case (positive constraint and rounding constraint).

All the experiments were carried out on computer with a NVIDIA GeoForce RTX 2080 graphical unit and an Intel Core i9 @ 3.0 GHz CPU.

Given the relatively small volume of the Contagio dataset and the size of the ML models, each run is carried out very quickly (within some seconds only).

#### 2.8.4.2 Demonstrator running

To run the demonstrator there are some parameters to provide. Some depend on the attacks and others on the defences.

The attacks parameters are:

- The list of attacks to use to evaluate the defences robustness. Ex: ["FGSM", "CW"]
- The list of maximum thresholds to use for adversarial examples generation. Ex : [0.1, 1.0, 2.0, 10]
- The number of optimization steps (gradient descent steps)

The defence parameters are:

- Defence method: two options « adversarial training » and « feature scattering »
- The attack method to use for adversarial examples generation (from set {"FGSM", "iter-FGSM", "CW", "CIA"})
- The rate of clean data to replace with adversarial examples in adversarial training (ex: 0.5)
- The number of iterations of adversarial retraining of models. Ex : 2.

**Remark:**

When we adopted feature scattering, we adversarially trained a SVM and a random forest to robustify them (added adversarial examples crafted using a NN to their training dataset). Therefore, the attack to use for adversarial examples generation is a parameter to provide all the time, whatever is the chosen robustification method.

### 2.8.5 *Results and discussion*

The results (output of the demonstrator) are curves displaying the transferability success rate of the selected attacks against the eight mentioned models. This success rate is given as a function of the logarithm of the maximum perturbations set up when running the demonstrator. Here are some of the obtained results.

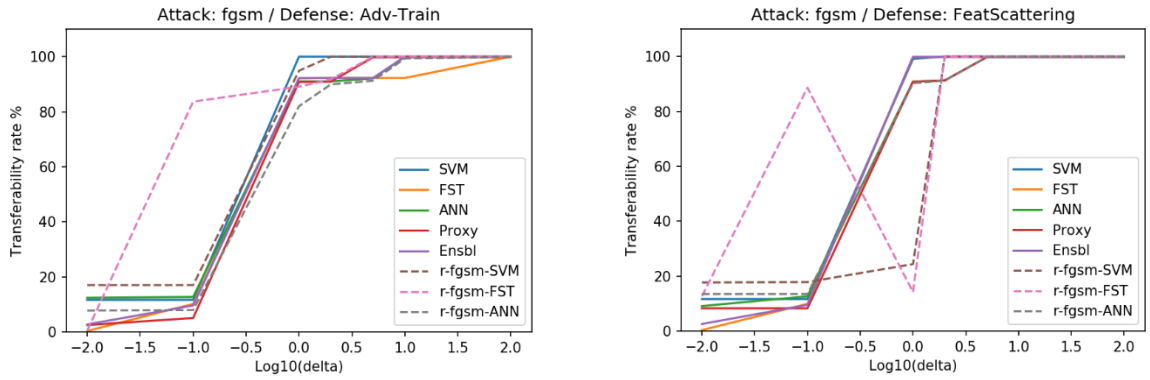



Figure 14: FGSM attack vs. Adversarial training (adv examples generated through FGSM attack) + Feature Scattering (hybrid with adv examples generated through).

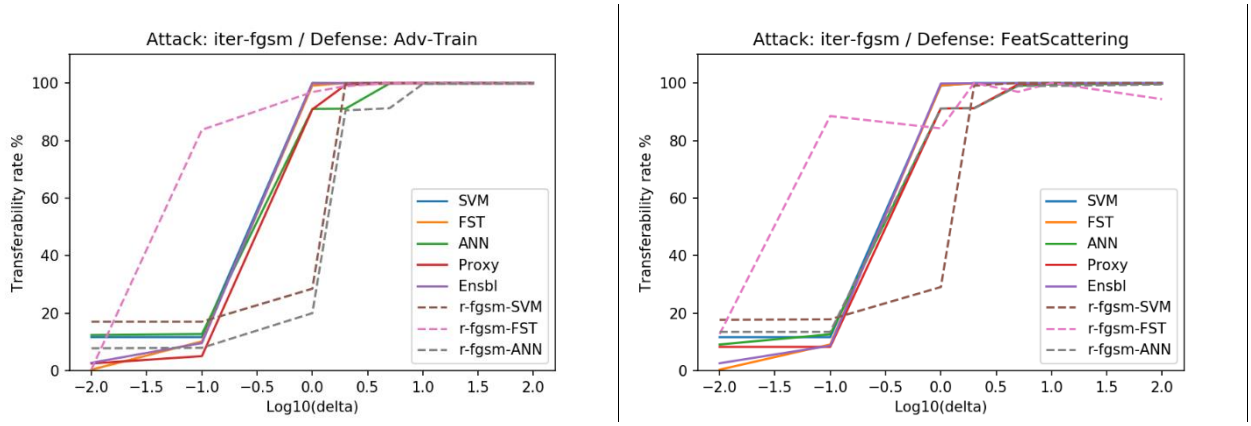



Figure 15: iter-FGSM attack vs. Adversarial training (adv examples generated through FGSM attack) + Feature Scattering (hybrid with adv examples generated through).

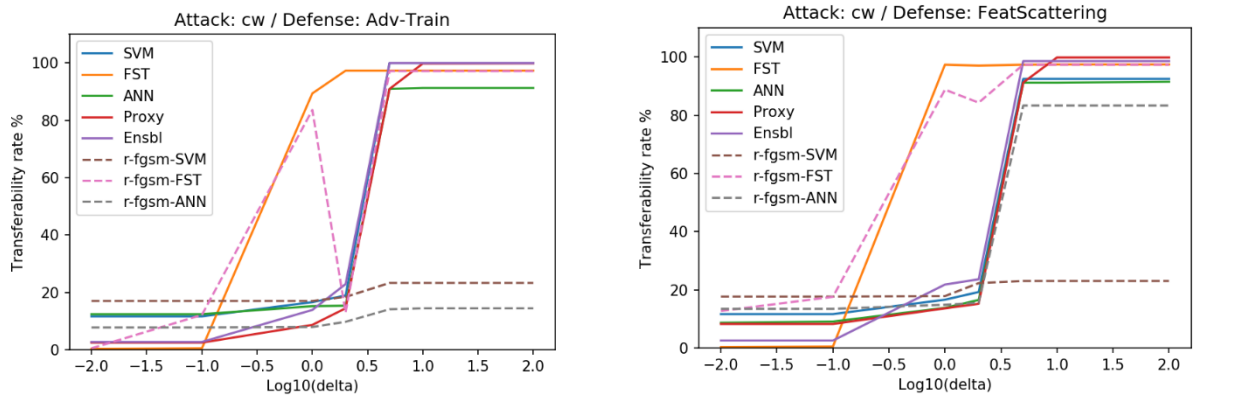



Figure 16: C&W attack vs. Adversarial training (adv examples generated through FGSM attack) + Feature Scattering (hybrid with adv examples generated through).

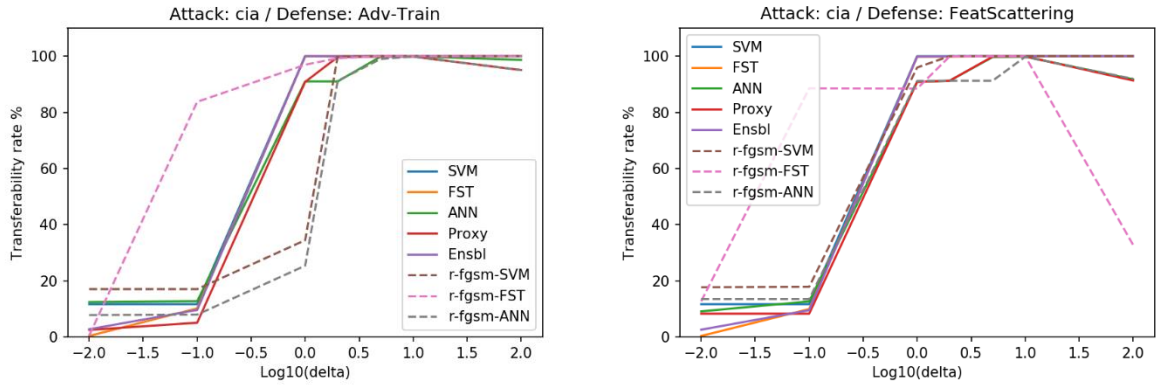



Figure 17: CIA attack vs. Adversarial training (adv examples generated through FGSM attack) + Feature Scattering (hybrid with adv examples generated through).

**Discussion of the results:**

From the curves above, we can note the following remarks:

- As expected, the attacks are transferred from a model to a different model, from neural networks to random forests (cross-technique transferability).
- The good news is that when the attack used for test is different from the one used for adversarial training, the robustness is improved (except for random forest), either using adversarial training or feature scattering.
- The difference between classical adversarial training and feature scattering is not striking. This is the case whatever the attack used to forge the adversarial examples. This is probably because of the fact that PDF malware case is a binary classification and therefore the inter samples coupling of features in feature scattering approach is limited.
- The C&W attack that is often referred to as state of art attacks is not that successful here because we adopted infinite norm and cropped the adversarial perturbation to the fixed threshold. The attack is therefore degraded.
- Adversarially trained Random Forests are not robust at all in almost all the cases. This is probably due to an overfitting of these models and not well generalizing to others examples.
- The most robust solution is the adversarially trained neural network or SVM (a SVM can be seen as a simple neural network indeed).
- The ensemble model (voting among NN, SVM and Random forest) without robustification is often presented a robust approach but this is not what we see here. Implementing it with the cost of many models running at the same time is not that interesting.

### 2.8.6 *Conclusion*

Adversarial training is a good approach to robustify AI/ML solutions but it is dependent on the attack used to generate the adversarial examples. The result may be robust against that attack but not against another.

A better alternative is therefore to use a mix of adversarial examples generated using different types of attacks.

When the available budget makes it possible (there are enough computing resources and memory), an ensemble of defences, even without being robustified, is also a good approach.

# Chapter 3 Explainability enhancing mechanisms

In this Section, we extend the results from D7.4. In D7.4, we present ShapKit, a Python module dedicated to local explanation of machine learning model and we present a new component called hybrid oracle explainer and based on decision trees, which has been applied to Intrusion Detection Systems.

In D7.5, we present a component that uses ShapKit based on Dash, a Python module that is used to provide web applications. The component helps users who are not familiar with Python to realize the local explanation of machine learning model with ShapKit. It accepts both sklearn and tensorflow formalism and can be applied to binary classification, multi-class classification and regression tasks. The user can adapt the reference population to his/her needs. After presenting the component, we illustrate its use on a case dedicated to denial of service attack detection. In the second part of the Section, we present some supplemental explorations of surrogate-type methods for explainable artificial intelligence. We apply it in the context of cybersecurity.

## 3.1 Local explanation of machine learning model

The technical elements of this part are developed in D7.4. In D7.5, we present a Dash application that is developed for the demonstration of the approaches. We demonstrate the application on a cyber-security use case dedicated to DoS (Denial of Service) attack.

Machine Learning models are used for various applications with already successful results. Unfortunately, a common criticism is the lack of transparency associated with these algorithms' decisions. This is mainly due to a greater interest in performance (measurable non-specific tasks) at the expense of a complete understanding of the model. Global method of interpretability aims at explaining the general behaviour of a model, where as a local method focuses on each decision of a model. The agnostic category (also called post-hoc explanation) considers the model as a black box. On the other hand, inherent or non-agnostic methods can modify the structure of a model or the learning process to create intrinsically transparent algorithms.

Local explanations focus on a single instance and examine what the model predicts for this input, and explain why. This application focus on additive local explanation: for one given instance, we search to explain the deviation of its prediction from the prediction of an average instance of a reference population by the sum of contribution of features.

It is important to keep in mind that the methods used explain **the reasoning of the model, not the reality.**

### 3.1.1 *Technical description*

Shapley values offer a solution for the local explanation from additive feature importance measure class ensuring desirable theoretical properties. A prediction can be explained by assuming that each feature value of the instance is a ``player'' in a game where the prediction is the payout. The objective is to fairly distribute the payout among all features to obtain the prediction. One can make the following correspondence between game theory and model interpretability:

- The features are the players;
- The model to explain is the game;
- The feature attribution is the gain attribution.

Following the links with cooperative game theory, Shapley's values (see [Shapley, 1953]) are the only indicators verifying local accuracy, i.e. that the sum of the feature contributions is equal to the prediction, missingness, i.e. if a feature has not had an impact on the output of the model, then its contribution will be zero and symmetry, i.e. that if two features have an identical effect when observed in any situation, then the Shapley values for the features must be the same. A major challenge for Shapley values is the overall computational cost: the potential coalition, and so the global cost (i.e. the total number of calculations), grows exponentially as a function of the number of features. Some authors propose some algorithms to approximate the Shapley Values (e.g. [84], [80], [76], [81]). We use **ShapKit [pypi.org]**, a Python module that uses both Monte Carlo approaches and a stochastic gradient algorithm of a Weighted Least Square Optimization problem. In this Dash application, we only use the Monte Carlo approach. The algorithms are described in [79] and in D7.4.

### 3.1.2 *Component overview*

The Table below provides the dependencies of the component that was tested.

| Module/Langage | Version |
|---|---|
| Python | 3.8.3 |
| dash | 1.14.0 |
| dash-core-components | 1.10.2 |
| dash-html-components | 1.0.3 |
| plotly | 4.9.0 |
| catboost | 0.24.4 |
| shapkit | 0.0.4 |
| numpy | 1.18.5 |
| pandas | 1.1.0 |
| pickleshare | 0.7.5 |
| tensorflow | 2.3.0 |

Table 10: Interpretability web application dependencies.

In this component, we propose an application dedicated to local explanation for three supervised machine learning tasks: regression, binary classification and multi-label classification. For regression, the reference population is the whole population. In this case, the Shapley values computed are about the difference between the prediction made for an average instance and the instance of interest. For binary classification, the reference population is the opposite of the class predicted for the instance of interest. We are interested here in studying the difference between the prediction for the opposite class and the instance of interest. We look at the features that contribute the most to change the class. For multi-label classification, we compute the Shapley value according the maximum probability of the instance of interest and we consider as reference population the population predicted as another class than the instance to explain or one class chosen by the user. Moreover, two models formats can be used: Sklearn format (including module like catboost) and Tensorflow/Keras format.

In Figure below, we give the general overview of the main panel of the application. The parameters are given below:

- **Drag and drop or Select Data Files:** Dataset that will be used in the application. Can be csv, xls, txt or tsv format.
- **Header case:** Dataset contains header or not. If the Header case is checked, then the first row of the datasert are considered as the header.
- **Choose the model:** Path to the model to explain. Can be a pickle (for sklearn model format) or a h5 (for Keras model format) file
- **Choose the task:** Model task, can be regression, binary classification and multi-label classification
- **Choose the instance:** Index of the instance to explain
- **Choose the size of the reference population:** size of the reference population. By default, the whole reference population is used.
- **Choose the number of attributes to show:** Number of features to plot on the graph. By default, all features are represented.
- **Choose model format:** Choose the format of the model, can be Sklearn format or Keras format.
- **Choose the number of iterations:** Number of Monte Carlo iteration for the approximation of the Shapley Values
- **Choose the reference population** (only in the case of the multi-label classification): By default, the reference population is the instance predicted in an other class that the instance of interest. If indicating an integer, the reference population is the individuals predicted in the class corresponding to the integer.
- **Choose the threshold of classification** (only in the case of the binary classification): Threshold of the score for the final prediction.
- **Download Shapley Value in CSV format:** Download the computed Shapley Values in a csv file.

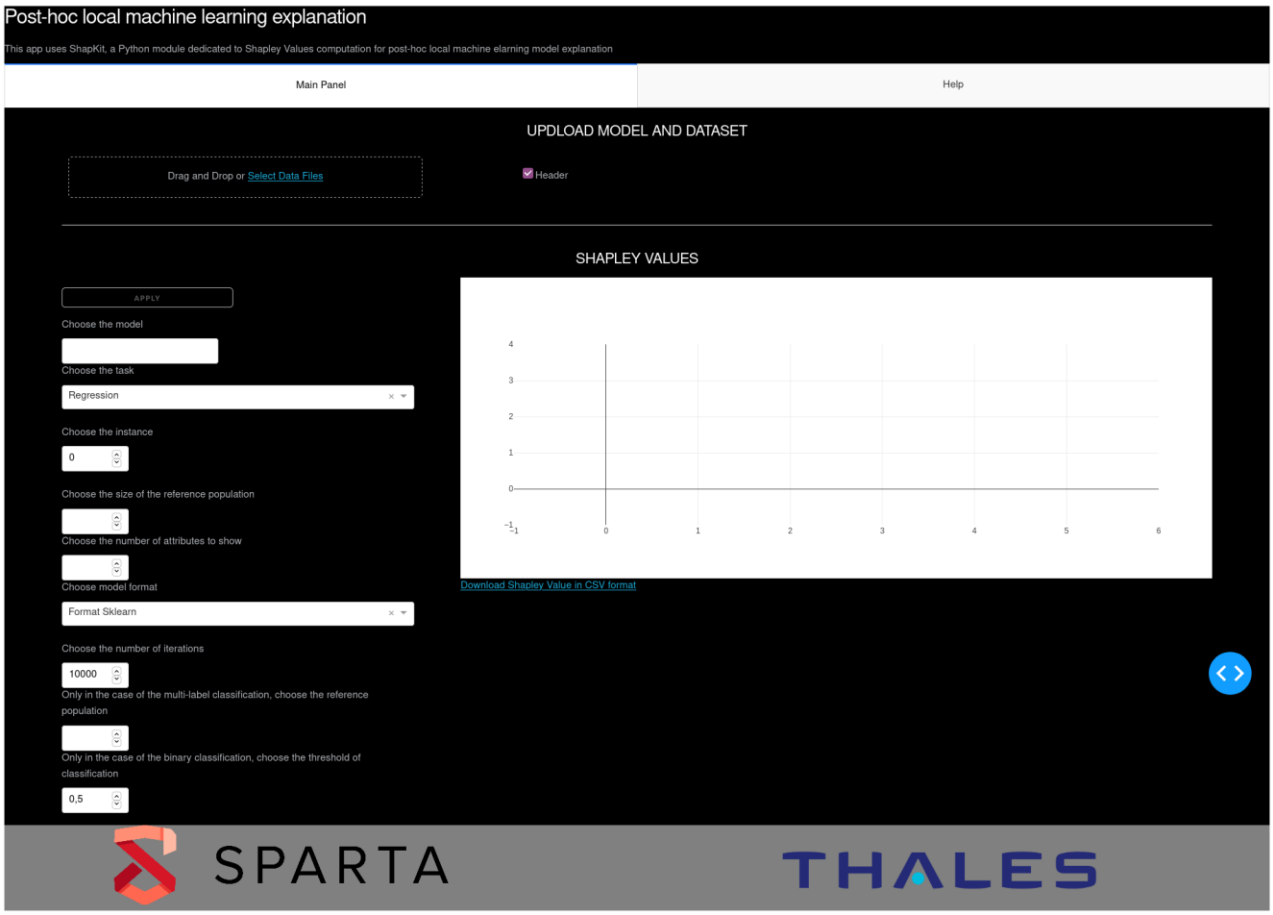



Figure 18: Local explanation application overview

When the user click on Drag and drop or Select data, it opens a new window where the user can upload her dataset.

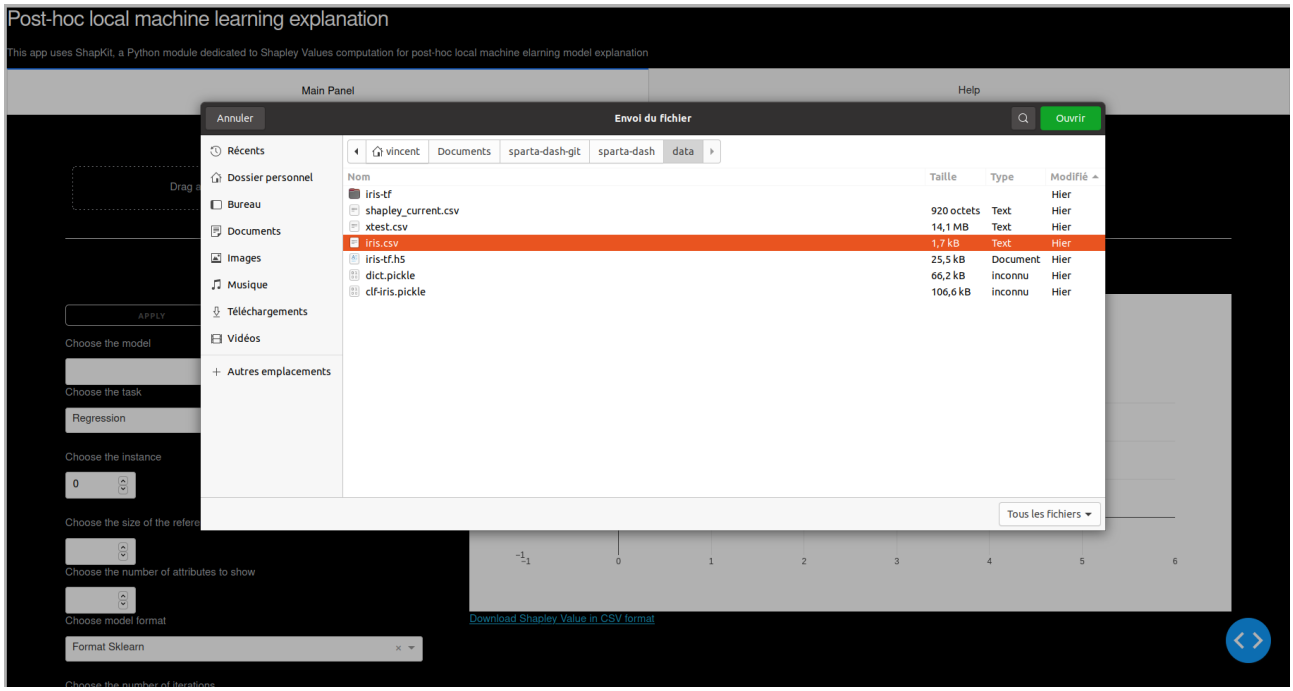



Figure 19: Uploading data in the local explanation application

In the Figure below, we provide a screenshot of the help tab.

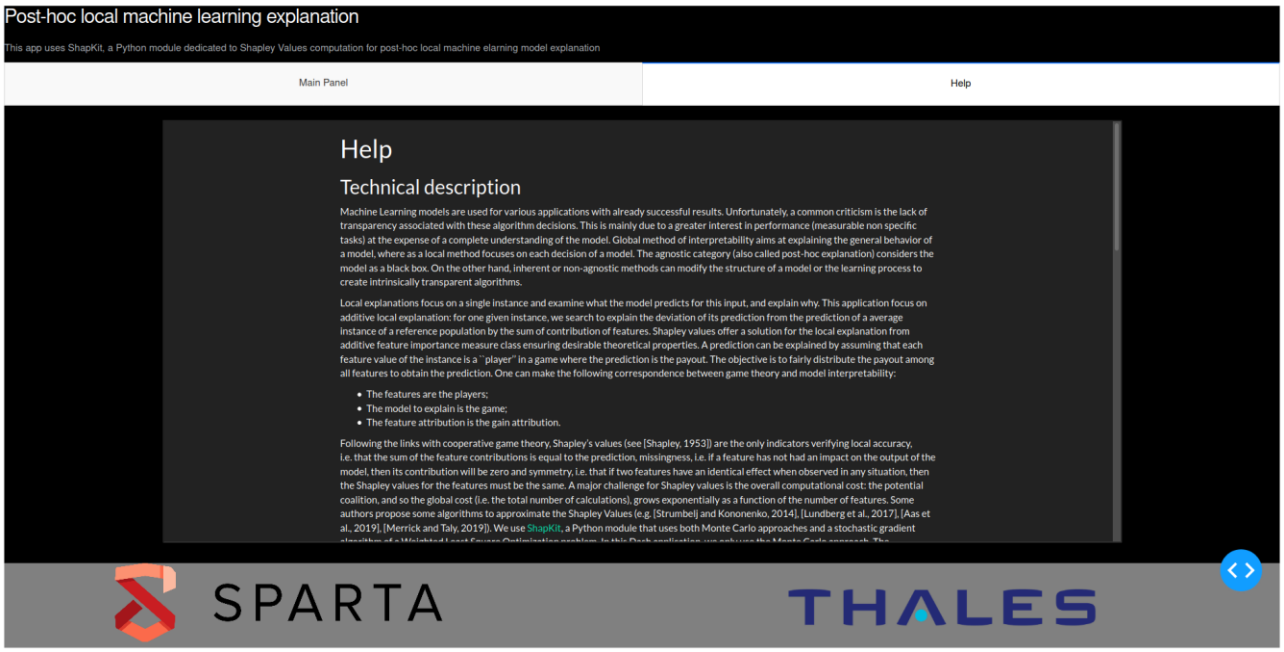



Figure 20: Help tab of local explanation application

### 3.1.3 *Component usage on toys datasets*

#### 3.1.3.1 Regression task

We illustrate the application of the component on the Boston housing prices dataset [77]. The target variable is the median house value for Boston districts. There are 506 instances.

| Features | Value |
|---|---|
| CRIM | per capita crime rate by town |
| ZN | proportion of residential land zoned for lots over 25,000 sq.ft. |
| INDUS | proportion of non-retail business acres per town |
| CHAS | Charles River dummy variable (= 1 if tract bounds river; 0 otherwise) |
| NOX | nitric oxides concentration (parts per 10 million) |
| RM | average number of rooms per dwelling |
| AGE | proportion of owner-occupied units built prior to 1940 |
| DIS | weighted distances to five Boston employment centres |
| RAD | index of accessibility to radial highways |

| Features | Value |
|---|---|
| TAX | full-value property-tax rate per $10,000 |
| PTRATIO | pupil-teacher ratio by town |
| B | 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town |
| LSTAT | % lower status of the population |
| MEDV | Median value of owner-occupied homes in $1000's |

Table 11: Features of Boston Housing dataset

The dataset is divided into a training and in a testing set. A Random Forest is trained on the training set. On the Figure below, we represent the Shapley values of the first instance of the testing set (see "choose the instance" parameter). We sample 1000 instances of the testing set to build the reference population ("choose the size of the reference population" parameter). As the parameter "choose the number of attrtibutes" is not fiiled out, all the features are represented on the plot. The Random Forest has been trained with sklearn, so we use Sklearn format as model format. We use 10000 iterations of Monte Carlo algorithm to approximate the Shapley Values. The two last parameters are dedicated to classification tasks, and so have no impact on the outputs when we explain regression model.

For the chosen instance, the model predicts that the price is 28,75. The prediction for the reference population is equal to 22,53. The Figure below helps to understand the deviation between these two predictions. On this Figure, the x-xasis gives the Shapley Values, the y-axis the features and the value of the value for the instance of interest, e.g. for the instance of interest, the average number of rooms per dwelling (RM) is equal to 6.85. The two features that contribute the most to the increase of the prediction for the instance to explain are the average number of rooms per dwelling and percentage lower status of the population.

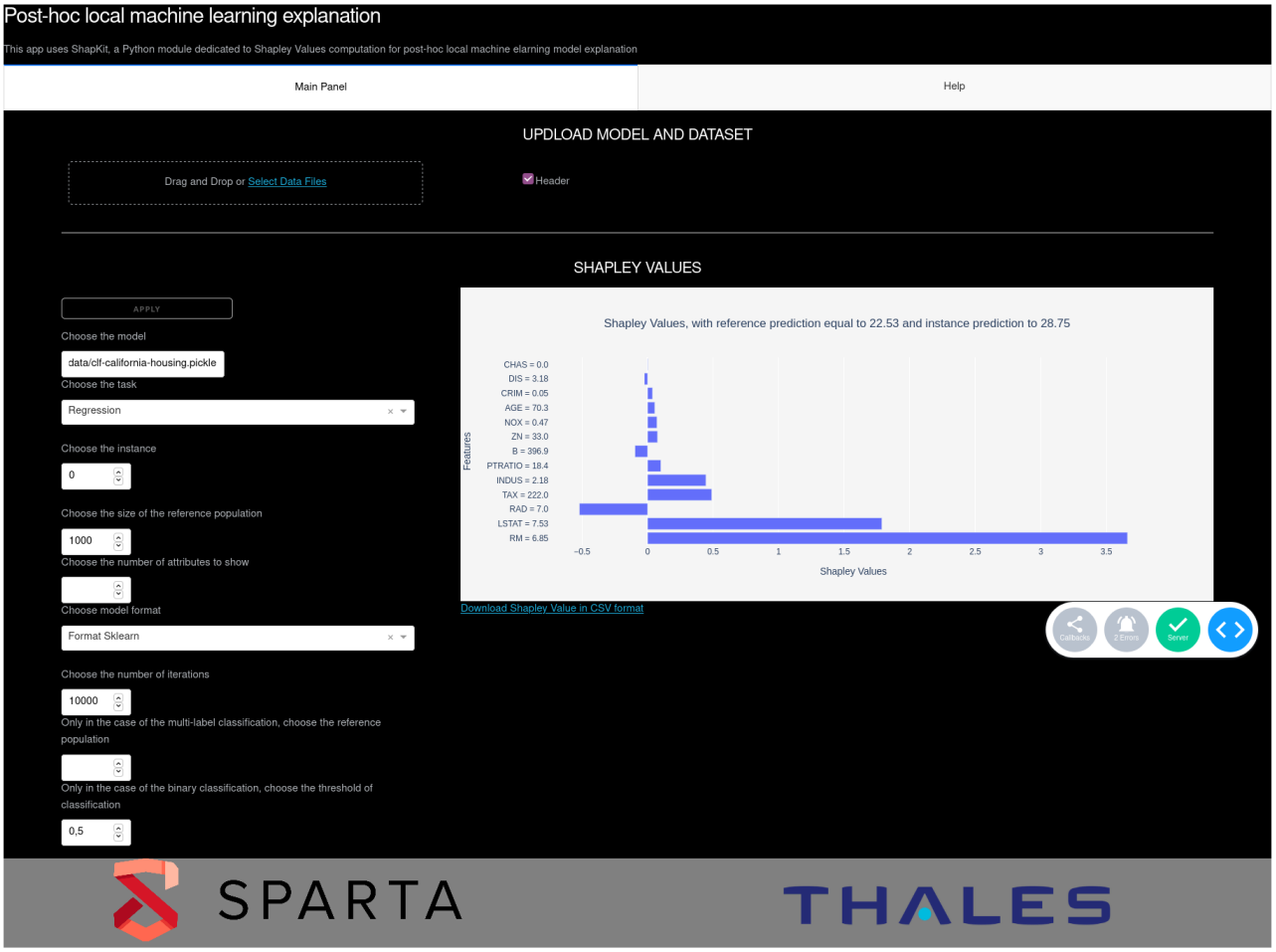



Figure 21: Regression task: Shapley Values for a Random Forest trained for the Boston housing prices dataset

### 3.1.3.2 Binary classification task

The binary classification task is illustrated on the cancer breast dataset [83]. The features are computed from a digitized image of a fine needle aspirate of a breast mass. They describe characteristics of the cell nuclei present in the image. The target is malignant or benign. The features are given by the following elements:

- radius (mean of distances from center to points on the perimeter);
- texture (standard deviation of gray-scale values);
- perimeter;
- area;
- smoothness (local variation in radius lengths);
- compactness (perimeter^2 / area - 1.0);
- concavity (severity of concave portions of the contour);
- concave points (number of concave portions of the contour);
- symmetry;
- fractal dimension (“coastline approximation” - 1).

The mean, standard error, and “worst” or largest (mean of the three worst/largest values) of these features were computed for each image, resulting in 30 features. For instance, field 0 is Mean Radius, field 10 is Radius SE, field 20 is Worst Radius.

There is 569 instances. Around 60% of the instances are benign, that corresponds to the class one, the remaining instances are malignant. The dataset is divided into a training set and in a testing set. A Random Forest is learnt on the training set. In the sections below, we explain the prediction of one random instance according two reference populations.

### Default reference population

For the default reference population, we compute the class predicted for the instance if the decision threshold is equal to 0.5 (parameters “choose the threshold of classification”). The instance whose the prediction is explained is the second instance of the testing set (“Choose the instance=1”). If there were more than 1000 instances in the reference population, we would have drawn 1000 at random according to the parameter "choose the size of the reference population". Here, there are less than 1000 instances in the reference population, so it is kept entirely. The parameter “Choose the number of attributes” is not filled in, so all the features are represented on the plot. The model is trained with sklearn, so we use Sklearn format. We use 10000 iterations of Monte Carlo to approximate the Shapley Values.

The score predicted by the model for the instance of interest is equal to 0.61. With this default threshold, the instance is predicted in the class one, i.e. benign. In this context, the reference population contains the instances whose score is smaller than 0,5 according the Random Forest model. The Shapley Values explain the contribution of each features to deviation between the score of the average instance of the reference population and the instance of interest. The attributes that contribute the most to the increasing of the score are the worst area, the worst radius and the mean concave points. If we look at the repartition fo the features for the benign and the malignant, these features are small for the the malignant class, but strong for the benign class.

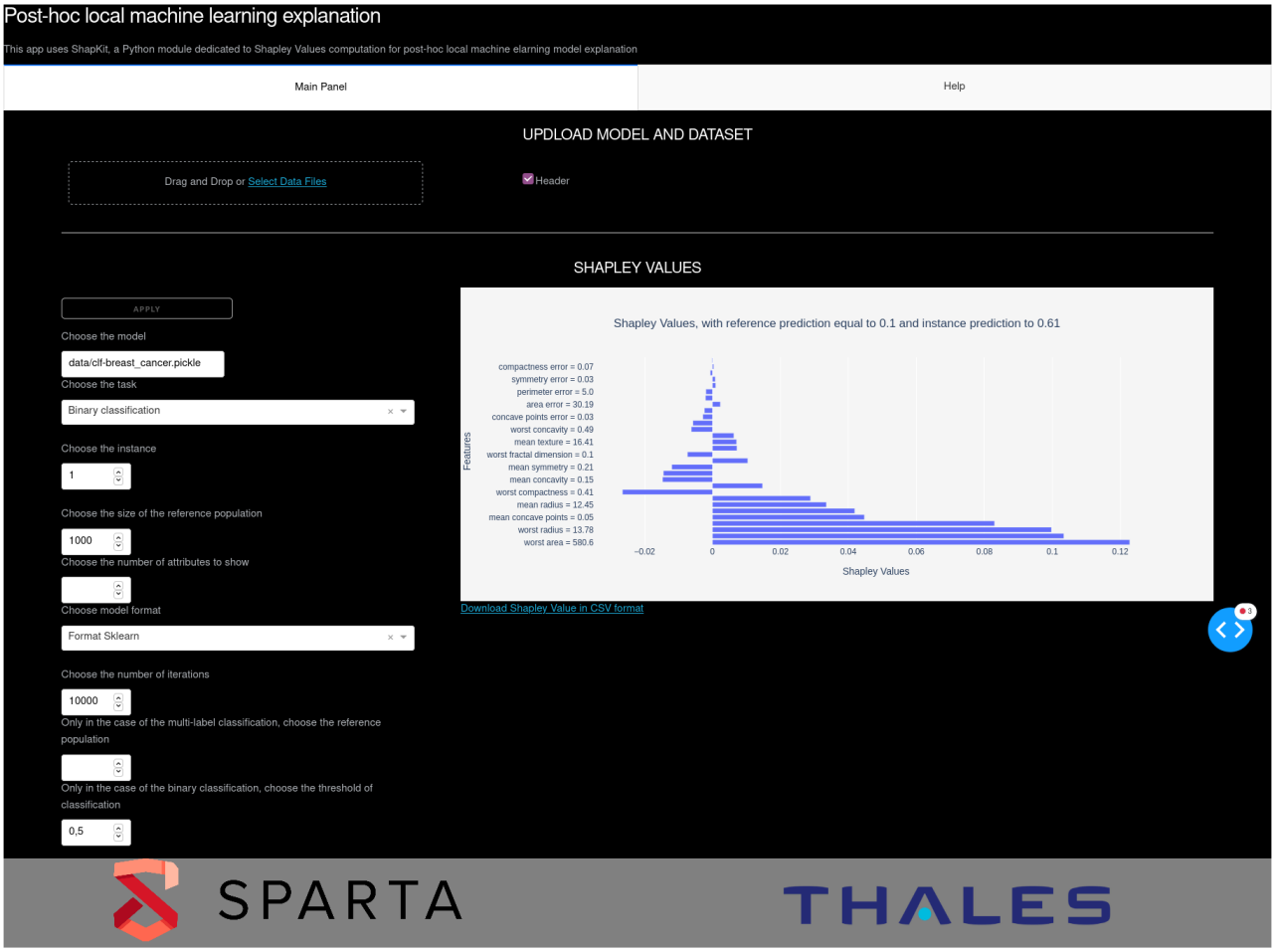



Figure 22: Binary classification: Shapley Values for a Random Forest trained for the cancer breast dataset when the threshold used to select the reference population is 0.5

## Chosen reference population

For the new reference population, we compute the class predicted for the instance if the decision threshold is equal to 0.65. The score predicted by the model for the instance of interest is equal to 0.61, so with this threshold, the instance is predicted in the class zero, i.e. malignant. In this context, the reference population contains the instances whose score is greater than 0,65 according to the Random Forest model, and so predicted as benign. We keep all other parameters fixed. In this context, the Shapley Values are given in the Figure below. The values of the worst compactness, of the worst concavity and the worst concave point are the elements that decrease the most the score of the model for the instance compared to the reference population.

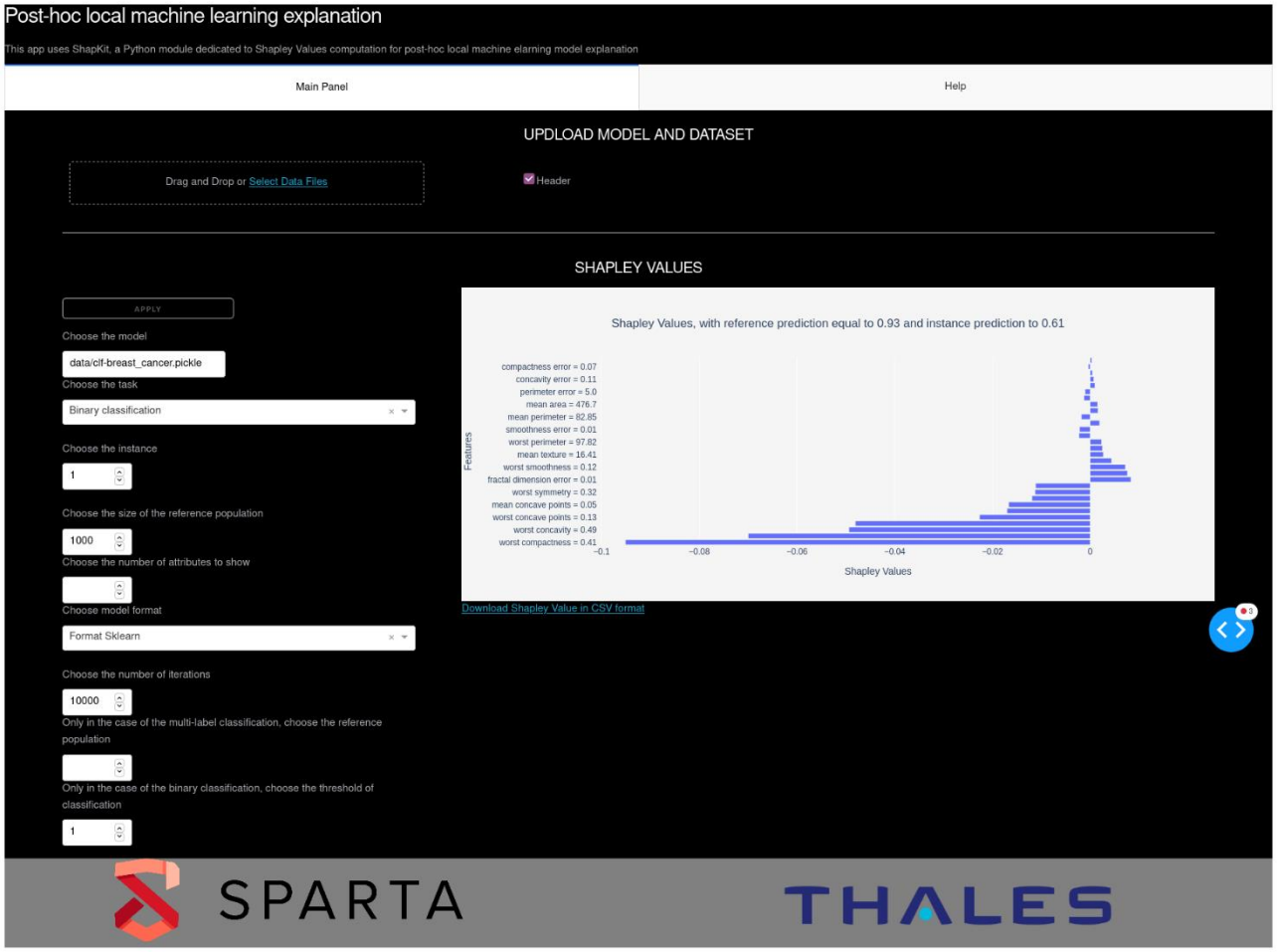



Figure 23: Binary classification: Shapley Values for a Random Forest trained for the cancer breast dataset when the threshold used to select the reference population is 0.7

### 3.1.3.3 Multi-class classification task

The multi-class classification task is illustrated on the iris dataset [78]. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are not linearly separable from each other. The three classes are iris-setosa, iris-versicolour and iris-virginica. There are 4 features: sepal length in cm, sepal width in cm, petal length in cm, petal width in cm.

The data set is divided in two datasets : a training set and a testing set. A Random Forest with 100 trees is learnt on the training set. An instance is randomly chosen in the testing set. We use the application to explain its prediction. The model accuracy on the testing set is around 0.95.

#### Default reference population

For an instance x, the model f predicts a vector of size three (there are three classes in the use case). To compute the Shapley Values, we keep the value corresponding to the arg max of this vector and we study the deviation of this score with the score of the reference population, as explain in the Figure below. On this Figure, the instance of interest is the first instance. According to the model, its maximum score is obtained with the first class. The Shapley Values will be computed according to the deviation for this score between the instance of interest and the reference population. By default, on this example, the reference population will contain the instances predicted in the second or in the last class.

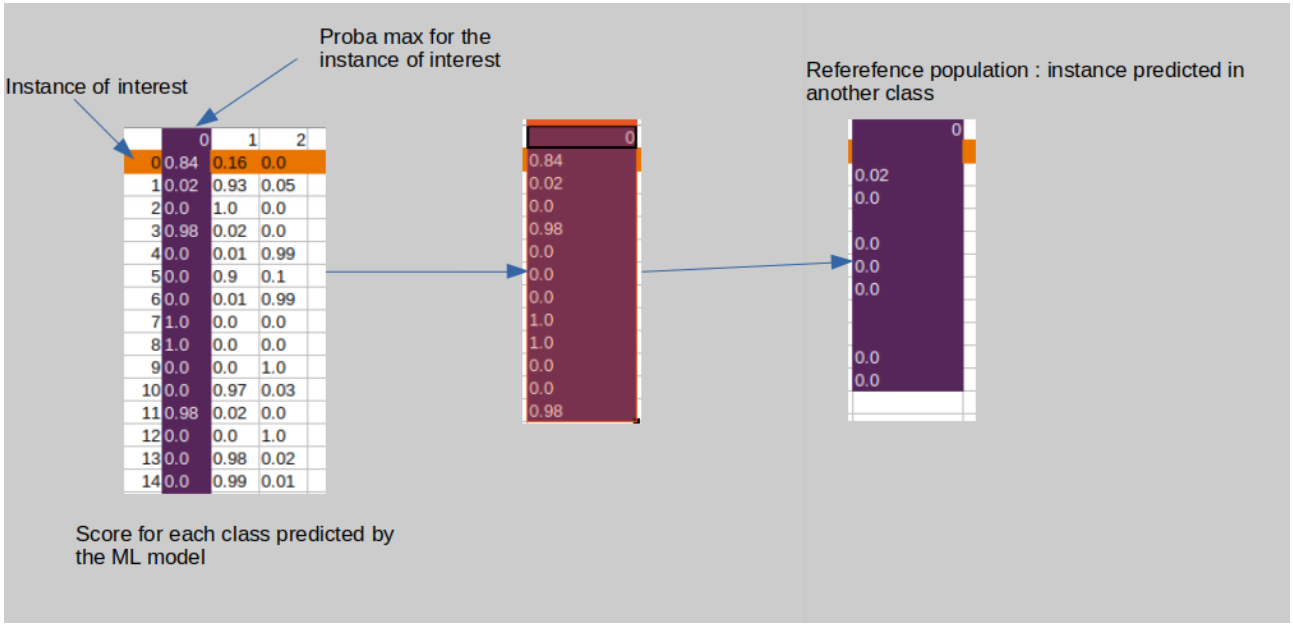



Figure 24: Default reference population used by the local explanation application for the multi-label classification task.

By default, the reference population is all the instances predicted in another class than the instance of interest (“Choose size of reference population” not fill).

Figure below gives the Shapley Values in this context for the first instance of the dataset. On this example, we keep the whole reference population as the size reference population is not used. Likewise, as the parameter number of attributes is not filled in, all the features are represented on the plot.

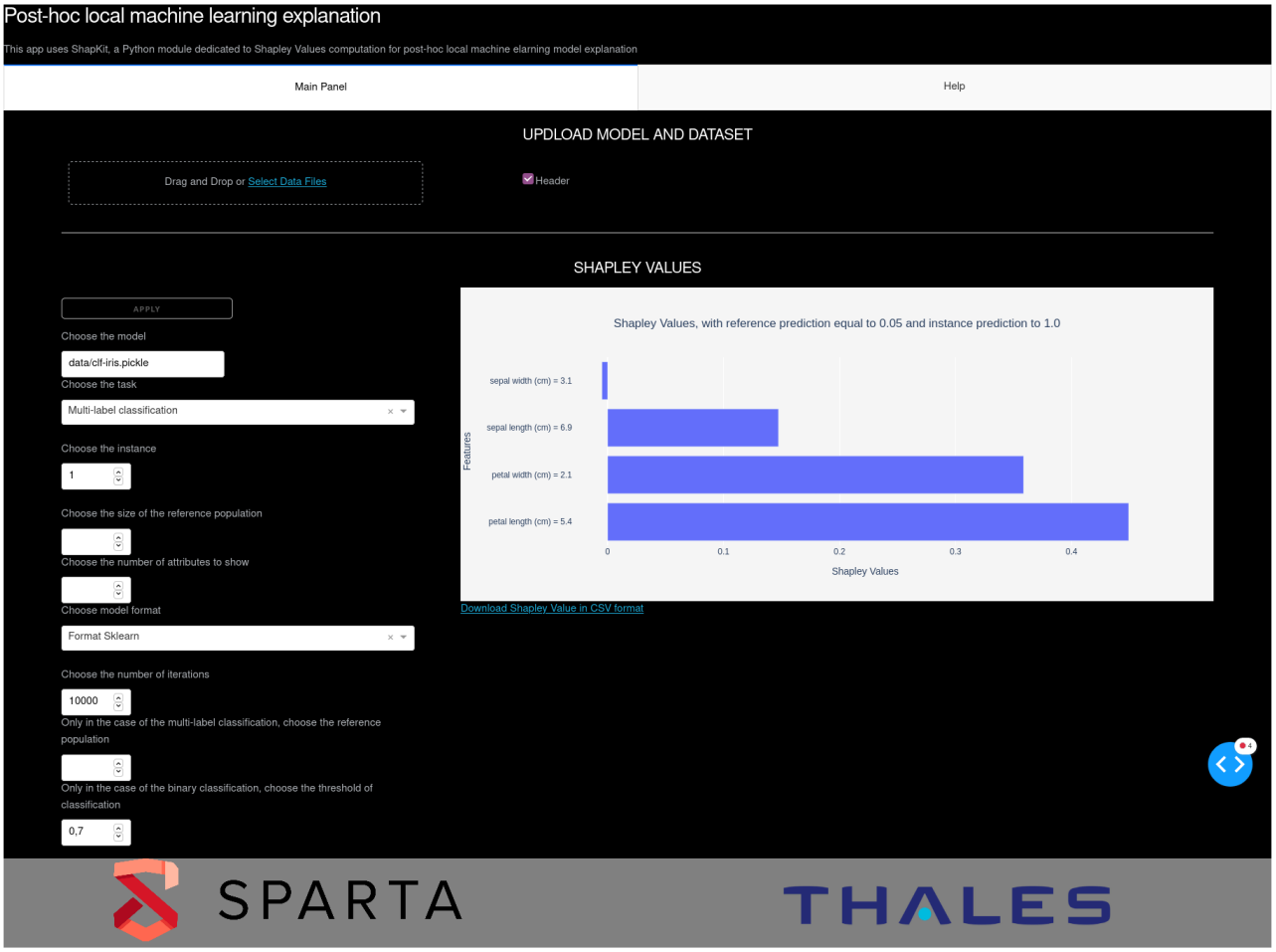



Figure 25: Multi-class classification: Shapley Values for a Random Forest trained for the iris dataset when the t the reference population is the default one, i.e. the instances predicted in another classes that the instance to explain

## Chosen reference population

For an instance x, the model f predicts a vector of size three (there are three classes in the use case). To compute the Shapley Values, we keep the value corresponding to the arg max of this vector and we study the deviation of this score with the score of the reference population, as explained in the Figure below. Compared to the previous section, the reference will change. The reference population will be all the instances whose score is higher for the second class. The reference population is given by the score predicted for class 1, which the class where the instance of interest is predicted.

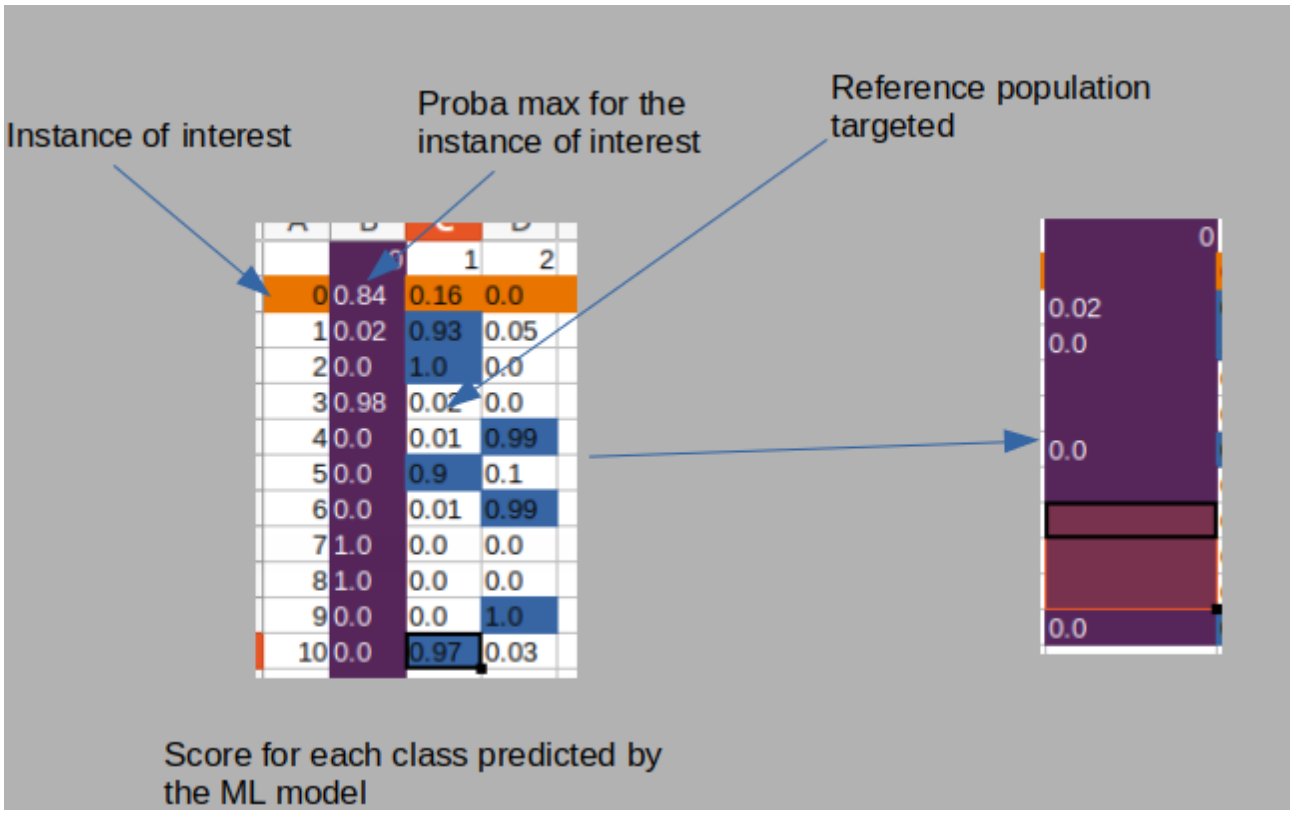



Figure 26: Instances predicted in the second class as reference population used by the local explanation application for the multi-label classification task

On the Figure below, we apply it on the second instance of iris dataset and consider as reference population the instances predicted in the class 2, which is the same class as the instance of interest. The petal width value for the second instance increase the score predicted by the model compare to the “average” instance predicted in class 2 by the model.

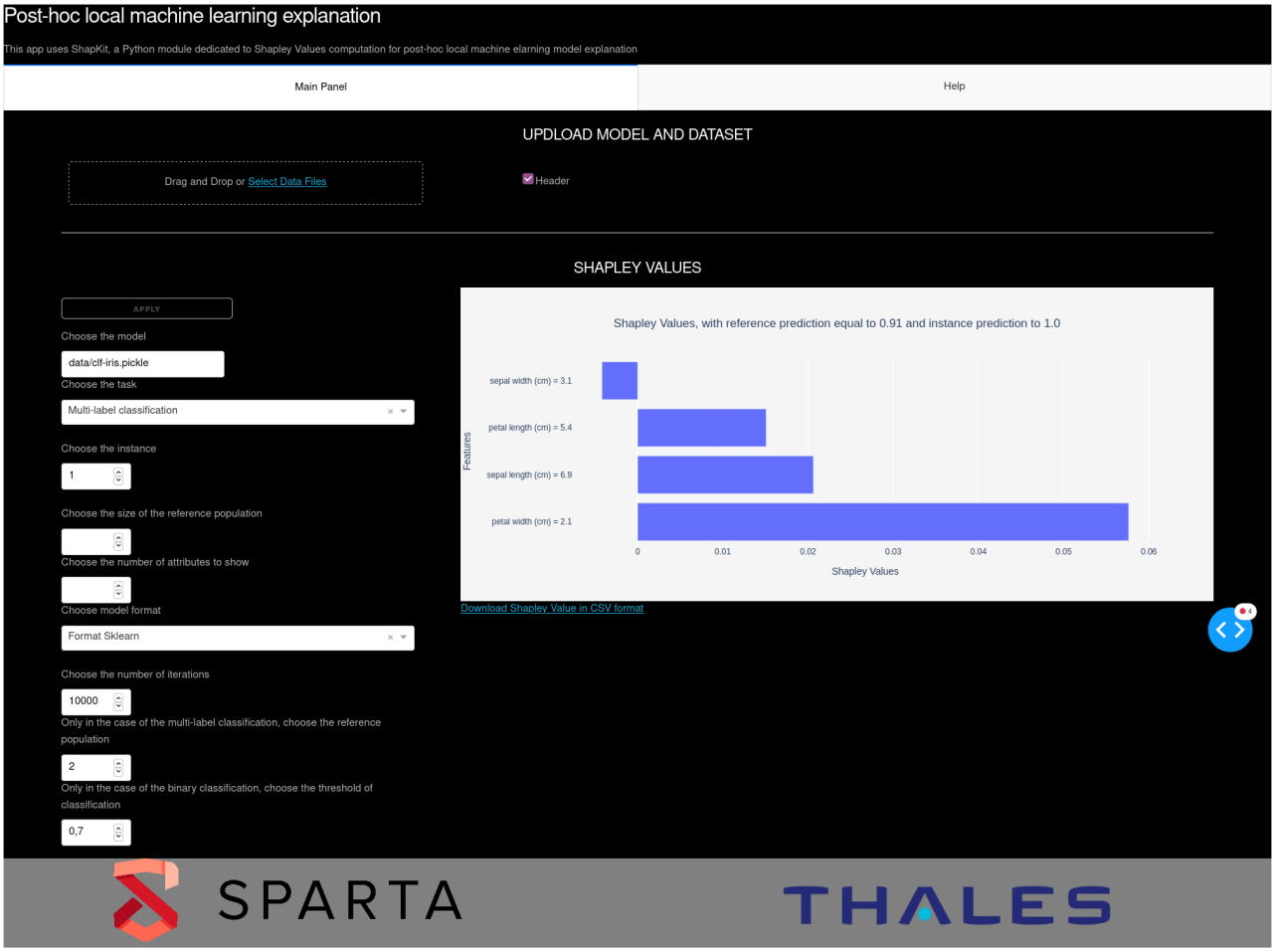



Figure 27: Multi-class classification: Shapley Values for a Random Forest trained for the iris dataset when the reference population is the instances predicted in the class 2

### 3.1.3.4 TensorFlow/Keras format model

In the same way as for the models using the sklearn format, it is possible to use some models in the TensorFlow/Keras format. They have to be saved in h5 format. For the illustration, we use the iris dataset and use the model described by the Figure below. It is a simple sequential neural network with two dense layers. The first layer uses a ReLu activation function and the second a softmax activation function. The loss used is the cross-entropy.

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_5 (Dense)              (None, 8)                 40
_________________________________________________________________
dense_6 (Dense)              (None, 3)                 27
=================================================================
Total params: 67
Trainable params: 67
Non-trainable params: 0
_________________________________________________________________
```

Figure 28: Model architecture to illustrate the use of TensorFlow/Keras format model in the local explanation application

Then we upload the model in the application and work on the local explanation of the second instance. The reference population contains the instances predicted in another class than the instance 2. The only thing to change from the previous part is to change the box "choose model format" and indicate Tensorflow format, as illustrated on the Figure below.

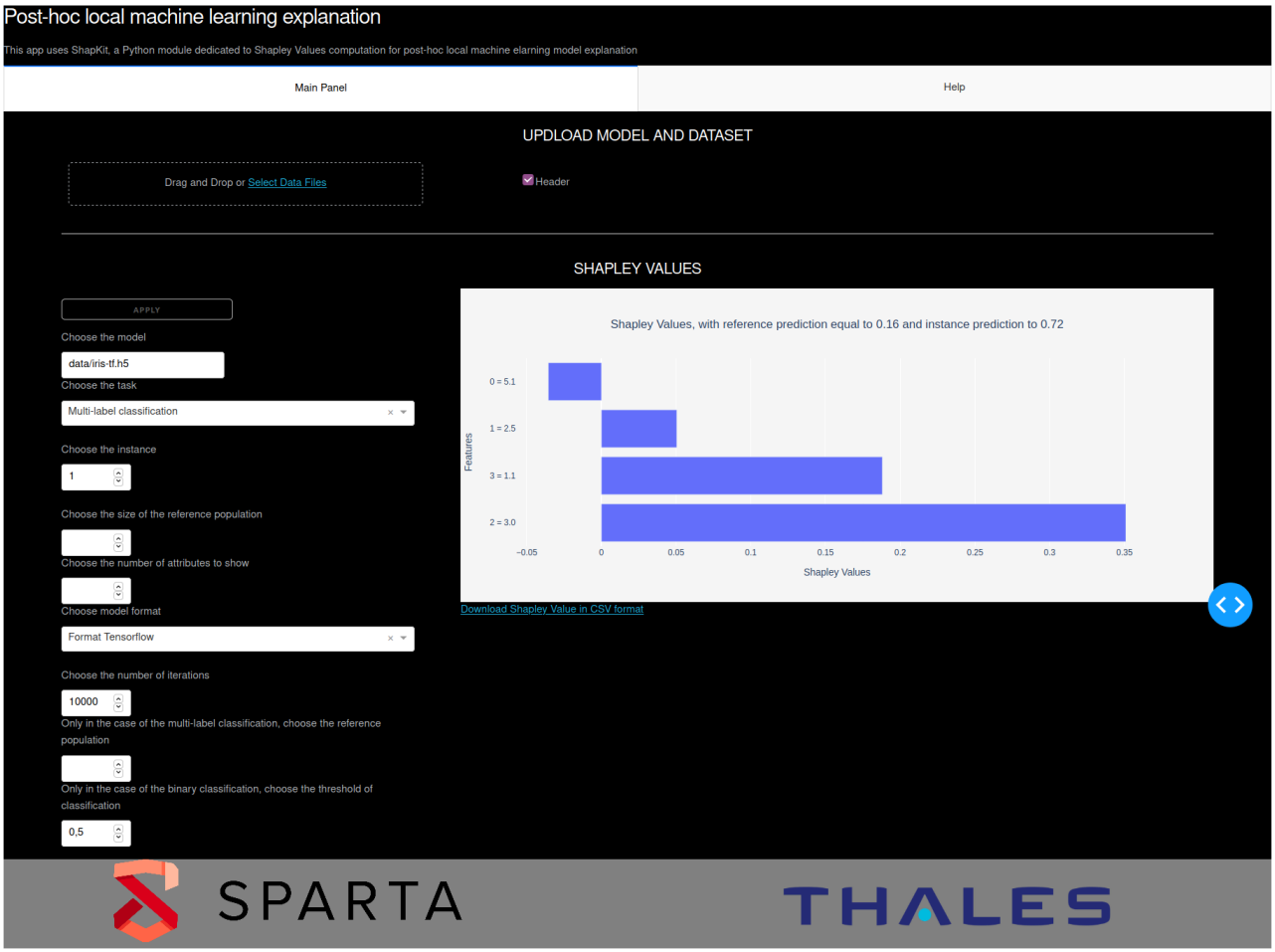



Figure 29: Multi-class classification: Shapley Values for a Neural Network trained for the iris dataset when the reference population is the instances predicted in another class that the instance of interest

### 3.1.4 *Demonstration on a cybersecurity use case*

#### 3.1.4.1 Use case presentation

In this use case, we are interested in the detection of network intrusions, protecting a computer network from unauthorized users, including perhaps insiders. The objective is to build a predictive model capable of distinguishing between illegitimate (intrusions) and legitimate connections and to be able to understand the prediction made and the global behaviour of the model. This second task is important for the operational use of the model, to be able to detect some false positives and effectively categorise the attacks. For this use case, we can use the KDD Cup 1999 Data http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, which includes a wide variety of intrusions simulated in a military network environment. For this use case, the common features are the basic features of individual TCP connections (e.g. number of seconds of the connection, type of protocol, etc.), some content features within a connection suggested by domain knowledge (e.g. number of failed login attempts, number of ``root'' accesses, etc.) and traffic features computed using a two-second time window (e.g. number of connections to the same host as the current connection in the past two seconds, number of connections to the same service as the current connection in the past two seconds, etc.). Four attack types are considered in KDD Cup 1999 dataset:

- DoS: denial-of-service, e.g. syn flood;
- R2L: unauthorized access from a remote machine, e.g. guessing password;
- U2R: unauthorized access to local superuser (root) privileges, e.g. various ``bufferoverflow'' attacks;
- probing: surveillance and other probing, e.g., port scanning.

In this report, we focus on DoS attacks, without distinguishing between the different categories of DoS attacks. Denial-of-service attack is a cyber-attack in which the attacker seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to the Internet. Denial of service is typically accomplished by flooding the targeted machine or resource with superfluous requests in an attempt to overload systems and prevent some or all legitimate requests from being fulfilled. The data are labelled: one class for the normal connection and the second for DoS attack, whatever the type of attacks it is (smurf, syn flood, etc.). A ML model is trained to distinguish between normal connection and DoS attacks with catboost. This model performs perfectly on a testing set (AUC and accuracy equal to 1): it is an easy task on this sample of data. Then, we use Shapley Values with two objectives: understand what are the important elements that lead to an alert and use these elements to try to refine the characterization of the attack undergone. The training set (resp. the testing set) contains 342 114 rows (resp. 146621 rows). Both datasets contain around 80% of DoS attacks and 39 columns, including the target.

#### 3.1.4.2 Application of the component

We will apply the component on two instances that have been predicted as DoS attacks by the model. For the first instance, the protocol uses is ICMP. The feature src_bytes is the number of data bytes from source to destination. For this instance, this number is strong. The count is the number of connections to the same host as the current connection in the past two seconds. Again, for this instance, this number is strong, as srv_count, which is the number of connections to the same service as the current connection in the past two seconds. We select as references sub-population 1000 random instances (see “Choose the size of the reference population parameter) that have been predicted as normal by the ML model. On the plot, we represent only the ten first ranking by the absolute values of the Shapley Values (see “Choose the number of attributes to show”). The Shapley Values for this instance is given by the Figure below. The protocol used, ICMP, and the number of data bytes from the source to the destination, which is strong, have the strongest contribution to the high score. These elements are characteristics of a smurf attack, which consists in a distributed

denial-of-service attack in which large numbers of ICMP packets with the intended victim's spoofed source IP are broadcast to a computer network using an IP broadcast address.

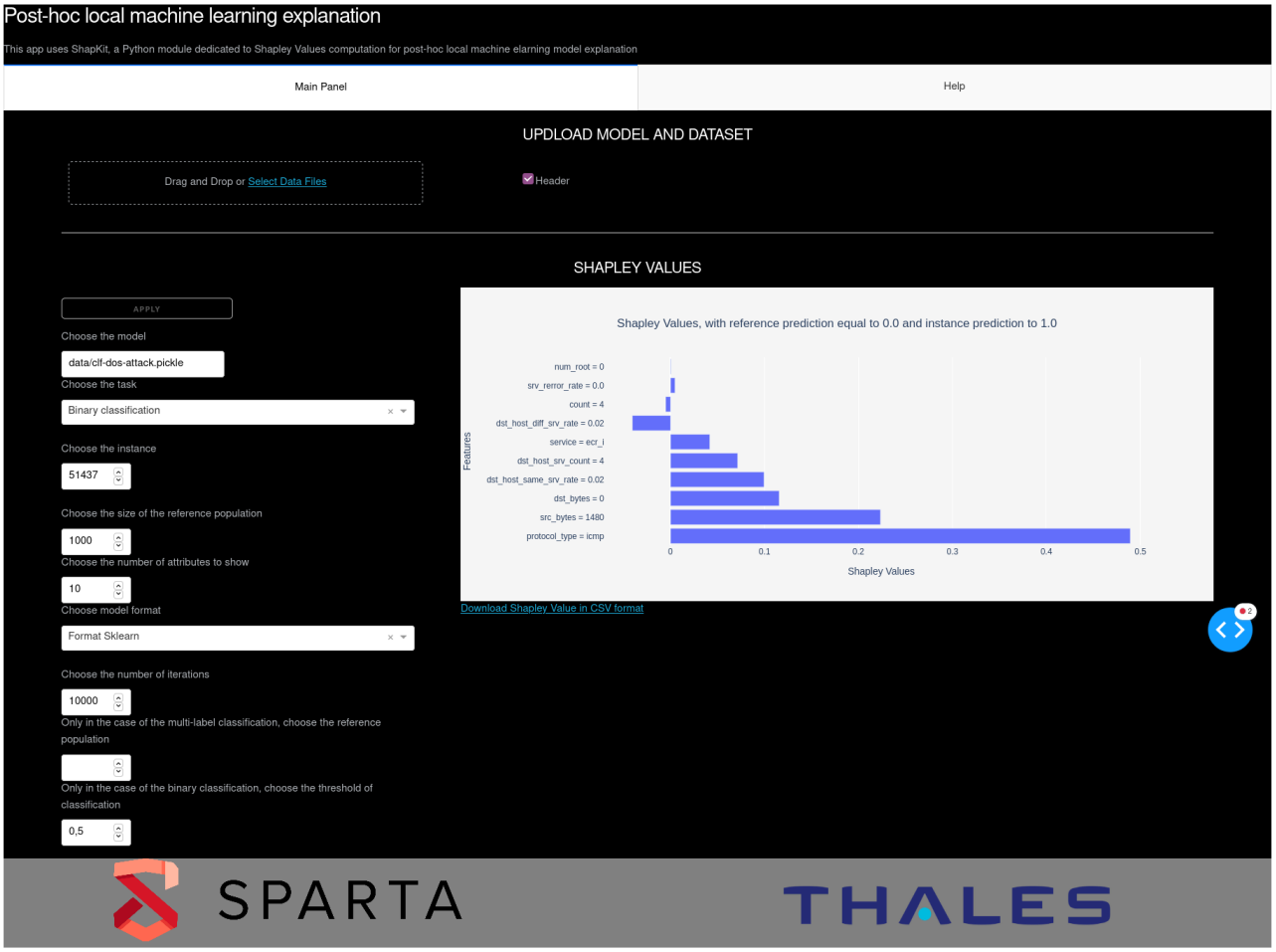



Figure 30: Shapley Values for a first instance predicted as a DoS attack by a ML model learnt with catboost

The second instance is completely different of the first one: the protocol is different and the number of data bytes from the source is null. dst_host_serror_rate, dst_host_serror_rate and dst_host_diff_srv_rate represent the percentage of connections that have SYN errors from destination to host referring to the same-service connections, the percentage of connections that have SYN errors from destination to host referring to the same-host connections and the percentage of connections to different hosts referring to the same-service connections. The two first values for the instance are equal to one and the last one is close to 0. As for previous instance, we select as references sub-population 1000 random instances that has been predicted as normal by the ML model. The Shapley Values for this second instance is given by the Figure below. A SYN flood is a form of DoS attack in which an attacker rapidly initiates a connection to a server without finalizing the connection. The server has to spend resources waiting for half-opened connections, which can consume enough resources to make the system unresponsive to legitimate traffic. When a client attempts to start a TCP connection to a server, the client and server exchange a series of messages which normally runs like this:

- The client requests a connection by sending a SYN (synchronize) message to the server;
- The server acknowledges this request by sending SYN-ACK back to the client;
- The client responds with an ACK, and the connection is established.

This is called the TCP three-way handshake, and is the foundation for every connection established using the TCP protocol. A SYN flood attack works by not responding to the server with the expected

ACK code. The malicious client can either simply not send the expected ACK, or by spoofing the source IP address in the SYN, cause the server to send the SYN-ACK to a falsified IP address, which will not send an ACK because it knows that it never sent a SYN. According to the Figure below, several elements contribute to increase the score predicted by the model compare to the reference population. One of them is that all the connections have SYN errors from destination to host referring to the same-service connection. Moreover there are no data bytes transmitted from the destination to the source (dst_bytes = 0) and from the source to the destination (scr_bytes = 0). These elements contribute the increase of the score too. At last, the percentage of connections to different hosts referring to the same-service connections is close to zero and a strong number of connections to the same host as the current connection in the past two seconds (94) contribute to increase the score too. All these elements are characteristics of a SYN Flood attacks.

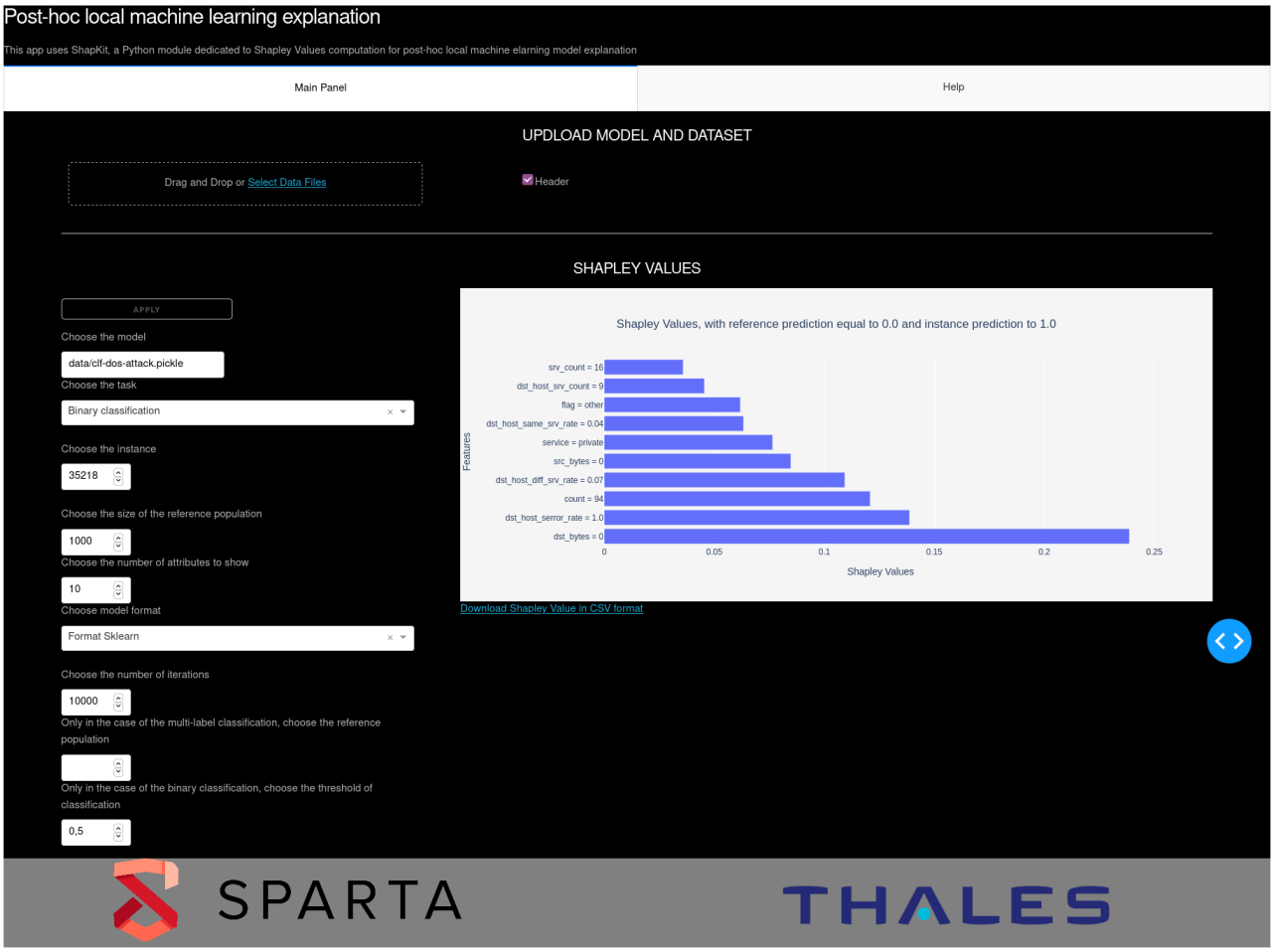



Figure 31: Shapley Values for a second instance predicted as a DoS attack by a ML model learnt with cat-boost

## 3.2 Surrogate type explanations in cybersecurity related environment

Over the past few months, further advancement of the solution described in D7.2 and D7.4 has been carried out. Additionally, a supplemental exploration of surrogate-type methods for explainable artificial intelligence (xAI) and their application in the context of cybersecurity has been conducted.

Surrogate-type methods denote techniques using a simpler, transparent algorithm to derive explanations for a complex, black-box method [28]. Particularly, methods explored in this work are local surrogates that train an interpretable model locally in the neighbourhood of the explained instance [29].

The improved Hybrid Oracle-Explainer approach [30] is described in section 3.2.1.

Furthermore, two ideas related to surrogate-type explanations were investigated. The first one is concerned with the effect that various data balancing methods can have on representatives of those techniques: Local Interpretable Model-agnostic Explanations (LIME) [31] and Hybrid Oracle-Explainer based on comprehensible decision trees [30]. To the best of our knowledge, this work is the first of its kind and opens the way for future exploration. The details and results are described in section 3.2.2.

The second idea investigates the proposition of surrogate-type approaches to xAI in Natural Language Processing (NLP). The scope of the approach is on fake news detection utilizing techniques from sentiment analysis. The model learns to investigate the text for patterns associated with either genuine or false information [32]. Surrogate methods such as LIME and ANCHOR [33] are employed to highlight patterns detected by a state-of-the-art BERT-based system. Results are presented in section 3.2.3.

### 3.2.1 *Further advancement of the explainable intrusion detection systems*

#### 3.2.1.1 Hybrid Oracle-Explainer approach based on comprehensible decision trees

This subsection serves as a brief overview of the work described in D7.2, D7.4, and in [30], to better compare with the final, expanded system's version presented later in this section. All the implementation details were already described in D7.4.

The system was based on three principles serving as a guideline for xAI solution in the intrusion detection system (IDS). They were as follows:

1. In the context of IDS, the accuracy and reliability of a system are the top priority.
2. One phenomenon can have more than one explanation [34].
3. The delivered explanation should be simple and help to develop trust [31].

The Hybrid-Oracle explainer, overviewed in Figure 32, uses the following procedure to generate explanations:

1. After obtaining a prediction from the Oracle Model, the sample in its original form and the Oracle output are forwarded to the Explainer module. The sample is then compared with the saved centroids of each cluster made during the Explainer training process, to find the 'n' closest in terms of the Euclidean distance.
2. Starting with the closest centroid, the decision tree trained on the related cluster is retrieved. If its prediction matches that of the classifier (Oracle), the search stops, and the local explainer is returned. Otherwise, the algorithm continues until it finds a supporting tree or runs out of centroids. In that case, the tree linked to the closest centroid is returned.
3. Finally, a visualisation of the decision tree is created, together with the highlighted path leading to the prediction made by the chosen explainer. The generation of clusters, centroids, trees, explanation visualisations, and example model are described in detail in D7.4 in sections 4.3.2.2, 4.3.2.3, and 4.3.2.4.

The web application prototype based on Rest API was created to make it possible to employ this solution in real life. The backend is written in Python.

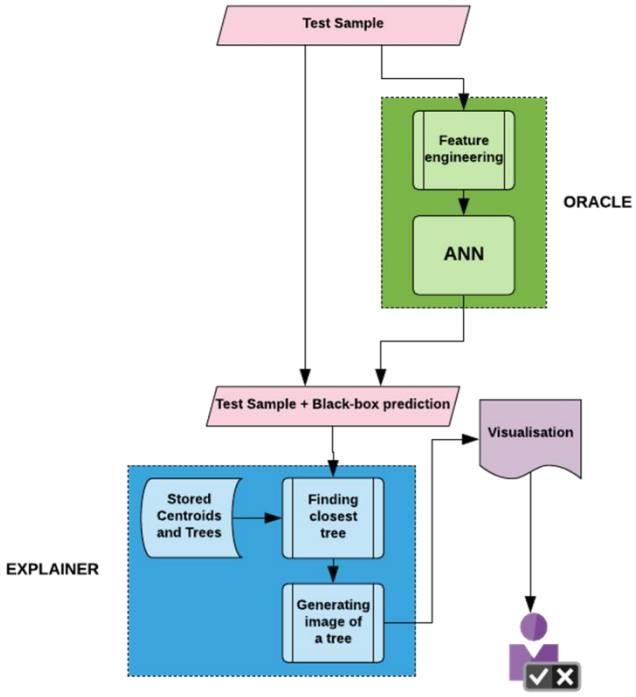



Figure 32: Overview of the explanation generation process.

React JS was selected to create the user interface, being one of the leading front-end modern technologies focused on the rapid development and the reusability of the components. Further description of the used technologies for the current version is in 3.2.1.2.

The previous iteration of the web application interface (as presented in D7.4) is visible in Figure 33. This hub had a table containing samples with predictions highlighted in red. The user could request an explanation of a given decision with a specific button located on the right side of a sample classification. Depending on the configuration, it would generate a visualisation similar to the one present in Figure 34.

The visualisation in Figure 34 presents a decision tree, where nodes show the attribute names together with the threshold value. Paths above the node represent the situation when the compared value is smaller than the threshold value, while paths below depict the opposite. At the last level, pie charts are used to depict leaves and their purity. The more homogenous a leaf is, t he better the quality of an explanation. Each consequent node represents a depth level of the decision tree. Orange path marks out the prediction path for the given sample's prediction. Finally, the orange table at the end illustrates values of the essential attributes present in the sample.

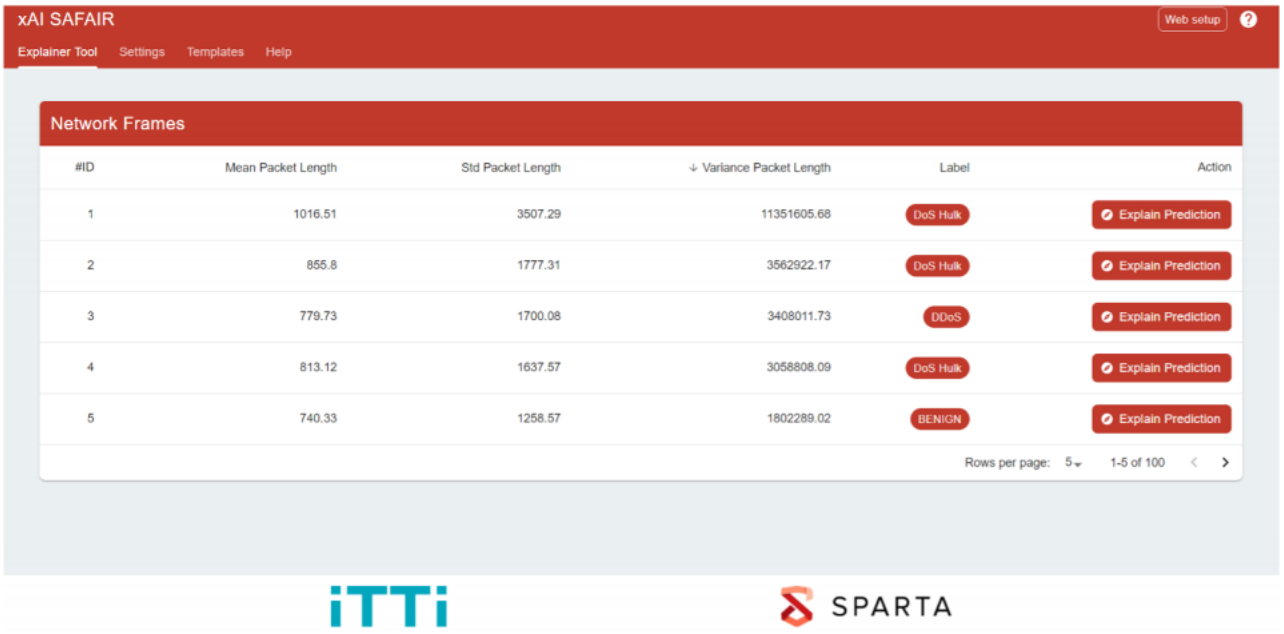



Figure 33: Previous version of the user interface.

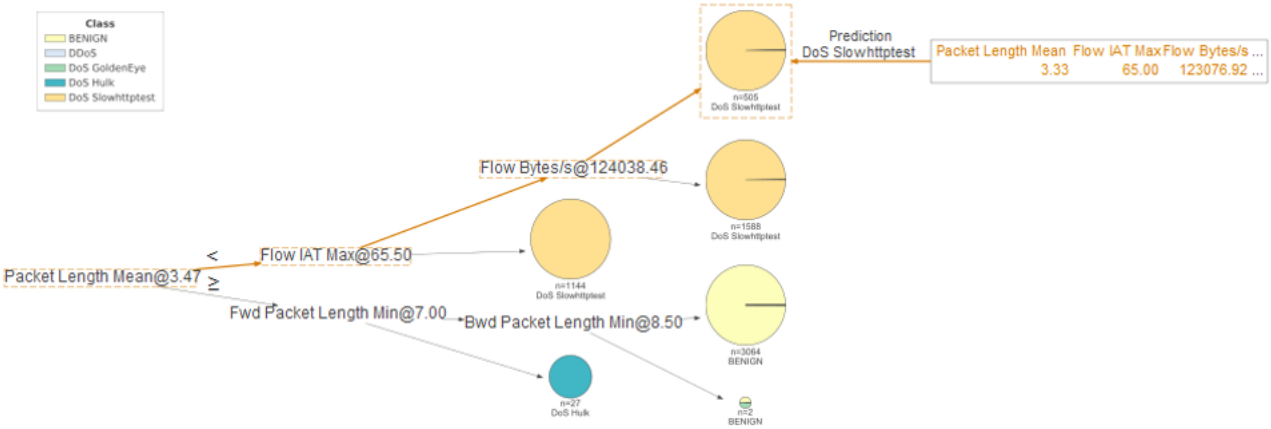



Figure 34: Example of the Oracle-Explainer prediction.

### 3.2.1.2 Technology

Main technologies used for the development are disclosed in the table below.

Table 12: Technology used in the solution development.

| Module/Langage | Version |
|---|---|
| Python | 3.7.8 |
| matplotlib | 3.3.3 |
| seaborn | 0.11.1 |
| plotly.js | 2.0.0 |

| Module/Langage | Version |
|---|---|
| tensorflow-cpu | 2.4.0 |
| pandas | 1.2.1 |
| scikit-learn | 0.24.1 |
| numpy | 1.19.5 |
| dask | 2021.1.0 |
| dill | 0.3.3 |
| Flask | 1.1.2 |
| Flask-Cors | 3.0.10 |
| confluent-kafka | 1.6.1 |
| elasticsearch | 7.12.0 |
| lime | 0.2.0.1 |
| React JS | 10.0.0.0 |

Most of the selected technologies are the same as those presented in D7.4, subsection 4.3.2.1, with only a few notable exceptions.

Dtreeviz [35], while still used for the decision trees' visualisation, is now directly imported as part of the project. It was modified to fix the found errors present in the module and to allow for further optimization and adaptations towards this project's end.

Plotly.js [36] is a high-level, declarative standalone JavaScript data visualisation library that can be used to create various chart types. Created charts are often interactive and enable fluid customisation.

### 3.2.1.3 Improvements and advancements

Since the description in D7.4, the system has been improved and advanced in several ways. In this subsection, the focus will be on the improvements added to the tool's capabilities. Explainability methods and their implementation are described in 3.2.1.4, while presentation of the new system is shown in detail in 3.2.1.5.

Following the results of the studies presented in sections 3.2.2 and 3.2.3, as well as the recommended practice for surrogate-type explanations, an additional technique was added to the system. Local Interpretable Model-agnostic Explanations (LIME) [31] is a model-agnostic method that is locally faithful, i.e., '*it corresponds to how the model behaves in the vicinity of the instance being predicted*' [31]. LIME samples instances around the prediction being explained and perturbates them. Then, it uses them to train an inherently interpretable linear model. The principle behind this is that any complex model is linear at the local scale and should provide an adequate local approximation. The output of LIME is '*a list of explanations, reflecting the contribution of each feature to the prediction of a data sample*' [37].

An instance of the explanation for the tabular data is shown in Figure 35. It shows the '*n'* specified features and rules such as '*duration is less or equal to -0.02*'. The chart visualises their impact and

allows to estimate respective weight. For example, '*duration is less or equal to -0.02*' can be interpreted as having tremendously positive influence on assigning the sample in question to the '*Benign*' class. Contrary, '*resp_pkts less or equal -0.05*' has negative weight and is an argument against it. This form of visualisation is further improved in the tool, which is shown in 3.2.1.5.

For the explainer based on comprehensible-decision trees [30], in addition to improvements in visualisation library a textual explanation extracted from the decision tree rules were added.

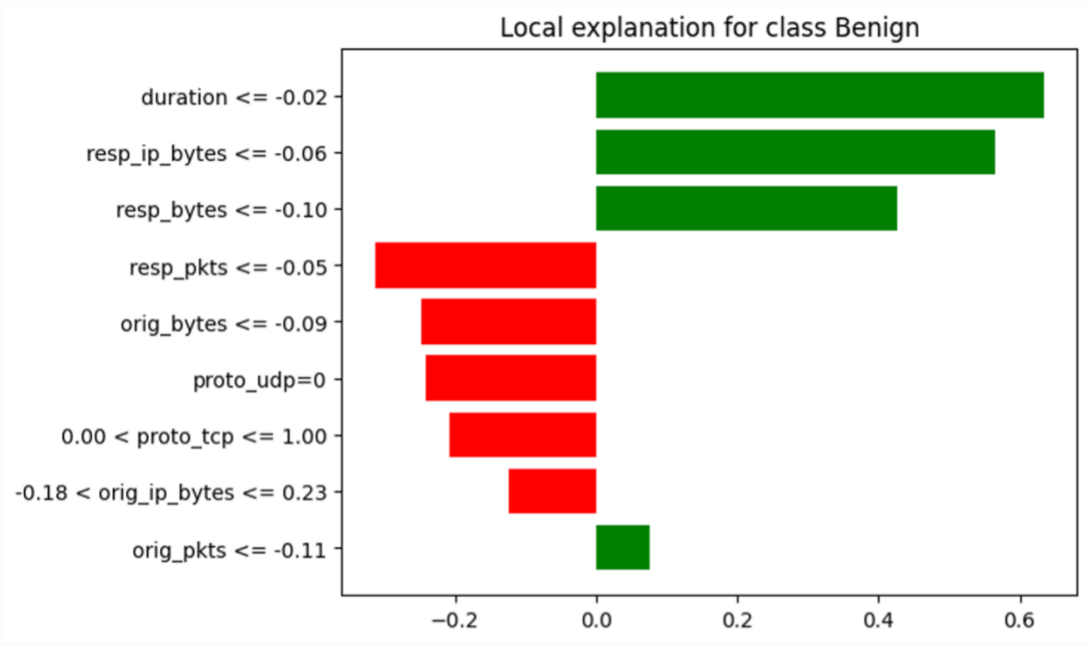



Figure 35: Instance of the LIME explanation.

### 3.2.1.4 Implementation of the explainers

Within the system, LIME is implemented as the '*LimeExplainer*' class, the code of which is presented in Listing 1. There is also supporting class '*LimeConfig*' necessary for the correct configuration of the method shown in Listing 2. Class *'LimeExplainer*' is responsible for creating and training the LIME explainer for the tabular data. Therefore, using methods from the '*lime'* module, the explainer is first created, trained, and then saved in the function '*create_lime_model*'. '*LimeConfig*' provides critical paths to store and load explainer, as well as feature names for later use. Prepared LIME explainer can be called to explain any sample, using the function '*explain_sample*', which returns a dictionary of features with their weights. It is used in the frontend to create a visualisation with the Plotly.js.

```
class LimeExplainer:

    def __init__(self, lime_config: LimeConfig):
        self.lime_config = lime_config
        self.lime_explainer = None

    def create_lime_model(self, train_dataset,
                          feature_names,
                          class_names,
                          categorical_features=None,
                          categorical_names=None):
        self.lime_explainer =
lime.lime_tabular.LimeTabularExplainer(train_dataset,

feature_names=feature_names,

class_names=class_names,

categorical_features=categorical_features,

categorical_names=categorical_names)
        with open(self.lime_config.lime_model_path + '.data', 'wb') as
file:
            dill.dump(self.lime_explainer, file)

    def load_lime_explainer(self):
        with open(self.lime_config.lime_model_path, 'rb') as file:
            self.lime_explainer = dill.load(file)
        return self


    def explain_sample(self, sample, model):
        result = self.lime_explainer.explain_instance(sample,
                                                      model.predict,
               num_features=len(self.lime_config.feature_names)).as_list()
        dict = {'x': [], 'y': []}
        for y,x in result:
            dict.get('x').append(x)
            dict.get('y').append(y)
        return dict
```

Listing 1: LIME explainer.

```
class LimeConfig:

    def __init__(self,
                 lime_model_path: str = None,
                 feature_names=None,
                 is_enable: bool = False,
                 ):
        if feature_names is None:
            feature_names = []

        self.is_enable = is_enable
        self.feature_names = feature_names
        self.lime_model_path = lime_model_path
```

Listing 2: ‘LimeConfig’ class code.

The explainer based on the comprehensible decision trees, while at the core similar to what was listed in D7.4, 4.3.2.2 and 4.3.2.3, was redesigned to fit the improved solution structure better. It is now split between three classes: '*TreeExplainerConfig*' shown in Listing 3, '*TreeExplainerGenerator*' shown in Listing 4, and '*TreeExplainerLoader*' in Listing 5.

```
class TreeExplainerConfig:

    def __init__(self,
                 is_enable: bool=False,
                 clusters_path: str = '',
                 centroids_path: str = '',
                 trees_path: str = '',
                 representativity=None,
                 depths=None
                 ):
        if representativity is None:
            representativity = [0.2]
        if depths is None:
            depths = [3]
        self.is_enable = is_enable
        self.clusters_path = clusters_path
        self.centroids_path = centroids_path
        self.trees_path = trees_path
        self.representativity = representativity
        self.depths = depths
```

Listing 3: 'TreeExplainerConfig' class code

The '*TreeExplainerConfig*' class serves the role of a configuration object passed to the explainer. It contains all the necessary values for its proper functioning, such as hyperparameters and the system paths.

The '*TreeExplainerGenerator*' has all the mechanisms responsible for the preparation of comprehensible decision trees. For that purpose, in the function '*generate*', it calls private methods '*_-generate_clusters_and_centroid'* and '*__generate_trees'*. The generated clusters, centroids, and trees are saved. This class utilizes the configuration object and needs appropriate training data to be used.

Finally, the '*TreeExplainerLoader*' is responsible for the actual provision of the explanation. It employs the saved output of the '*TreeExplainerGenerator*', functions of the dtreeviz library, and extends the methods presented in [28] to deliver both textual and visual explanation. These are sent to the frontend and presented to the user as explanations of the requested sample.

```
class TreeExplainerGenerator:

    def __init__(self,
                 train_set_x,
                 train_set_y,
                 representativity=None,
                 depths=None):
        if representativity is None:
            representativity = [0.2]
        if depths is None:
            depths = [3]

        self.train_set_x = train_set_x
        self.train_set_y = train_set_y
        self.depths = depths
        self.representativity = representativity

    def generate(self,
                 tree_explainer_config: TreeExplainerConfig
                 ):
        clusters, centroids = self.__generate_clusters_and_centroid(
                                representativity=self.representativity)
        trees = self.__generate_trees(clusters, self.depths)

        for centroid in centroids:
            pickle.dump(centroid, open(
                        tree_explainer_config.centroids_path, "wb"))

        for cluster in clusters:
            pickle.dump(cluster, open(
                        tree_explainer_config.clusters_path, "wb"))

        for tree in trees:
            pickle.dump(tree, open(
                                tree_explainer_config.trees_path, "wb"))

    def __generate_clusters_and_centroid(self, representativity=None):
        if representativity is None:
            representativity = [0.2]

        clusters = []
        centroids_of_clusters = []
        K = [int(len(self.train_set_x) * r) for r in representativity]
        for k in K:
            clustering, centroids = mdav(self.train_set_x,
                                         self.train_set_y, k)
            clusters.append(clustering)
            centroids_of_clusters.append(centroids)

        return clusters, centroids_of_clusters

    def __generate_trees(self, clusters, tree_depths=None):
        if tree_depths is None:
            tree_depths = [3]

        tree_explanations = []
        for tree_depth in tree_depths:
            explanations = gen_explanations(clusters[0], tree_depth)
            tree_explanations.append(explanations)

        return tree_explanations
```

Listing 4: ‘TreeExplainerGenerator’ class code.

```
eeExplainerLoader:
__init__(self,
         cluster_path,
         centroid_path,
         trees_path,
         target_name,
         feature_names,
         class_names: list
         ):
self.clusters = pickle.load(open(cluster_path, 'rb'))
self.centroids_centers = pickle.load(open(centroid_path, 'rb'))
self.trees = pickle.load(open(trees_path, 'rb'))
self.target_name = target_name
self.feature_names = feature_names
self.class_names =
{index_class: class_name for index_class, class_name in enumerate(class_names)}

generate_visualization(self, sample, prediction):
explanation_ext_prediction, ret_exp, \
ret_cen, ret_cluster_number = self.__find_best_tree(sample, prediction)

viz = dtreeviz(ret_exp[0],
               self.clusters[ret_cluster_number[0]][0],
               self.clusters[ret_cluster_number[0]][1],
               target_name=self.target_name,
               orientation='LR',
               fancy=False,
               feature_names=self.feature_names.tolist(),
               class_names=self.class_names,
               X=sample)

return viz.svg()

generate_text_explanation(self, sample, prediction):
explanation_ext_prediction, ret_exp, \
ret_cen, ret_cluster_numbers = self.__find_best_tree(sample, prediction)

return explain_prediction_path(ret_exp[0],
                               sample,
                               explanation_type='plain_english',
                               feature_names=self.feature_names,
                               target_name=self.target_name,
                               class_names=self.class_names),
                               ret_exp[0].predict(sample.reshape((1, -1)))

__find_best_tree(self, predicted_sample, prediction):
explanation_ext_prediction = []

p, q, ret_exp, ret_cen, ret_cluster_number = pre_explanations_ext(self.trees,
                                                    self.centroids_centers,
                                                    predicted_sample.reshape(1, -1),
                                                    [prediction],3)
explanation_ext_prediction.append(q)

return explanation_ext_prediction, ret_exp, ret_cen, ret_cluster_number
```

Listing 5: ‘TreeExplainerLoader’ class code.

### 3.2.1.5 System presentation

This subsection focuses on the presentation of advancements described in 3.2.1.3. The frontend of the IDS is showcased in Figure 36, Figure 37, Figure 38, Figure 39, Figure 40, Figure 41.

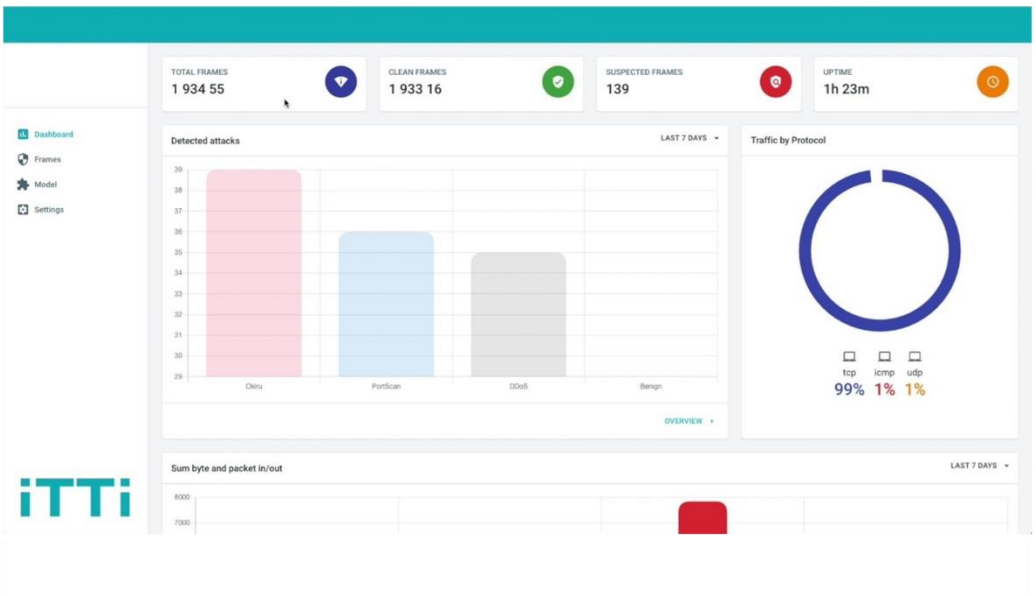



Figure 36: Application hub – charts and summary I.

Dataset summary depicted with panels at the top and supplemented with the charts allows for an instant overview of the data. Visualisation can be adjusted for the demands of the operator. For example, in Figure 36, panels show the total number of frames and how many of them are either '*clean*' or '*suspicious*'.

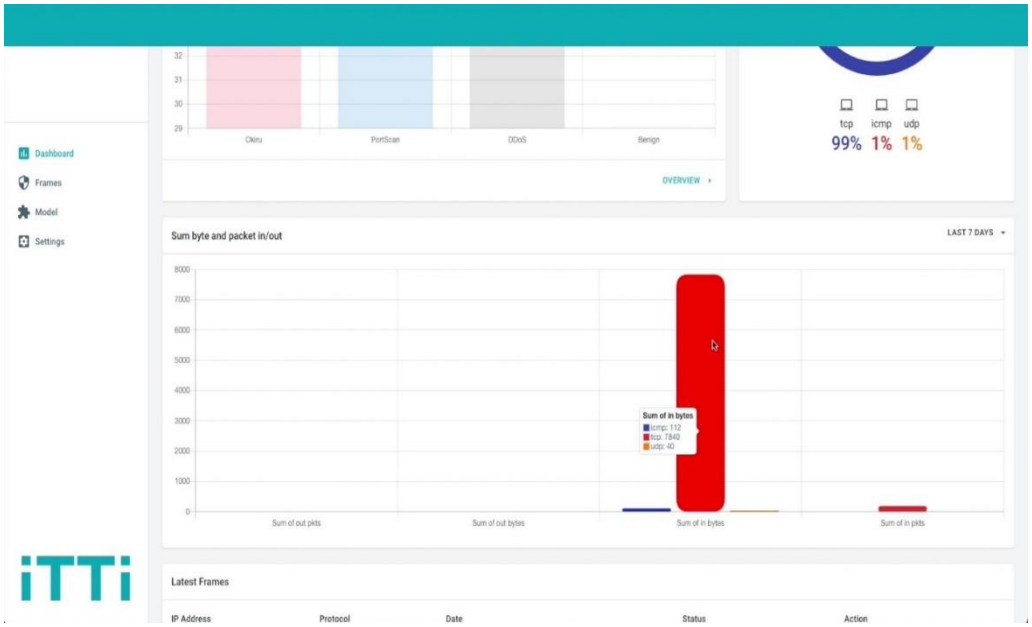



Figure 37: Application hub – charts and summary II.

Furthermore, charts show types of detected attacks, information about protocol type's prevalence, and in Figure 37, a summary of packets and bytes. Finally, in Figure 38, there is a table with the five latest frames, annotated with basic descriptors such as IP address, protocol, date, and classification. The '*View*' buttons allow for more detailed inspection.

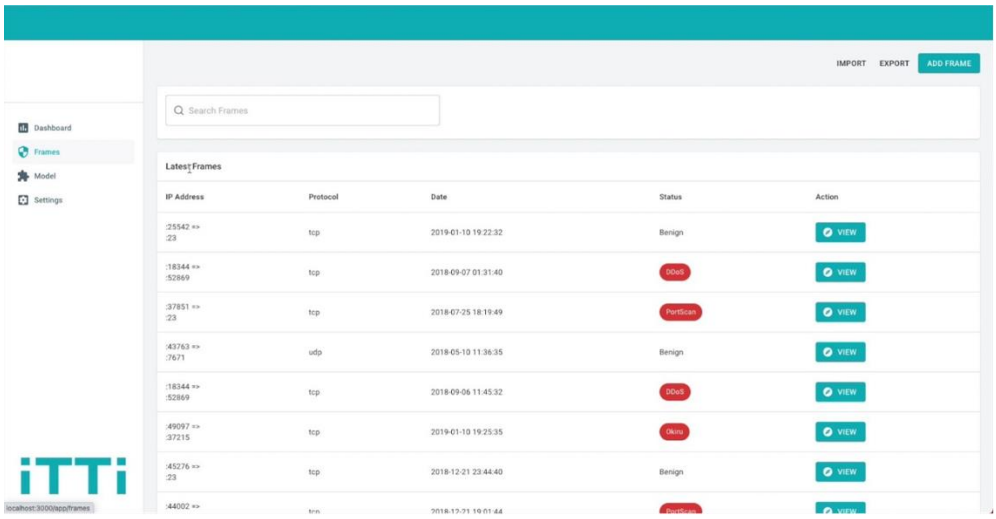



Figure 38: 'Frames' view.

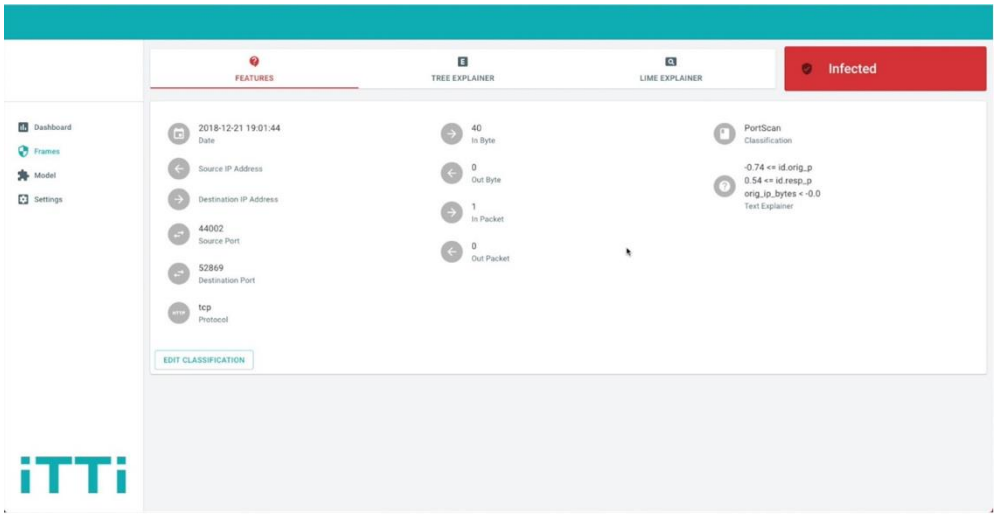



Figure 39: Frame panel – detailed view.

In Figure 39 a panel with a detailed description of a frame can be found. The textual explanation coming from the Hybrid Oracle-Explainer is present. An instance of explainer output is displayed in Figure 40, while Figure 41 depicts LIME explanation.

Hybrid Oracle-Explainer output present in Figure 40 is similar to what was shown in Figure 34. The main difference is that it is now supplemented with classifications from both the model and the explainer. Moreover, the extracted decision tree's rules are shown as the auxiliary explanation for better clarity.

The LIME explanation in Figure 41 is distinct from the one in Figure 35. Weights and attributes have an interactive plot created with Plotly.js. A supplementary box chart is also provided.

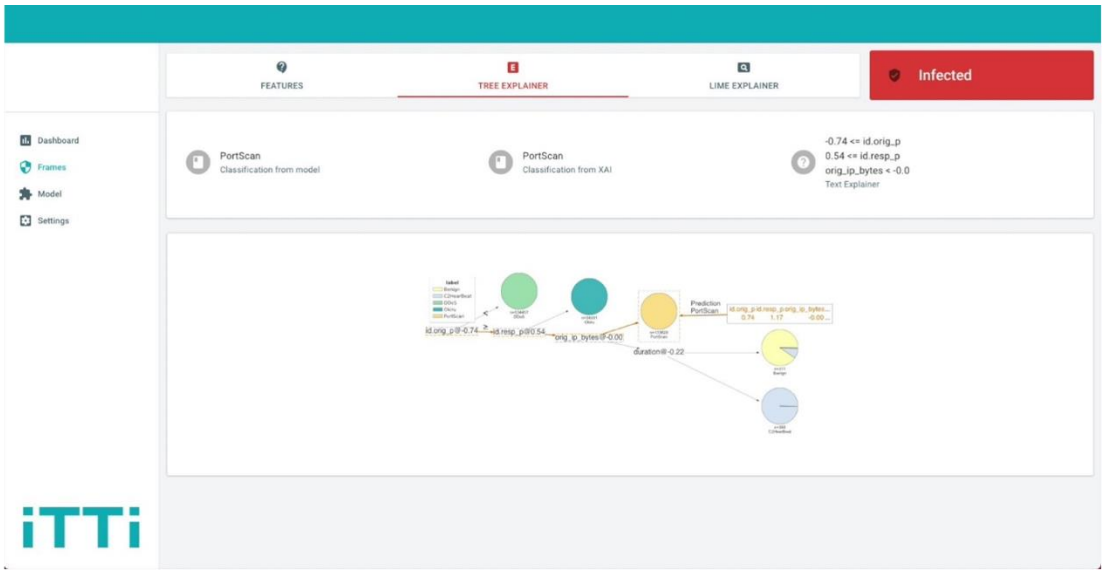



Figure 40: Frame panel – Tree's explainer output.

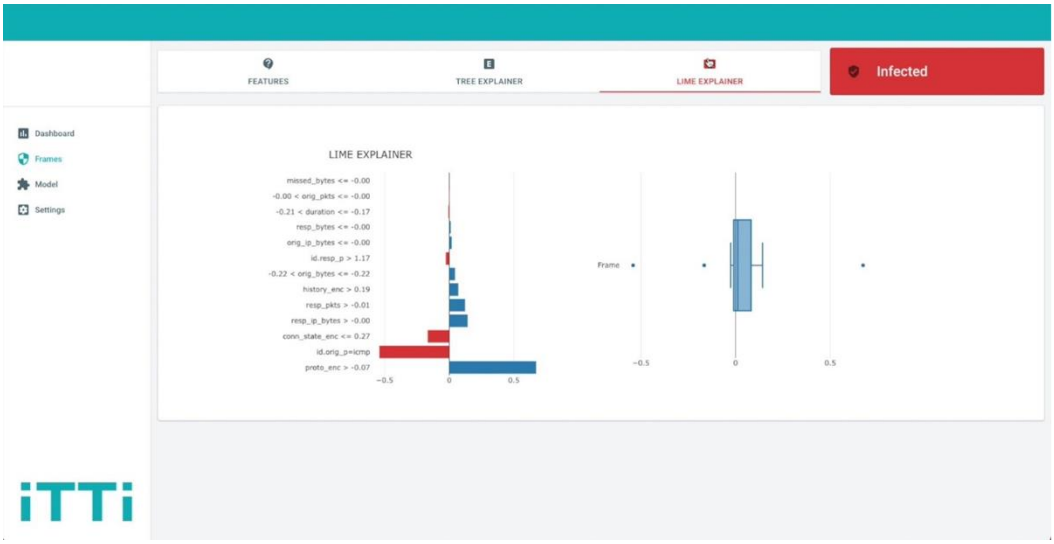



Figure 41: Frame panel – LIME explainer output.

### 3.2.2 *The effects of data balancing procedures on surrogate explainability methods in network cybersecurity-related streamed difficult data*

#### 3.2.2.1 Context and rationale

The negative influence of the imbalanced data factor on the various models' performance is well known [38]. There are also strategies to mitigate it. The under-sampling technique aims to reduce the number of samples from the majority class to balance the data distribution [39]. Over-sampling is another popular approach that inflates the number of minority class samples. One of the most proliferated algorithms from this class is SMOTE [40], where synthetic data points are introduced along the line segments joining the '*k*' minority class nearest neighbours.

The need for explanations of the black-box AI models rises with applications in certain domains. Because of their black-box nature, AI techniques can be subject to distrust while being prone to obscure errors and biases [41]. Thus, a comprehensive suite of explanatory techniques was developed to improve the transparency of such systems.

However, although both of those issues separately are commonly acknowledged, there were no attempts to study their interaction, that is, to establish a relation between the model trained on imbalanced data, data balancing techniques, and impact of those on surrogate-type explanations.

This work explores the impact of data balancing on the usage of surrogate-type explainability techniques in network intrusion detection systems (NIDS).

#### 3.2.2.2 Data

During the experiment, samples from the recent Internet of Things (IoT23) cybersecurity dataset [42] were used. This dataset contains network traffic from IoT devices, having 20 scenarios with malware capture and three with benign traffic. All samples were obtained from real hardware in a controlled network environment.

While the full dataset features 16 classes, for this experiment, only two were chosen. Additionally, after thorough feature selection, from the original 21 features, the eight presented in Table 13 were used.

Table 13: Features used in the tests.

| Feature | Description |
|---|---|
| proto | Type of communication protocol |
| duration | Duration of package transition |
| orig_pkts | Number of packets the originator sent |
| orig_ip_bytes | Number of IP level bytes the originator sent |
| resp_pkts | Number of packets the responder sent |
| resp_bytes | The number of payload bytes the responder sent |
| orig_bytes | The number of payload bytes the originator sent |
| resp_ip_bytes | Number of IP level bytes the responder sent |

Any '-' values were turned to '0' and the rows with '*NaN*' were deleted. The samples of '*Benign*' traffic were taken from the concatenation of scenarios 1, 7, 8, 34, 35, 36, resulting in 469 395 data points, while the samples of '*C&C*' attack were taken from scenario 43 count 3 490 data points. Also, one randomly selected sample from each category was cut to observe how the behaviour of explanatory algorithms changes between tests.

#### 3.2.2.3 Methodology

This research focused on two selected explainability methods: LIME [4] and the Oracle-Explainer [30]. The employed process is as follows; First, the general model performance is investigated to see how different methods had affected its accuracy, and how the two xAI test samples used for explanation generation were classified. Then, the output of the LIME method for the two samples between the trials is evaluated. It is done to find out to what degree the importance of the features changes. The Oracle-Explainer-generated explanations undergo a similar examination in the following subsection.

Four tests were conducted using data as described in 3.2.2.2. The research scenarios employed in this work were as follows:

1. **Imbalanced** – data used to train the model is not balanced, making it the '*Default*' scenario,
2. **Undersampled** – data is undersampled using Random Subsampling,
3. **Oversampled** – data is oversampled using SMOTE [40],
4. **Over- and Undersampled** – data is first oversampled and then undersampled with SMOTEEN [43] from imblearn.

Each test followed an identical pipeline. Samples were first shuffled randomly to improve the quality of clusters used by the decision trees in the Oracle-Explainer [30]. The feature columns, except for one-hot encoded column '*proto*', were scaled using '*StandardScaler*' afterwards. On the other hand, Label columns were encoded by '*LabelEncoder*' offered by scikit-learn [44].

Subsequently, the dataset was split into train and test set with the test size equal to 25% of all the data points. Then, in tests 2, 3, and 4, the selected balancing algorithm was utilised. Before training the Artificial Neural Network (ANN), the clusters, centroids, and decision trees necessary for the Oracle-Explainer were prepared. The details of this procedure are available in [30]. For this research, the representativity parameter is set to 0.2, while the maximal depth of decision trees is limited to 3 levels. It should also be noted that the version of Oracle-Explainer in this study is modified with a mechanism ensuring the retrieved decision tree has more than just one class in it. Also, wherever possible, the random seed was set to 0 to ensure reproducibility.

The ANN utilised in the experiment had two dense layers with 64 and 32 neurons each. Both of them used ReLU as the activation function, while the output layer employed Softmax. The Categorical cross-entropy was used as the loss function, while the ADAM with the learning rate equal to 0.1 was the optimizer. Model performance in each test was measured with the prepared test set.

The obtained model was used to generate predictions for two xAI test samples cut from the dataset beforehand, as mentioned earlier in this section. The samples were transformed with the identical encoders and scalers as the rest of the dataset. Finally, they were fed to the two selected explainability methods to generate interpretations of model predictions.

#### 3.2.2.4 Model performance

In Table 14, all the results of the models for every experiment were combined. Though the model trained with the imbalanced dataset achieved the highest accuracy, in a situation with significant discrepancies between samples, this can be a very misleading metric. By looking at other scores, it became clear that it classified nearly all samples as benign. The two xAI test samples were also classified incorrectly. High recall for the '*Benign*' class combined with very low recall for the '*C&C*' class suggests that the classifier could not distinguish between classes. A decrease in precision for the '*Benign*' class supports this assertion.

Balancing the dataset through undersampling or oversampling helped tackle the very low recall. Unfortunately, it was achieved at the price of accuracy and precision. The model trained on undersampled data recognised attacks very well, but it misclassified the benign samples. It should be noted that since there are only 874 '*C&C*' samples in the test set, even a few mistreated benign samples lead to a considerable drop in precision.

The SMOTEEN procedure used in the test case '*over- and undersampled*' had the lowest recall score of all the balancing methods. This, along with high execution time and a significant drop in precision, made it the worst-performing balancing method in this experiment.

Table 14: Model performance for differently Balanced Dataset.

| Dataset | Imbalanced | Undersampled | Oversampled | Over- and Under-sampled |
|---|---|---|---|---|
| Accuracy | 99% | 91% | 91% | 97% |
| Imbalance ratio train set | 1:134 | 1:1 | 1:1 | 1:2 |
| Imbalance ratio test set | 1:134 | 1:134 | 1:134 | 1:134 |
| Precision Benign | 99% | 100% | 100% | 100% |
| Precision C&C | 100% | 7% | 7% | 12% |
| Recall Benign | 100% | 91% | 91% | 97% |
| Recall C&C | 1% | 100% | 100% | 54% |
| Test Sample I Classification | Benign | C&C | C&C | Benign |
| Test Sample II Classification | Benign | C&C | Benign | Benign |

### 3.2.2.5 LIME explanations comparison

Table 15 presents the three most essential features highlighted by LIME for both samples in each test scenario. The '*Feature*' column shows the name of the chosen feature and whether or not it was bigger than some discovered threshold value, while '*Score*' presents its impact on the final prediction. Positive value can be interpreted as an argument for the given prediction, while negative one as the opposite. Features are sorted by the absolute value of their score.

The analysis of the table clearly shows that the importance of the features changes between tests. For example, in the imbalanced dataset, the most critical sign that a class belongs to the '*Benign*' traffic was a duration lesser than -0.02. In contrast, for the undersampled dataset, it was '*resp_pkts*' lesser or equal to -0.05. The '*duration*' then became the most significant for the oversampled dataset, again falling behind in the last test case.

A similar dynamic holds for the second test sample. Here also, depending on the balancing approach used, the importance of the features changes. For the case with the unbalanced dataset, '*resp_ip_bytes'* bigger than -0.06 was the most significant feature, along with the duration greater than -0.02. However, after undersampling the dataset '*resp_ip_bytes*' lost to '*orig_bytes'*, duration even started to be treated as a negative indicator.

Table 15: LIME scores.

| | | Sample | | | |
|---|---|---|---|---|---|
| | | Benign | | C&C | |
| Dataset | Importance | Feature | Score | Feature | Score |
| Imbalanced | 1 | duration ≤ -0.02 | 0.04 | resp_ip_bytes > -0.06 | 0.05 |
| | 2 | resp_bytes ≤ -0.10 | 0.02 | duration > -0.02 | 0.04 |
| | 3 | orig_pkts ≤ -0.11 | 0.01 | orig_pkts ≤ -0.11 | -0.03 |
| Under-sampled | 1 | orig_bytes ≤ -0.09 | -0.33 | orig_bytes ≤ -0.09 | 0.42 |
| | 2 | resp_pkts ≤ -0.05 | 0.32 | resp_ip_bytes > -0.06 | 0.30 |
| | 3 | resp_ip_bytes ≤ -0.06 | 0.25 | resp_pkts > -0.05 | 0.29 |
| Oversampled | 1 | duration ≤ -0.02 | 0.60 | resp_bytes ≤ -0.10 | -0.48 |
| | 2 | resp_bytes > -0.10 | 0.51 | resp_ip_bytes > -0.06 | 0.48 |
| | 3 | resp_ip_bytes ≤ -0.06 | 0.49 | resp_pkts > -0.05 | -0.36 |
| Over- and Un-dersampled | 1 | resp_bytes ≤ -0.10 | 0.62 | resp_bytes ≤ -0.10 | -0.51 |
| | 2 | duration ≤ -0.02 | 0.59 | resp_ip_bytes > -0.06 | 0.48 |
| | 3 | resp_ip_bytes ≤ -0.06 | 0.57 | resp_pkts > -0.05 | -0.26 |

#### 3.2.2.6 Oracle-Explainer explanations comparison

Table 16 presents the prediction paths obtained from Oracle-Explainer for the two test samples in each test scenario. It must be noted that the Oracle-Explainer finds the closest explanation to the label provided by the opaque model based on the feature vector and the label returned by the oracle. It is the reason why the same explanations are returned for samples assigned to the same category.

For the Oracle-Explainer, a pattern similar to the one noticed with LIME re-emerges. It means that depending on the sample distribution within the dataset, different explanations are generated. Specifically, decision tree splits are made using distinct features and values, leading to different prediction paths. The starkest evidence of this occurs when comparing a prediction path for the unbalanced dataset with the one made based on the undersampled data. The former utilises three distinct nodes, while the latter is based only on the feature '*orig_ip_bytes*' and whether or not its value is smaller than 0.02. This phenomenon only becomes more evident with further investigation of the gathered results.

Table 16: Prediction Paths of Oracle-Explainer.

| Sample | Imbalanced | Undersampled | Oversampled | Over- and Undersampled |
|---|---|---|---|---|
| Benign | duration < 0.27<br>proto_tcp ≥ 0.50<br>duration < 0.18 | orig_ip_bytes ≥ 0.02 | orig_ip_bytes ≥ 0.15<br>resp_ip_bytes < 0.84 | duration < 0.27<br>duration < 0.07 |
| C&C | duration < 0.27<br>proto_tcp ≥ 0.50<br>duration < 0.18 | orig_ip_bytes ≥ 0.02 | orig_ip_bytes ≥ 0.15<br>resp_ip_bytes ≥ 0.84 | duration < 0.27<br>duration < 0.07 |

The results of the experiment suggest that, depending on the dataset balance, the chosen surrogate-type methods can procure different explanations.

### 3.2.3 *Insights from the surrogate type explanation in a sentiment analysis based Fake News detection*

#### 3.2.3.1 Context and rationale

The worldwide inception of social media and their deep integration in the contemporary society has given people ways to interact, exchange information, form groups, or earn money, on a scale never seen before. The new possibilities paired with widespread popularity contribute to the level of impact they possess. Unfortunately, benefits brought by them come with a risk. They can be employed by various entities to spread fake news, either to make a profit or influence the population's behaviour, and can have a negative impact on society, posing a real danger that should not be underestimated. For instance, they can contribute to the rising distrust in children vaccination [44] or even lead to international tensions [45].

The term '*Fake News*' is not new, though it was popularised and politicised during the 2016 U.S. election, which has diluted its meaning [46]. Since then, it has become a buzzword, used in contexts deviating from its previous definition [47]. Initially, it meant an inaccurate piece of news, often fabricated on purpose, mimicking news media content [46]. Here, the term will denote purposefully fabricated pieces of information presented as legitimate, setting focus on the disinformative aspect of the phenomenon [47].

Social media play an essential role in the dissemination of fake news. As an example, data obtained during the 2016 U.S. election will be used here. Studies presented in [48] show that, on average, 41.8% of all the traffic to fake news outlets during that period was generated through social media. For genuine news sites average traffic share from this type of activity was equal only to 10.1% [48]. It is worth noting that this statistic does not show how many fake news headlines or '*tweets*' were just seen without clicking on the link; thus, it can be safely assumed that the exposition to fake news

and its presence in social media was higher than that. Generally, careful estimation is that, during the election period, every American adult encountered, on average, from 1 to 3 fake news articles [48].

Potential threats of fake news have raised concerns [47] and lead to the development of various countermeasures [44]. Fake news detection tools and methods can be distinguished into one of the two main categories: network-based approaches or linguistic-based approaches, with the existence of hybrid approaches using elements from both [32] [49].

Network-based approaches can estimate the truthfulness of news by assessing the veracity of the source. They utilise network properties such as, for example, authors, timestamps, or included links [32] [49]. Those tend to be used as complementary for linguistic-based approaches [32] [49].

Linguistic-based methods focus on the content of the investigated news [32] [49]. According to the idea that specific patterns exist for fake news [32] [49], they try to find anomalies in the text to verify its legitimacy. To illustrate, the unusually high frequency of some words may be a cue suggesting the abnormality of investigated text.

Lately, Bidirectional Encoder Representation from Transformers (BERT) [50]-based models have become prominent for linguistic-based techniques and natural language processing at large [51]. It applies the bidirectional training of a Transformer, an attention mechanism that learns contextual relations between words [51]. In contrast to the typical directional models, in the BERT architecture, Transformer's encoder reads the entire test sequence at once. It is the reason why it is considered bidirectional. It allows the model to learn to a better extent the context of a word based on its surroundings [51]. In consequence, this leads to the high performance of such models.

Since an approach where a BERT based model is trained on corresponding datasets to distinguish between real and fake news is recognized [52], it was an excellent opportunity to verify how surrogate-type methods will work in this context. There was not much work done in this regard, which makes these studies especially worthwhile. Additionally, because of the role that social media have in fake news dissemination, it was only appropriate to simulate their environment. That is the exchange of short messages and wide appearance of controversial titles encouraging users to engage.

#### 3.2.3.2 Data

The dataset used for this study is publicly available [53] and originally comes from [54]. Authors took then-genuine news articles from Reuters.com, while the fake ones were collected from another dataset on the portal kaggle.com.

The dataset is separated initially into two csv. files; one for the verified news with 21 417 samples and the other for the fake ones with 23 481 samples. Those separate files had to be merged and later reshuffled.

Four attributes describe each sample: the title, the text of the article, the subject of an article, and the date of publication. Since the purpose of this work was to simulate the content present on social media platforms such as Twitter, of the four attributes, only the *'title*' had been used. Additionally, a column with labels had to be created and added to the dataset manually. The dataset was split between the training portion with 80% of all samples, and the test portion with the rest of data points.

#### 3.2.3.3 Methodology

There were three major steps of the experiment. The first one was data preparation. The second one was the utilisation of the BERT-based classifier and training it to distinguish between real and fake news. Finally, two surrogate-type methods were employed to explain some of the predictions. Therefore, the general process followed these steps:

1. Data preparation,
2. Construction and training of the classifier,
3. Configuration and application of the selected xAI surrogate-type methods.

Before the actual training, BERT models need each input sentence to be transformed by a tokenizer. The tokenizer firstly breaks words into tokens. Then it adds unique [CLS] and [SEP] tokens at the beginning and the end of the sentence accordingly. Lastly, the tokenizer replaces each token with the corresponding ID. The ID comes from the pre-trained embedding table. The reasons behind this process and further details are available here [51].

The tokenizer and the pre-trained BERT model used in this study come from the '*transformers*' module in version 4.3.3. DistilBERT is the variant employed here. It is a lighter and faster version of the original BERT, which retains similar performance, developed by the team at HuggingFace [55]. The used model imposes tokenizer selection since those two must match and are pre-trained to work together. Additionally, the tokenizer was configured to either truncate or pad data to the '*max length*'. In this case, this parameter is set to 59, appropriately to the demands of short titles. Additionally, everything is converted to the lower case.

This project utilises transfer learning to create an effective model by leveraging the pre-trained BERT model and adapting it to the task. The only layers that are being optimised are those added to the distilBERT to perform classification. Those are one LSTM layer, one polling layer, one dense feed-forward layer with the ReLU activation function, and an output layer using softmax. The model is built using TensorFlow and Keras. Details of each trainable layer are collected in Table 17.

Table 17: Parameters of the network's trainable layers.

| Layer | Class | Units | Activation | Additional parameters |
|---|---|---|---|---|
| LSTM | tf.keras.layers.LSTM | 50 | activation=tanh<br>recurrent_activation=sigmoid | return_sequences=True<br>dropout=0.1<br>recurrent_dropout=0.1 |
| Poolling | tf.keras.layers.GlobalMaxPool1D | - | - | - |
| Dense | tf.keras.layers.Dense | 50 | ReLU | dropout=0.2 |
| Output | tf.keras.layers.Dense | 2 | Softmax | - |

The model uses '*Adam*' as the optimizer, while '*Sparse Categorical Crossentropy*' serves as the loss function. The utilised metric is '*Sparse Categorical Accuracy*'. Batch size is 100, while tests have proven that three epochs are enough for data used.

For explanation purposes, the test data was expanded with an additional column. It contains model predictions, and was added to inspect misclassified samples.

Two chosen surrogate-type methods are Anchors and LIME. LIME was described earlier in 3.2.1.3. Anchors is a model-agnostic explanation algorithm based on '*if-then*' rules, called '*anchors'* [33]. An '*anchor'* is a rule applied to the local prediction where *'changes to the rest of the feature values of the instance do not matter'* [33]. It means that the prediction is always supposed to be the same, for the instance on which the anchor holds [33]. As the authors of [33] highlight, anchors are intuitive, easy to comprehend, and have clear coverage.

A version of LIME designed to work with text was employed. It was configured to present the top five features and to use 5000 samples in the neighbourhood to train a local linear model.

Anchors needed the SpaCy [56] object in the textual explanation. Default trained pipeline package, '*en_core_web_sm*' has been used. Attribute '*threshold*' was set to 95%, 'temperature' to 0.3, '*beam_size*' to three and '*top_n*' to 1000. Examples shown were set to be perturbed by replacing words with UNKs.

Both algorithms needed auxiliary functions, which tokenize the text and return the model prediction. Explanations were derived on the test set expanded with the model's predictions.

Finally, to see how patterns impact the model's predictions, the following situations were investigated:

1. When the model correctly classified a title as fake news,
2. When the model correctly classified a title as real news,
3. When the model incorrectly classified a title as fake news,
4. When the model incorrectly classified a title as real news.

#### 3.2.3.4 Model's evaluation

The prepared model achieved accuracy on the level of 98%. Precision for real news is 97%, while for fake news it is equal to 99%. The recall has 99% for the real news and 97% for fake news. Finally, the F1-Score for both classes is 98%. Figure 42 presents the confusion matrix for this specific BERT-based model.

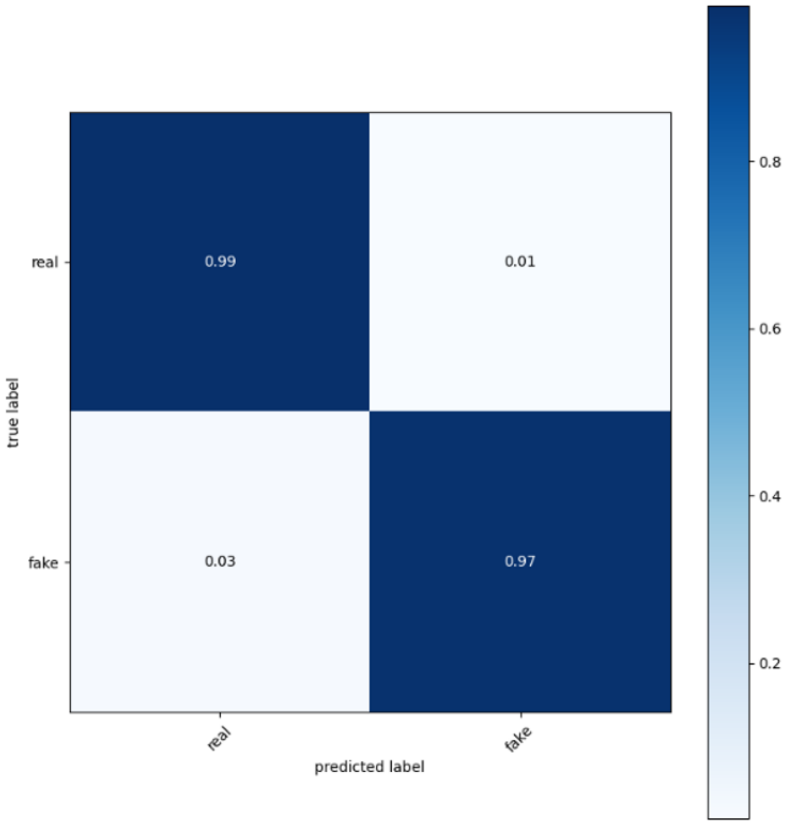



Figure 42: Confusion matrix for fake news detection.

### 3.2.3.5 Insights from surrogate-type explanations in fake news detection

The first test case was a situation where the model correctly classified a title as fake news. The picked instance for this case was '*FBI NEW YORK FIELD OFFICE Just Gave A Wake Up Call To Hillary Clinton'*. The visualization of the explanation offered by the LIME module is presented in Figure 43.

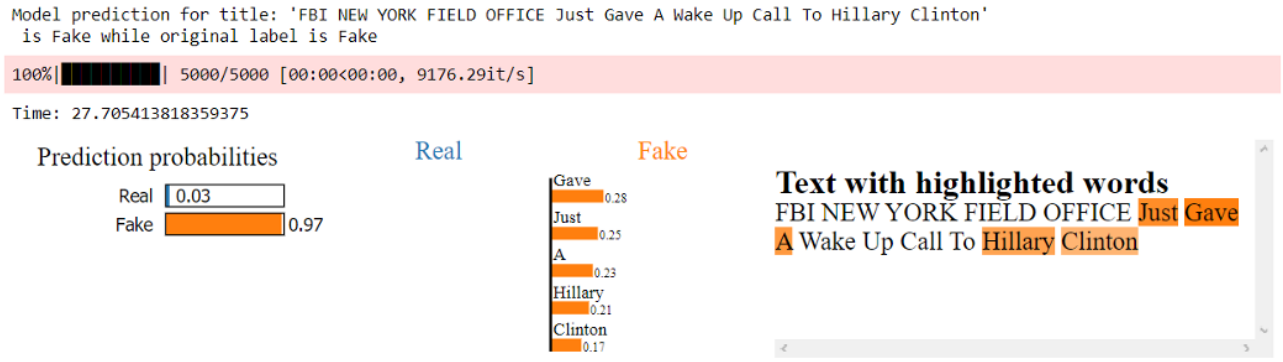



Figure 43: LIME explanation for the sentence '*FBI NEW YORK FIELD OFFICE Just Gave A Wake Up Call To Hillary Clinton*'.

In this example, the model predicts the falsehood of the title. In the middle of the figure, the weights of the top five most influential words are given. If the word '*Gave*' was to be removed, the probability of classification would drop by 0.28. For user convenience, the influential words are highlighted with a gradient representing their impact on the right side of the visualisation.

Anchors in this study were sometimes unable to find '*if-then*' rules. This was the case in this particular example and was very common in all titles classified as fake.

The second test case was when the model correctly classified a title as real news. The selected title was '*Turkey-backed rebels in Syria put IS jihadists through rehab*', and LIME explanation for it is presented in Figure 44.

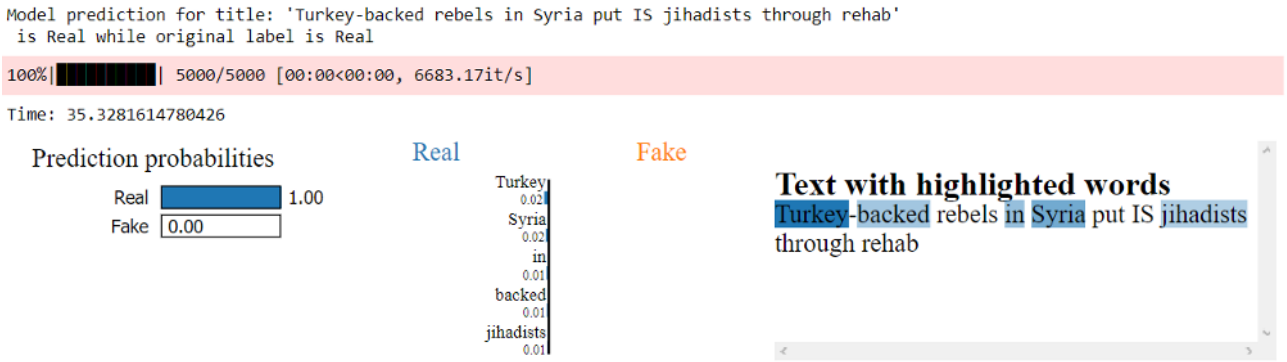



Figure 44: LIME explanation for the sentence '*Turkey-backed rebels in Syria put IS jihadists through rehab*'.

In this test, Anchors did successfully find the '*if-then*' rule. The output of the algorithm is in Figure 60.

```
Time: 73.39899110794067
Anchor: rehab AND Turkey AND through
Precision: 1.00

Examples where anchor applies and model predicts Real:
Turkey UNK UNK UNK UNK UNK UNK UNK UNK through rehab
Turkey - UNK UNK UNK UNK put IS jihadists through rehab
Turkey UNK backed rebels in Syria UNK UNK jihadists through rehab
Turkey - UNK UNK UNK UNK put IS jihadists through rehab
Turkey UNK backed UNK UNK UNK put IS jihadists through rehab
Turkey - backed UNK UNK Syria put IS jihadists through rehab
Turkey UNK backed rebels UNK Syria put UNK UNK through rehab
Turkey - backed rebels in Syria UNK UNK jihadists through rehab
Turkey UNK UNK UNK UNK Syria put IS jihadists through rehab
Turkey UNK backed UNK in UNK put UNK jihadists through rehab

Examples where anchor applies and model predicts Fake:
```

Figure 45: Anchors output for the sentence '*Turkey-backed rebels in Syria put IS jihadists through rehab*'.

It shows that the '*anchor*' is a combination of words/symbols '*rehab*', '*Turkey*' and '*through*'. Moreover, looking at the output, when these three appear together, the model's prediction is always '*real*'. '*Turkey*' overlaps with LIME explanations, which is a strong indicator of its importance.

The sentence representing the third case when the model misclassified real news as fake is '*Trump looms behind both Obama and Haley speeches*.', while the LIME explanation is in Figure 46. In this instance, the model has assigned relatively similar probabilities to both classes, with a relatively moderate advantage of 0.16 to the '*fake*' category. It seems this comes from the presence of names

'*Obama*' and '*Haley*'. Perhaps many pieces of fake news concern those persons, therefore they have a significant impact on the model's prediction.

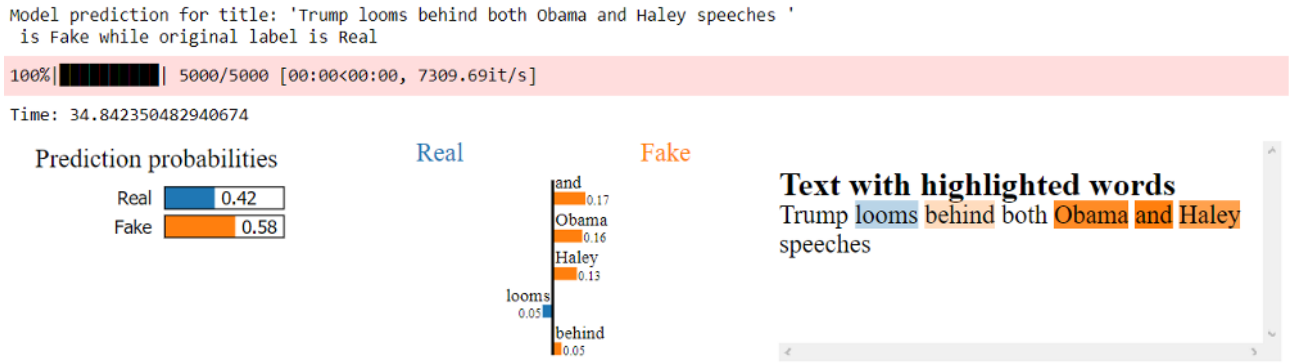



Figure 46: LIME explanation for the sentence '*Trump looms behind both Obama and Haley speeches*'.

It is reinforced by the output of anchors, present in Figure 47. It is worth noting that the anchor does not always give the prediction '*fake*' in this case. However, the name '*Obama*' is part of the '*anchor*', overlapping with LIME explanations. This suggests that names of characters or places that are often the subject of fake news can have a crucial impact on the model's decisions and could perhaps mislead it.

```
Time: 32.36743950843811
Anchor: and AND Obama
Precision: 0.96

Examples where anchor applies and model predicts Fake:
Trump looms UNK both Obama and Haley UNK
Trump looms behind UNK Obama and UNK speeches
Trump looms UNK both Obama and Haley UNK
UNK UNK UNK both Obama and Haley speeches
UNK looms behind both Obama and UNK UNK
Trump UNK behind UNK Obama and UNK speeches
Trump UNK UNK both Obama and Haley speeches
Trump UNK UNK both Obama and UNK UNK
Trump looms UNK both Obama and Haley UNK
Trump looms UNK both Obama and Haley UNK

Examples where anchor applies and model predicts Real:
Trump looms UNK UNK Obama and Haley speeches
Trump looms UNK UNK Obama and Haley speeches
Trump looms behind UNK Obama and Haley speeches
```

Figure 47: Anchors output for the sentence '*Trump looms behind both Obama and Haley speeches*'.

The instance where the model misclassified fake news as real is represented by the sentence '*Pope Francis Demands Christians Apologize For Marginalizing LGBT People*'. LIME's explanation for it is present in Figure 48, and it shows, that the model was confident to consider this title as '*real*'. Based

on this explanation, no strong patterns suggested that it was '*fake*' with marginal weights of words that could have changed it.

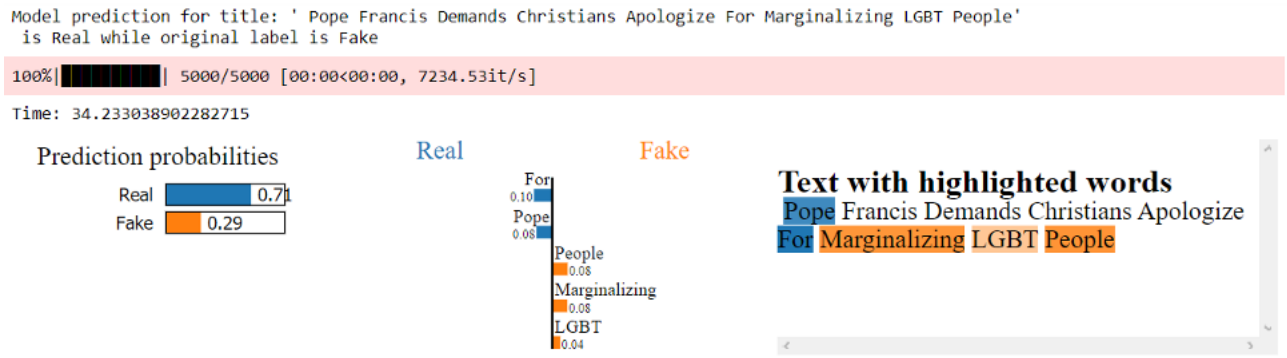



Figure 48: LIME explanation for the sentence 'Pope Francis Demands Christians Apologize For Marginalizing LGBT People'

Anchors' explanation for it is presented in Figure 49. An extensive '*anchor*' was necessary since it encompasses almost every word except for '*Marginalizing LGBT People*'. Those seem to have no impact, and according to this explanation, there were no possibilities to influence the outcome.

```
Time: 185.89239931106567
Anchor: Francis AND Pope AND   AND For AND Apologize AND Christians AND Demands
Precision: 1.00

Examples where anchor applies and model predicts Real:
  Pope Francis Demands Christians Apologize For Marginalizing UNK UNK
  Pope Francis Demands Christians Apologize For Marginalizing LGBT People
  Pope Francis Demands Christians Apologize For Marginalizing LGBT UNK
  Pope Francis Demands Christians Apologize For Marginalizing UNK People
  Pope Francis Demands Christians Apologize For Marginalizing UNK People
  Pope Francis Demands Christians Apologize For UNK UNK People
  Pope Francis Demands Christians Apologize For Marginalizing UNK People
  Pope Francis Demands Christians Apologize For UNK UNK People
  Pope Francis Demands Christians Apologize For Marginalizing UNK People
  Pope Francis Demands Christians Apologize For UNK UNK UNK

Examples where anchor applies and model predicts Fake:
```

Figure 49: Anchors output for the sentence 'Pope Francis Demands Christians Apologize For Marginalizing LGBT People'.

### 3.2.3.6 Conclusion

The achieved results suggest the following conclusions.

Firstly, it is possible to use surrogate-type methods to locally explain BERT-based models to some degree. Parts of the explanations and recovered patterns primarily focused on names, which made sense and agreed with the intuition. Those tend to appear in the output of both methods, and as the last case shows, through their usage, the model's decision can be impacted.

Secondly, the results confirm that more than one surrogate-type method should be used to derive explanations. As it can be seen, various methods in different configurations tend to highlight different patterns. Additionally, it would be recommended to support the surrogate-type methods with the algorithm from different classes. For instance, an attribution method such as SHAP could be valuable, since it is faithful to the model and can provide a global explanation.

## 3.3 Conclusion

The results of the finished work prove that surrogate-type explanations are a valuable tool which, under certain conditions, can effectively introduce explainability into the cybersecurity-related system. The presented final version of the Hybrid Oracle-Explainer tool in 3.2.1 is an actual example of that. This modern IDS solution can efficiently perform its designated role and provide insights into the data and its own decisions.

As for the conditions and potential weaknesses of those methods, there will always be an issue of fidelity. The study described in 3.2.2 depicts that the explanation can differ from method to method. Therefore, it is recommended to employ more than one of such techniques at once. Then it is possible to focus on the overlaps, which are the vital indicator of the potential attribute's importance. There is also still the matter of finding a way to estimate explanation’s faithfulness and the factors that affect its quality.

However, as the study presented in 3.2.3 has shown, surrogate-type explanations can be effectively employed in different domains, such as detecting and combating the dissemination of fake news in social media. Though there is still a field for improvement, their utilisation has already helped to understand potential vulnerabilities present in a state-of-the-art BERT model trained to make decisions only based on the semantic patterns in a text.

Ultimately, though those methods have their drawbacks, the benefits they offer outweigh the costs. They are flexible, model-independent, simple, easy to understand, and their proper usage allows to gain insights into the model. Therefore, they will probably remain a valid proposition in the foreseeable future.

# Chapter 4 Fairness ensuring mechanisms

In deliverable D7.4, we define some fairness metrics based on demographic parity, equality of opportunity, predictive rate parity, differential fairness and individual fairness. Moreover, we propose two algorithms dedicated to fair machine learning: one uses a fair adversarial network and the second one uses a fair version of random forest. These algorithms and metrics were used on the toy dataset adult-income, also known as the "Census Income" data set. The prediction task is to predict if a person makes over 50 k$ a year.

In this chapter, we describe a tool that is dedicated both to interpretability and to fairness inspection and presents its usage on the adult-income dataset. All the functions we use and the resulting plots in this part are implemented in *ethik*.

## 4.1 A model inspection tool to study fairness

### 4.1.1 *Short technical description*

In this chapter, we use *ethik*[6], a Python module dedicated to AI fairness and interpretabiltiy. This module uses some counterfactual distributions that permit answering some "what-if" scenario (e.g. what happens if the average of the age feature is equal to 55 instead of 40 in the dataset). The key principle is to stress one or more features of a test set and observe how the trained machine learning model reacts to the stress. The stress is based on a statistical re-weighting scheme called entropic variable projection. The main benefit of the approach is that it will only consider realistic scenarios, and will not build fake examples. Moreover, it computes an explanation with low algorithmic complexity, making it scalable to real-life large datasets. It uses an information theory framework that allows quantifying the influence of all inputs/outputs observations based on entropic projections. It proposes a model agnostic formalism enabling data scientists to interpret the dependence between the input features, their impact on the prediction errors and their influence on the output predictions. The authors of *ethik* proposes in the paper [85] the theoretical description behind the module.

### 4.1.2 *Application on a toy dataset*

#### 4.1.2.1 Toy dataset description

The data set[7] used is also known as the "Census Income" data set. The prediction task is to predict if a person makes over 50k$ a year. Thus the target variable is named "income" and worth either>50k$ or ≤50k$. The features used to predict the target are continuous or categorical. The Table below is a description of feature types and categories.

Table 18: Adult-income features

| Features | Description |
|---|---|
| **age** | continuous |
| **workclass** | Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked |

[6] Ethik AI (xai-aniti.github.io)
[7] UCI Machine Learning Repository: Adult Data Set

| Features | Description |
|---|---|
| **education** | Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool |
| **education-num** | continuous |
| **marital-status** | Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse |
| **occupation** | Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces |
| **relationship** | Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried |
| **race** | White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black |
| **sex** | Female, Male |
| **capital-gain** | Continuous |
| **capital-loss** | Continuous |
| **hours-per-week** | Continuous |
| **native-country** | United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad-Tobago, Peru, Hong, Holand-Netherlands |

In this chapter, we will focus on the fair adversarial network describes in deliverable 3.4 while protecting both the gender and the origin. We will use *ethik* to compare the impact of the feature gender on the outputs of the fair model with the impact on the outputs an unfair neural network whose architecture is the same as the classifier part of the adversarial network and with the impact of the gender on the true outputs. The two neural networks have similar performance on the test set.

#### 4.1.2.2 Impact of Fair Adversarial Network about the prediction made for the gender

With an optimal (according Kullback-Leibler divergence) reweighting of the observations, with *ethik,* we can study the evolution of the proportion of individual with an income greater than 50k$ when we stress the data distribution to force to increase or decrease the proportion of women in the dataset. Not surprisingly, the proportion of individuals earning more than 50k$ decreases as the proportion of women increases, illustrating the discrimination against women presents in these data. There would be no discrimination if the blue line was a horizontal line passing the orange point.

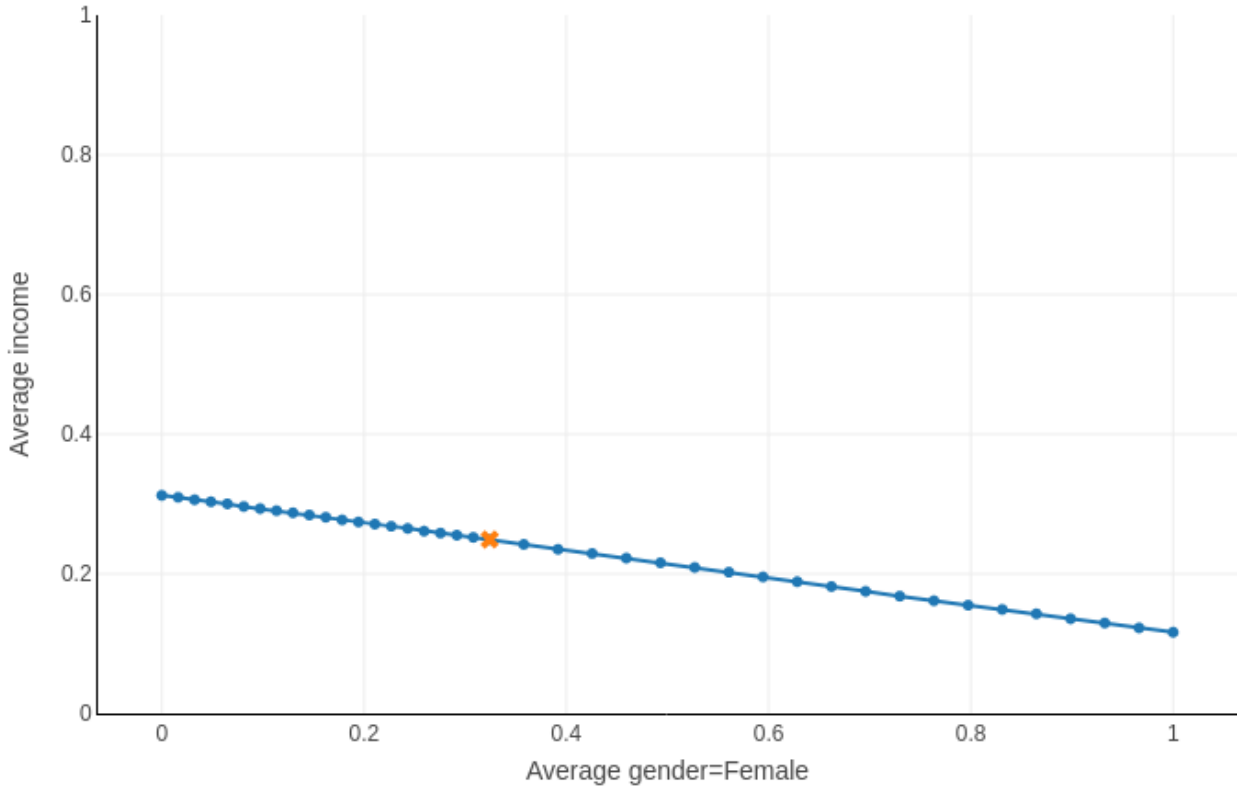



Figure 50: Proportion of instances whose the income is greater than 50k$ when the data distribution is stressed to change the proportion of women in the dataset.

From the Figure above computed for all the feature, *ethik* deduces a feature importance measure by computing, for a specific feature, the mean absolute difference between their influence curve and a horizontal curve equal to the original average prediction. The result obtained from this is given in the Figure below. According to this criterion, if there are no discrimination, the feature importance value for the gender should be equal to 0. The lower it is, the better it is in terms of discrimination.

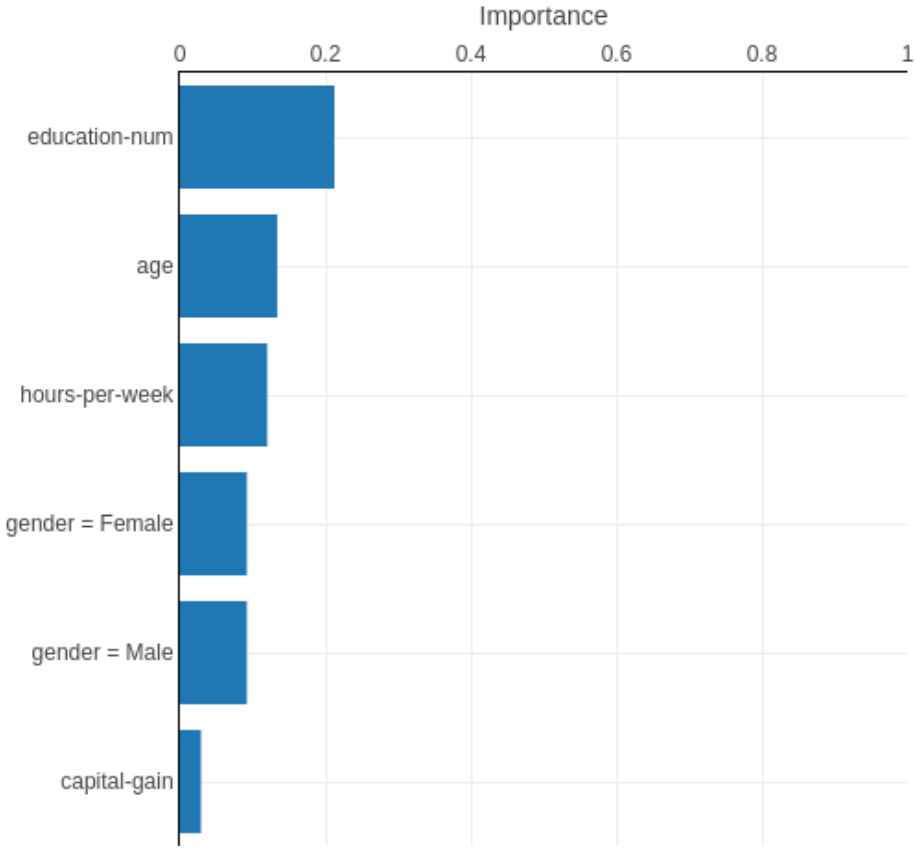



Figure 51: Features importance of some features according the true labels.

With *ethik* it is also possible to compare two instances of the dataset. We compare a woman to a man on the Figure below. To simplify the comparison, we call them Mary and Bob. According to this comparison, people of Mary's gender (women) are about 19% less likely to earn more than 50k$ per year than people of Bob's gender (men).

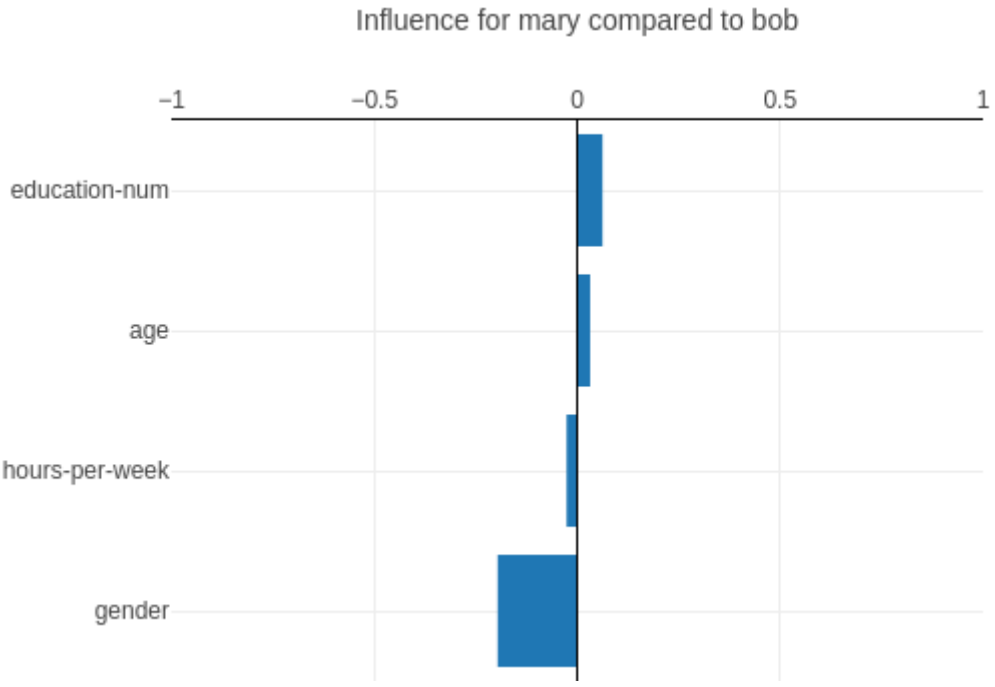



Figure 52: Comparison of two individuals from the testing one: one is a woman and the other are man.

In the previous figures, we looked at what was happening to the real population. We will now look at the same information concerning the predictions of the fair and the unfair neural networks. On the two Figures below, we plot the proportion of instances whose predicted income according the fair neural network (respectively the unfair) is greater than 50k$ when the data distribution is stress to change the proportion of women in the dataset. For both model, an increasing of the proportion of women leads to a decreasing of the instance predicted as earning more than 50k$. However, the slope of the curve is weaker in the case of the fair neural network, which discriminates less the women.

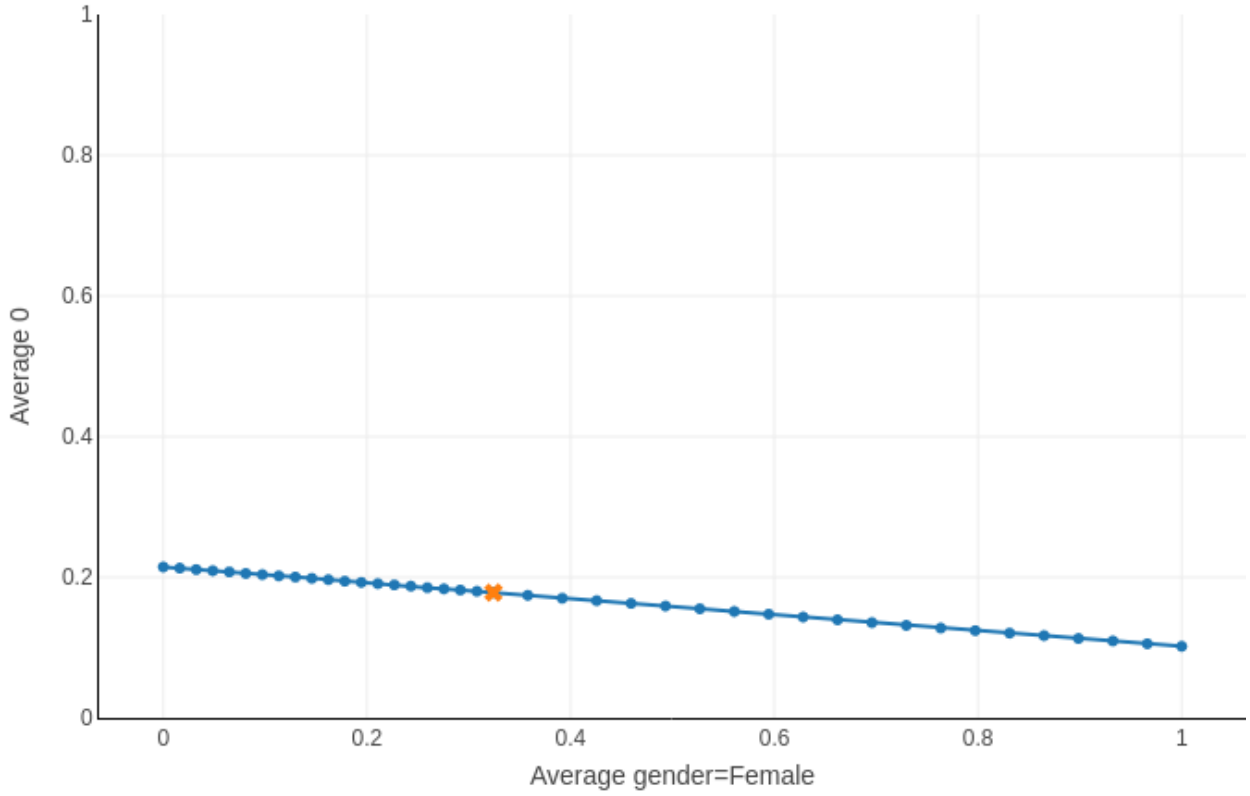



Figure 53: Proportion of instances whose predicted income according the fair neural network is greater than 50k$ when the data distribution is stressed to change the proportion of women in the dataset.

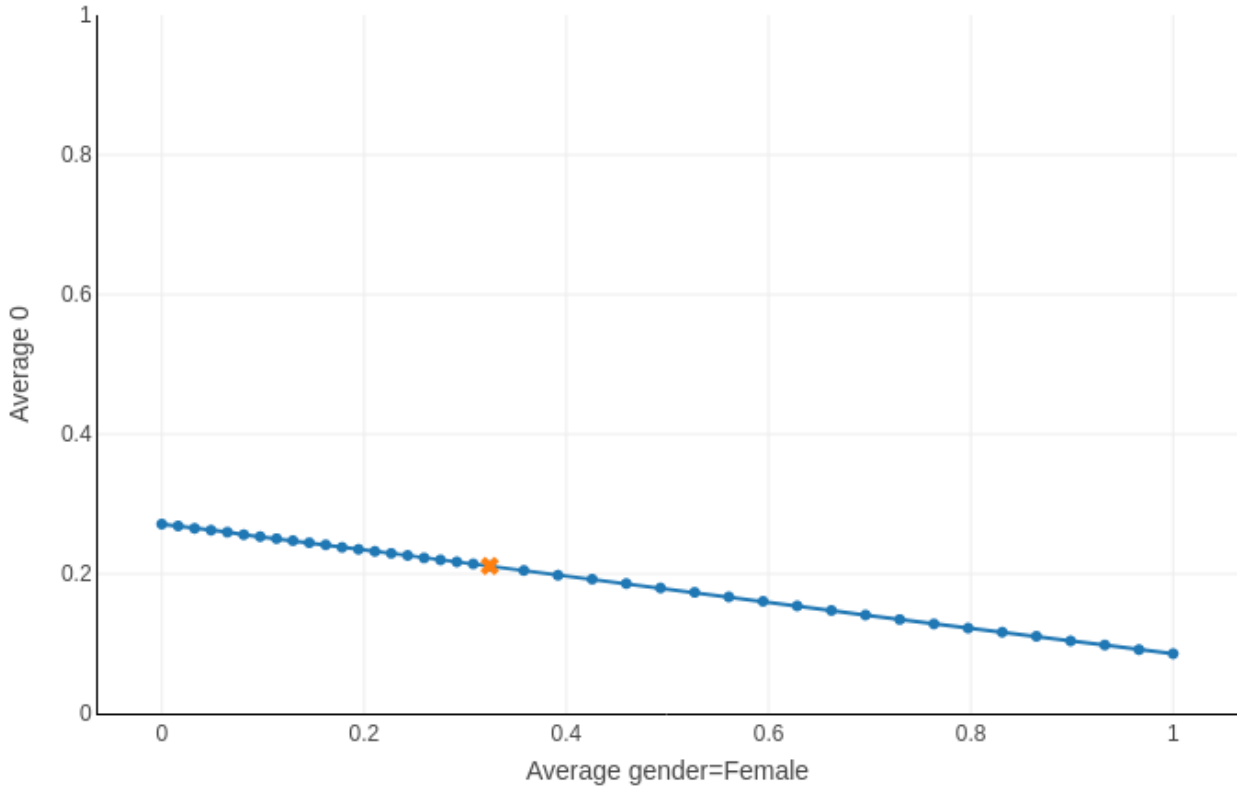



Figure 54: Proportion of instances whose predicted income according the unfair neural network is greater than 50k$ when the data distribution is stressed to change the proportion of women in the dataset.

The fact that women are less discriminated by the fair model is confirmed by the Figure below. This one represents the features importance criteria calculated as explained above for the fair (left) and unfair (right) models. The importance is weaker for the gender for the fair neural network.

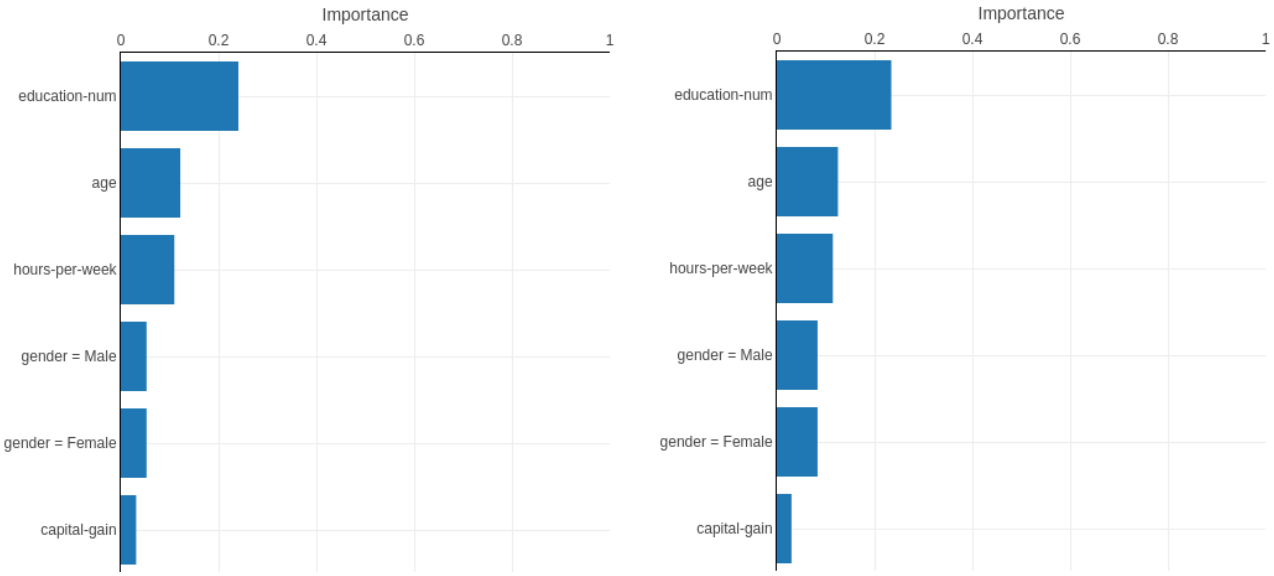



Figure 55: Features importance based on prediction of some features according for the fair (left) and unfair (right) neural networks.

We compare a woman to a man on the Figure below. To simplify the comparison, we call them Mary and Bob. According to this comparison, people of Mary's gender (women) are about 11% (resp. 18.5%) less likely to be predicted as earning more than 50k$ per year than people of Bob's gender (men) according the fair neural network (resp. the unfair neural network).

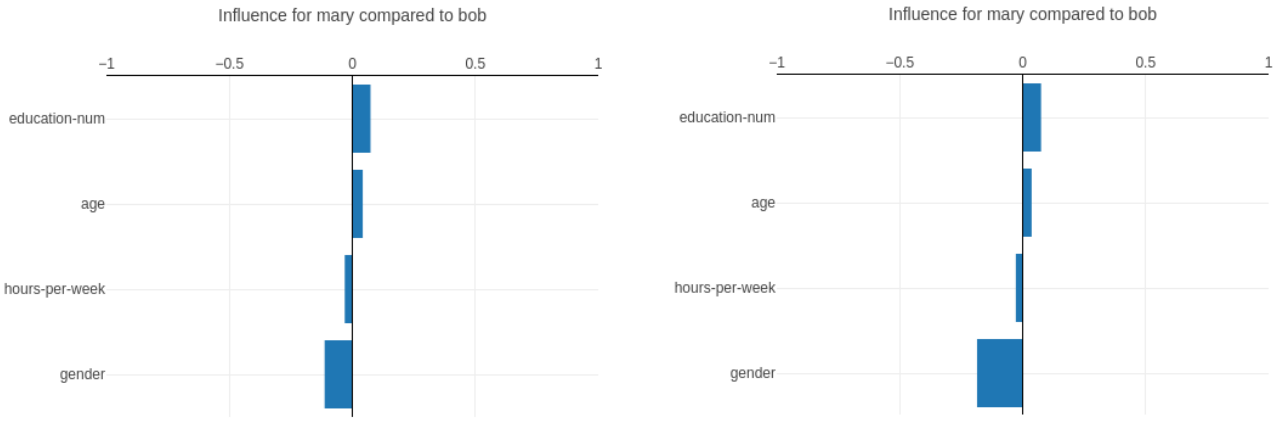



Figure 56: Comparison of two individuals according the fair (left) and unfair (right) from the testing one: one is a woman and the other are man.

#### 4.1.2.3 Impact of Fair Adversarial Network about the errors made for the gender

In the previous section, *ethik* allowed us to show that the fair neural network discriminates less against women in the sense that gender brings less disparity in the predictions of the model. Moreover, the performances of the two models are equivalent on a test sample. However, we can wonder if we are not accentuating another source of discrimination between the genders: the discrimination coming from the errors committed by the models according to the gender. Indeed, we have expressed in the deliverable 7.4 that it is impossible to fulfill the three fairness criteria (demographic parity, equal opportunity and predictive parity rate) simultaneously. It is possible with *ethik* to do same studies made for the average prediction for some performance metrics. In the two Figures below, we compute the accuracy when we stress the data distribution to force the proportion of women to change respectively for the fair and the unfair neural network. Again, if there is no impact of the gender, the blue curve should be horizontal line passing the orange point. For the two models, the accuracy is weaker when the proportion of women decreases. However, the two curves are similar, the two model discriminate in a similar way the woman and the man according the accuracy criteria.

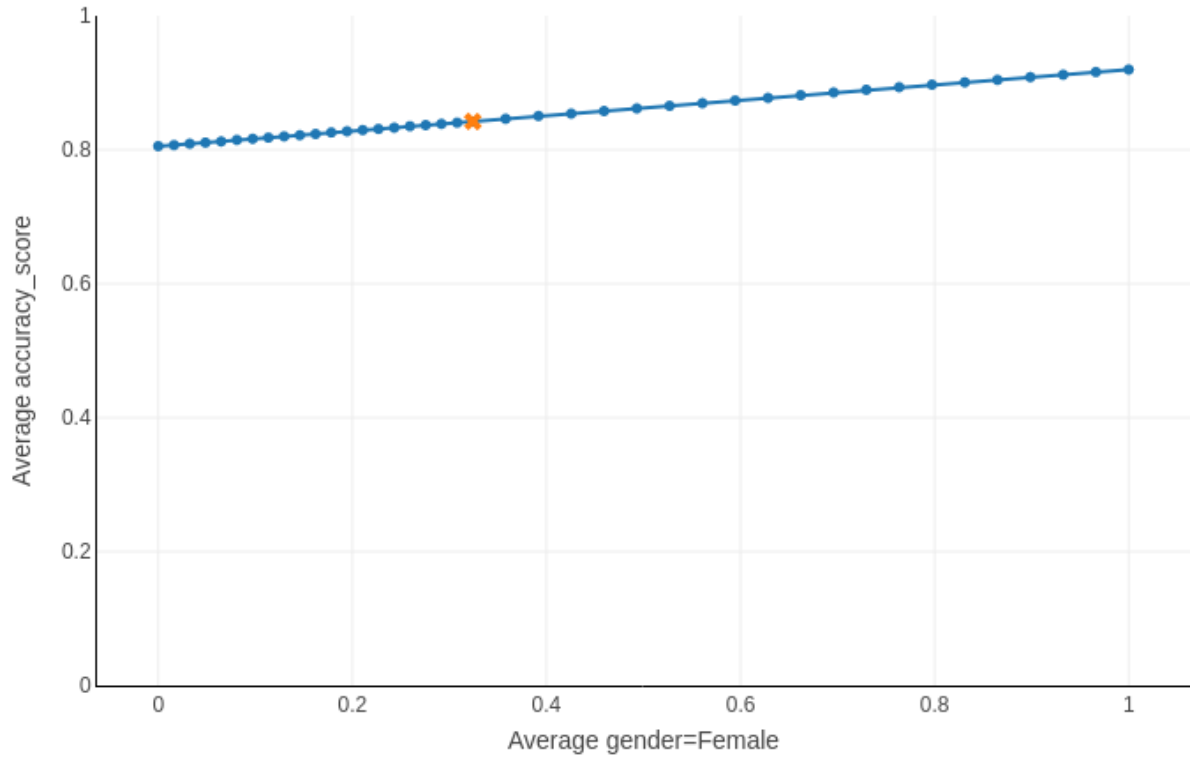



Figure 57: Accuracy of the fair neural network when the data distribution is stressed to change the proportion of women in the dataset.

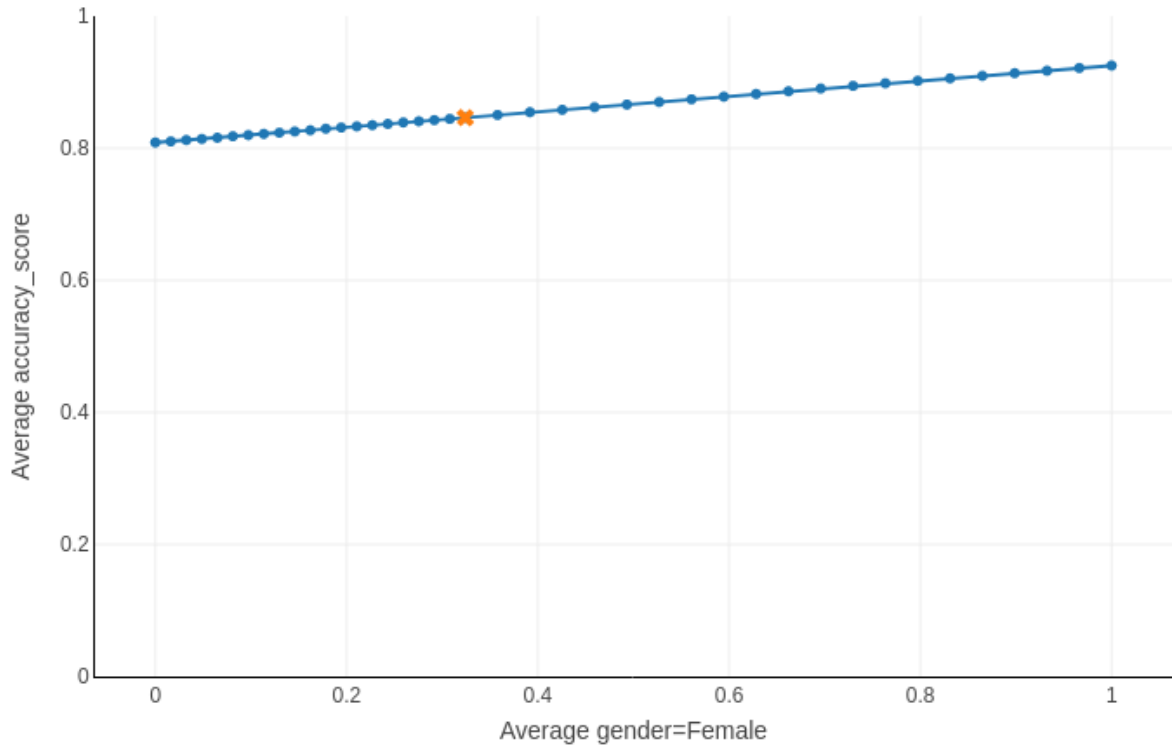



Figure 58: Accuracy of the unfair neural network when the data distribution is stressed to change the proportion of women in the dataset.

From previous curves, *ethik* allows to compute two features importance criteria, which are the minimum and the maximum accuracy after distribution stressing. If a feature does not impact the errors of the model, this two values should be near the accuracy obtain for the model on the original test set. The two Figures below represents these criteria for respectively the fair and the unfair models. For these two models, the gender has similar impact on the errors.

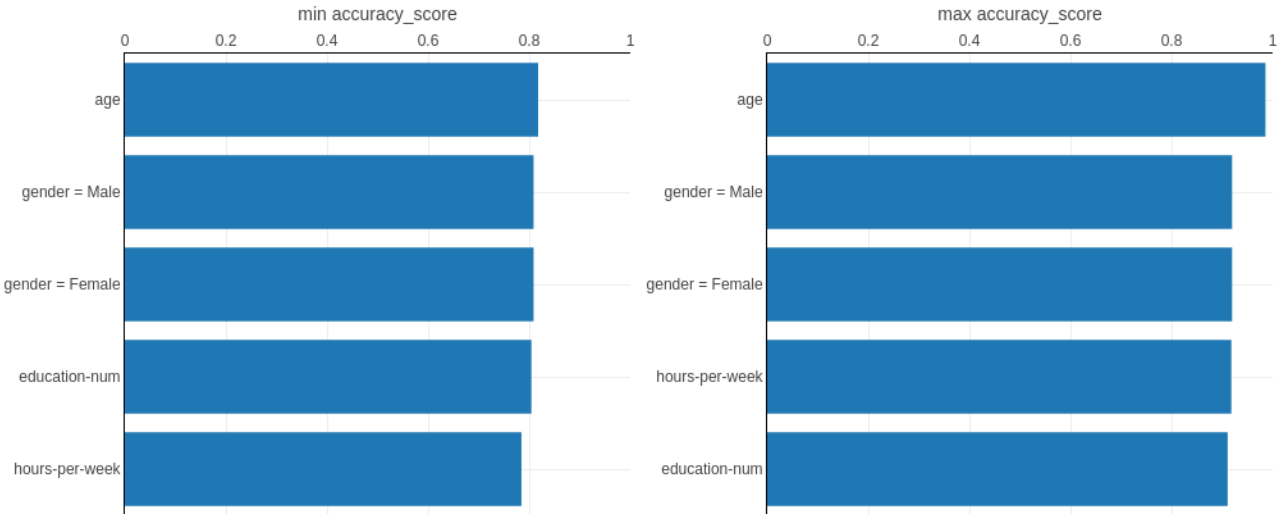



Figure 59: Features importance based on performance of some features according for the fair neural network.

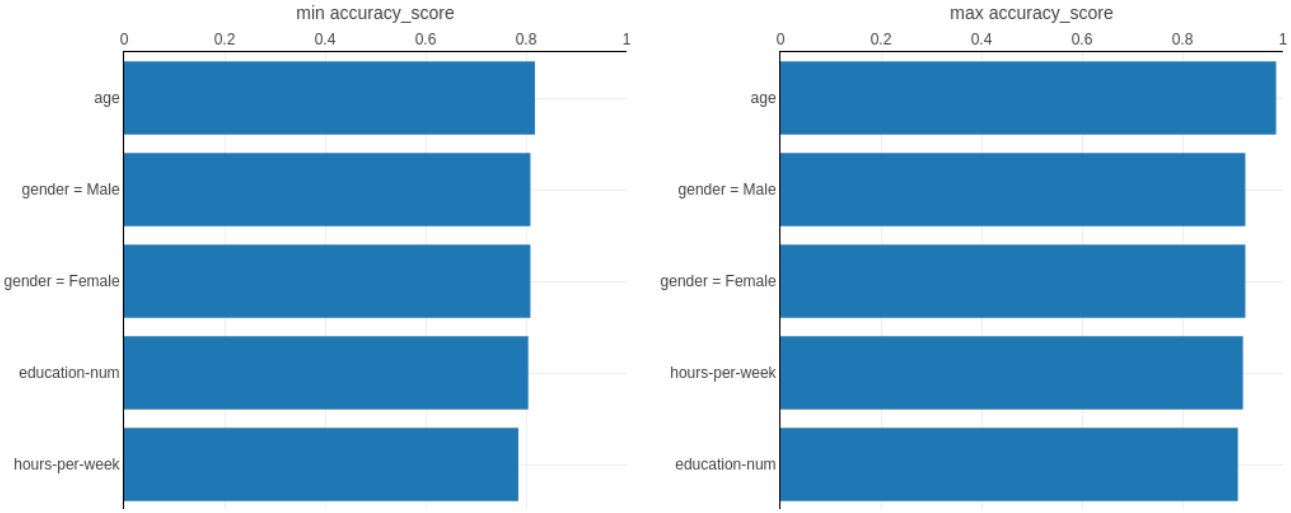



Figure 60: Features importance based on performance of some features according for the unfair neural network.

As with models prediction, it is possible with *ethik* to compare two individuals according to the accuracy. In the Figure below, we compare a woman, called Mary, and a man, called Bob. For the gender, this criterion is almost equal for the fair and the unfair models.

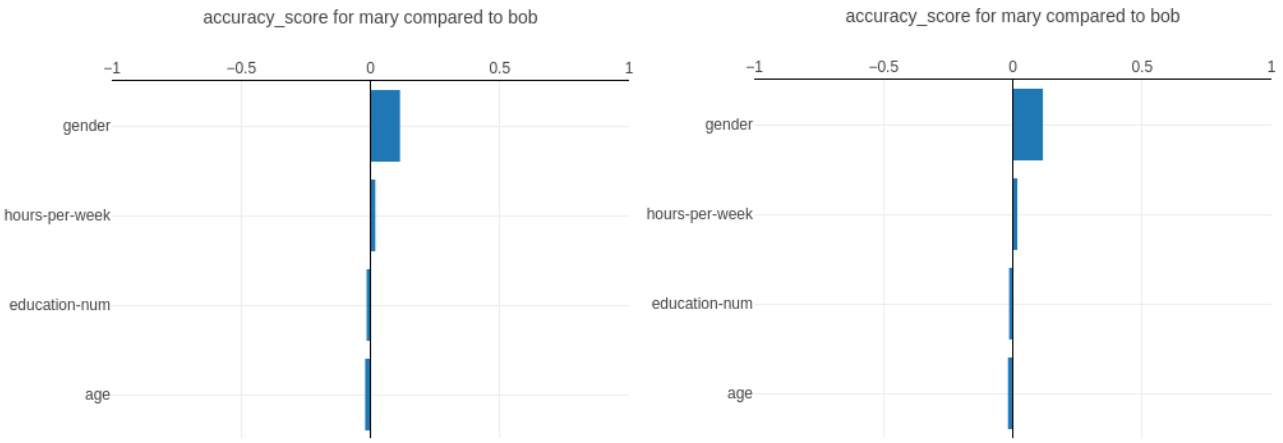



Figure 61: Comparison of two instances according the accuracy for the fair and unfair neural networks.

In this section, we use the accuracy, but we can perform the same study with any other performance metrics (AUC, etc.).

#### 4.1.2.4 Conclusion

Thanks in part to *ethik*, we can see that the fair model keeps overall performance and gender performance similar to an unfair model, while discriminating less against women in terms of the predicted value made for the income. Moreover, *ethik* allows to compute some plot about the impact of each feature on the model outputs and errors thus answering to interpretability issues.

## 4.2 Conclusion

In this chapter, we present the use of *ethik* on a classification task where we compare a fair adversarial network with an unfair adversarial network. This is only one use among others that can be done with this module. In deliverable 3.4, we define some metrics and algorithms to measure and obtain fairness. In this deliverable, we study a tool to inspect a model according its fairness properties. This module allows to enrich both the chapter dedicated to interpretability and to fairness.

# Chapter 5 SAFAIR adversarial AI contest results

The SAFAIR adversarial AI contest (the proposal and the plan for the contest is delivered by the SPARTA D7.3) aimed to evaluate the robustness of defence techniques proposed by participants. This encourages the creation of deep learning models which are robust to a variety of attack methods. In the contest, the defence teams created a variety of machine learning models based on the technique of their choices. They have submitted their defence solutions in the domain of image recognition by well-known or hybrid approaches. For more information about the structure and the content of the contest, please visit the contest homepage[8] and the deliverable D7.3. We should also notice that the full report analysis of the contest will be published in the M36 for the deliverable D7.6, and by this little section (Chapter 5), we are just announcing a summary of the contest submitted solutions.

## 5.1 Contest schedule

The SAFAIR adversarial AI contest was announced in February 2021, launched on March 1st, 2021 and ended on June 13th, 2021. Also, due to the Covid-19 pandemic and trying to have more participants, the contest organisers extended the final submission deadline two times till June 13th. The schedule summary for the contest is the following:

- **February 2021.** Launched the website with the announcement and contest rules. Start an active advertisement for the contest.
- **March 1, – June 13, 2021.** The contest has started in the first of March. Participants were working on their solutions. In the meantime, we organized a few intermediate rounds of evaluation.
- **June 13, 2021.** Deadline for the final submission.
- **June 13 – July 13, 2021.** Organisers have evaluated submissions.
- **July 31, 2021.** Announce contest results and release evaluation set of images.

## 5.2 Tasks

The contest proposed four different tasks and tracks, which is also available in deliverable D7.3 and contest homepage:

1. **Targeted Face Re-Identification.** In this track, participants are given a set of face images and target identities. 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.** In this track, participants are given a set of face images and a k-number of features ids. 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.** In this track, participants design models robust to perturbations for face attribute alterations. The purpose of this task is to create a machine learning model which is robust to adversarial perturbations in the attribute alteration scenario. For instance, detecting adversarial images accurately.
4. **Defence against Targeted Face Re-Identification.** In this track, participants design models robust to perturbations for face re-identification. The purpose of this task is to create a

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

machine learning model which is robust to adversarial perturbations to cause the model to classify the sample image as the particular target class.

## 5.3 Dataset

In the contest, we have used the CelebA dataset [86] to train the models. The CelebA dataset is a large-scale dataset of more than 200k celebrity images. The images are annotated with 40 facial attributes and consist of 10K unique identities with 40 binary attributes per image. The CelebA dataset is also publicly available.

We have released a development toolkit[9] to simplify access to the data. The development toolkit also consists of PyTorch code for baseline models. During the testing phase, the images have been chosen by the contest organisers. Furthermore, we have collected 1000 test images which are similar to the training dataset and they have been kept secret by TUM contest organisers till the end of the contest to evaluate the participant's solutions.

## 5.4 Evaluation metrics

For the final evaluation of participant's solutions, we have discussed in detail the parameters and methods in deliverable D7.3 and the contest homepage based on their Attack or Defence submissions. Regarding our participant's submitted solutions, they have just participated in defence track both in face re-identification and attribute alterations tasks. For that purpose, then we just needed to compute the delta value based on their initial model accuracy and the new accuracy computed after the attacks on the model. The delta value is computed as follows: $delta = D\_initial - D\_final$.

If two submissions have the same delta value, then the next metrics to rank the submission are the initial accuracy and run-time duration of submission. The higher in accuracy and lower in run-time values gets the better the ranking score.

## 5.5 Contest results

For the final round, we had in total seven submitted solutions in the Defence track. Six submissions are for the re-identification defence task and one for the attribute alteration defence task. For this deliverable D7.5, we will announce the best score value team, and for more information and analysis, the next deliverable, D7.6 is the full report for our submitted solutions.

The team “SD” has got 0 delta value on our attack baseline, which means his submitted model is robust 100% to our attacks lists. They have used three different methods to the robustness of the model against adversarial examples. On the first try, They have tried Projected Gradient Descent (PGD) training with hard examples with a Convolutional Neural Network (CNN). On the second try, they have submitted a transfer learning in combinations with gradient obfuscating. An ensemble of 6 learners, each individually trained on a random (with replacement) subset of the training set (i.e, bootstrapping). The final decision is made by averaging, not voting. Note that the ensembling was only there to increase clean accuracy and it does not play much in terms of defence. The gradient obfuscating technique after training is done during inference. Last but not least, in their third try, they have proposed a hybrid approach which is using the combination of transfer learning, gradient obfuscating, and adversarial training in a CNN model too. We should notice that they participated in the task of re-identification, and they have used the CelebA dataset for training the model.

Since the contest was designed in the two-player game, the attack submissions should evaluate the defence models. Regarding to our final deadline, we have not received any attacks submissions and, instead, we have plan to extend our attacks baselines in order to evaluate the robustness of submitted models in the next phase. The attack baselines in the contest are four well-known attack techniques to generate adversarial examples. The baseline attacks are Fast Gradient Sign Method (FGSM), Basic Iteractive Method (BIM), Carlini and Wagner (C&W), and Projected Gradient Descent (PGD).

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

## 5.6 Conclusion

Adversarial examples are an interesting phenomenon and an important problem in machine learning security. The main purposes of this contest were to raise awareness of the problem and motivate the researchers to introduce novel methods.

The contest also tried to stimulate people to investigate new approaches and enhance existing techniques to the problem. Top submissions in the defence track achieved very high accuracy on all adversarial images generated by all attack baselines, and we will fully describe all submissions in the deliverable D7.6.

# Chapter 6 SAFAIR AI Threat Model updates

The present chapter describes the updates performed on the SAFAIR AI Threat model and Knowledge Base of SPARTA deliverable D7.1 since Month 18, where the initially presented approach has been extended and improved to capture in the model new results from ENISA and other relevant initiatives on AI threat landscape.

## 6.1 Introduction

The initial version of the SAFAIR AI Threat model and accompanying Knowledge Base delivered as part of the D7.1 in Month 18 captured the results of the AI threat analysis and classification work by SAFAIR. Since then, a number of relevant efforts and reports have been issued including the "AI Cybersecurity Challenges" by (ENISA, 2020)[10] the Mitre's ATLAS - Adversarial Threat Landscape for Artificial-Intelligence Systems (MITRE ATLAS, 2021)[11] and the ETSI-SAI's "Mitigation Strategy Report".[12] While the first impacts directly in the SA SAFAIR AI Threat model definition and taxonomies used therein, the other two relate to instances of attack techniques and countermeasures, respectively.

Furthermore, the SAFAIR AI Threat Knowledge Base, which was delivered as part of the D7.1 too, is undergoing a continuous update and maintenance process to align the database structure with the latest updates in the SAFAIR AI Threat model, and to enlarge the knowledge corpus with defence mechanisms as well as explainability and fairness improvement solutions studied in the present D7.5. The knowledge base is also being extended with the results of a new literature review of the state-of-the-art AML attacks and safeguards. This way, the information captured in the database will keep reflecting cutting-edge knowledge by the end of the SPARTA project.

The following sections outline the main updates carried out on the SAFAIR AI threat model and Knowledge Base (Section 6.2 and Section 6.3 respectively), as well as the planned evaluation of both results within SAFAIR programme (Section 6.4).

Please note that D7.6 will report on the final version of the SAFAIR AI threat model and knowledge base together with the results of its evaluation.

## 6.2 Updates to SAFAIR AI Threat model

Bearing in mind that the main purpose of SAFAIR work on AI Threats is to aid raising awareness and capacity of European industry on trustworthy AI, it is clear that the initial version of the SAFAIR AI Threat model had to be revisited in the light of the new ENISA report (ENISA, 2020) which should be embraced as the main reference in Europe. The previous SAFAIR AI Threat model, which was already aligned with generic taxonomy in ENISA Big Data Threat landscape (Damiani et al, 2016)[13] was therefore updated so as to keep as much as possible consistency with the AI threats identified and classified by ENISA. In the following we summarize the major updates performed.

[10] 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

[11] MITRE ATLAS, Adversarial Threat Landscape for Artificial-Intelligence Systems. Available at: https://atlas.mitre.org/

[12] ETSI GR SAI 005 V1.1.1 (2021-03), Securing Artificial Intelligence (SAI); Mitigation Strategy Report

[13] E Damiani, C. A. Ardagna, F. Zavatarelli, E. Rekleitis (ed.) and L. Marinos, "Big Data Threat Landscape and Good Practice Guide", 2016. Available at: https://www.enisa.europa.eu/publications/bigdata-threat-landscape/at_download/fullReport

### 6.2.1 *Updates to asset taxonomy*

A deep analysis of ENISA's asset taxonomy dedicated to AI systems in (ENISA, 2020) has been conducted in order to identify to what extent the defined assets of the AI ecosystem should be included into the SAFAIR AI Asset taxonomy that we had already defined prior to the report was made public.

First, some new concepts have been included in the taxonomy, although sometimes in a different way. For instance, the new *Asset Category* defines at a high level the type of assets as: *Data*, *Model*, *Artefact* and *Environment/Tools (hardware/ software)*. However, other types of asset categories such as the involved *Actors* and *Processes* from ENISA report have not been included since the focus of SAFAIR asset taxonomy is on tangible technical elements of the AI system.

Second, the SAFAIR's concept named Target Asset has been renamed as *Affected Asset* which is the term used by ENISA. The *Affected Asset* would be the ultimate target or a malicious attack or the subject of an intentional threat. *Affected Assets* are instances of *Asset* and have an *Asset Category* from the list above.

Finally, it should be highlighted that the Asset taxonomy has been extended with some more specific terms such as *Labelled Data Set, Pre-processed Data Set* and *Model Testing Tool* among others.

### 6.2.2 *Updates to phase taxonomy*

The AI lifecycle reference model defined by ENISA has been analysed and, therefore, an extension to the Phase taxonomy has been performed. Previously, only training and testing/inference phases were considered. Therefore, a more granular definition has been applied with the aim of helping end-users to understand better where a specific threat applies. In this way, they could focus on applying security improvements to their weakest AI lifecycle points.

However, it is remarkable that not all ENISA's phases have been considered of interest. In fact, currently the Phase taxonomy only includes: *Data Ingestion*, *Data Pre-processing*, *Model Training*, *Model Tuning*, *Transfer Learning*, *Model Deployment* and *Model Maintenance.*

### 6.2.3 *Updates to phase taxonomy*

This chapter covers the updates of the SAFAIR concepts related to Threat Agents and Threat Groups.

Regarding Threat Agents and their knowledge, SAFAIR considered the Agent taxonomy defined originally in the ENISA Big Data Threat Landscape and Good Practice Guide (Jan 2016). This concept maintains such denomination, although ENISA denotes it currently with the term *Threat Actors* (ENISA, 2020). After the analysis of both taxonomies, it has been detected that updating some terms for a more modern denomination is needed.

In brief, the following categories of Threat agents were finally included in the SAFAIR AI Threat model which will keep the definitions b (ENISA, 2020):

- Cybercriminals.
- *Insiders,* new term that substitutes the previous *Employees* concept.
- Nation states.
- Cyberterrorists.
- *Hacktivists,* new term for ideologically motivated hackers. It substitutes the previous concept *Online social hackers*.
- Script kiddies.
- *Competitors,* new term that replaces the previous *Corporations* concept.

Regarding Threat Groups or type of threats, both, SAFAIR and ENISA AI Cybersecurity Challenges (ENISA, 2020) are based on the Threat group taxonomy defined originally in the ENISA Big Data Threat landscape (Damiani et al, 2016).

In brief, the following categories of Threat groups were finally made part of the SAFAIR AI Threat model which will keep the definitions by (ENISA, 2020):

- Nefarious Activity/Abuse.
- Unintentional damage, which has been renamed from "Unintentional damage/loss of information".
- Failures or malfunctions (new concept).
- Eavesdropping/ Interception/ Hijacking.
- *Physical attacks* (new concept added in the SAFAIR AI Threat model).
- *Outages* (new concept added in the SAFAIR AI Threat model).
- Legal.
- *Organisational* (not in the new ENISA taxonomy (ENISA, 2020) though it was already in the previous one (Damiani et al, 2016).
- *Fairness weakness*. It is a new concept, not defined by ENISA (ENISA, 2020) as such.
- *Explainability weakness*. It is a new concept, not defined by ENISA (ENISA, 2020) as such.

At this point, it is worthy to mention that the category *Organisational* threats that were part of the SAFAIR AI Threat model has been considered as still relevant and different from *Legal* threats, so it was not removed.

The last two concepts were also added in the SAFAIR AI Threat model in order to abstract the threats related to fairness (lack of bias) as well as the explainability (interpretability) respectively. Being these two aspects of Trustworthy AI so relevant for SAFAIR programme, we preferred to keep them separately even if they could fit within the "Unintentional damage" or "Failures or malfunctions" categories.

### 6.2.4 *Updates to attack technique taxonomy*

As a major international reference, the MITRE ATLAS's classification of techniques distinguishes between the following 12 main categories of attack techniques applicable to AI systems:

- Reconnaissance
- Resource Development
- Initial Access
- ML Model Access
- Execution
- Persistence
- Defence Evasion
- Discovery
- Collection
- ML Attack Staging
- Exfiltration
- Impact

These categories are the result of Mitre gathered results on real-world observations, demonstrations, and state-of-the-art literature analysis. The updated version of the SAFAIR AI Threat model is also embracing this categorization by Mitre (MITRE ATLAS, 2021) since there was no previous classification for the techniques in the SAFAIR AI Threat model.

## 6.3 Updates to SAFAIR AI Threat Knowledge Base

### 6.3.1 *Updates to threats*

Regarding new threat categories and threats to include in the SAFAIR AI Threat Knowledge Base different updates were made and we summarize below the updates due to the alignment with the new ENISA report (ENISA, 2020) The existing attack techniques in the database are also being mapped to the new attack technique categories defined by (MITRE ATLAS, 2021) introduced above.

The Table 19 differentiates between threats that are already included in the initial version of SAFAIR AI Threat model and those that will be included in the final version as needed updates.

Please note that according to the main focus of the SAFAIR AI Threat model which limits the attacks and incidents to assets and supporting artefacts within the AI system, rather than processes and people, some threats fall out of scope. Furthermore, physical attacks, outages and disasters are not the primary focus of the model and thus left for future extensions.

Table 19: SAFAIR AI Threat Knowledge Base threat updates to align with (ENISA, 2020).

| Threat Category | Threat | SAFAIR AI Threat |
|---|---|---|
| *Nefarious Activity/ Abuse* | *Access Control List (ACL) manipulation* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Adversarial examples* | This threat is already included as an attack technique. |
| | *Backdoor/insert attacks on training datasets* | This threat is already included as an attack technique. |
| | *Compromising AI inference's correctness - data* | This threat is already included by the following attack tactics:<br>• Direct poisoning - Data Manipulation - Label Manipulation<br>• Direct poisoning - Data Manipulation - Input Manipulation<br>• Direct poisoning - Data Injection<br>• Direct poisoning - Logic Corruption<br>However, *Bias in raw data* will be introduced as a new attack tactic. |
| | *Compromise and limit AI results* | Out of scope since this threat is related to methodological flaws. |
| | *Compromising ML inference's correctness – algorithms* | This threat is already included by the *Poisoning* and *Evasion* attack tactic groups. |
| | *Compromising ML pre- processing* | This threat will be included as a new attack tactic called *Schema Poisoning* and their corresponding attack techniques. |

| Threat Category | Threat | SAFAIR AI Threat |
|---|---|---|
| | *Compromising ML training – augmented data* | Out of scope since this threat is related to methodological flaws. |
| | *Compromising ML training – validation data* | This threat is partially included as the attack technique named *Adversarial examples*.<br>The part related to methodological flaws is considered out of scope. |
| | *Compromise of data brokers/providers* | This threat is already covered with the following attack tactics:<br>• Direct poisoning - Data Injection<br>• Direct poisoning - Data Manipulation - Input Manipulation |
| | *Compromise of model frameworks* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Corruption of data indexes* | This threat will be included as a new attack tactic and their corresponding attack techniques |
| | *Data poisoning* | This threat is already included by the *Poisoning* attack tactic groups. |
| | *Data tampering* | This threat is already included by the *Direct poisoning - Data Manipulation* attack tactics. |
| | *DDoS* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Elevation-of- Privilege* | This threat is already included by the *Oracle* attack tactic group. |
| | *Insider threat* | Out of scope since threats to *Actors* and *Processes* are not the focus of SAFAIR AI Threat model. |
| | *Introduction of selection bias* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Label manipulation or weak labelling* | This threat is already included by the attack tactic called as *Direct poisoning - Data Manipulation - Label Manipulation*. |
| | *Manipulation of data sets and data transfer process* | This threat is partially included by the attack tactic called as *Direct poisoning - Data Manipulation - Input Manipulation*.<br>The manipulation of data transfer process is not covered since *Processes* and their related concepts are not the focus of SAFAIR AI Threat model. |

| Threat Category | Threat | SAFAIR AI Threat |
|---|---|---|
| | *Manipulation of labelled data* | This threat is already included by the attack tactic called as *Direct poisoning - Data Manipulation - Label Manipulation.* |
| | *Manipulation of model tuning* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Manipulation of optimization algorithm* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Misclassification based on adversarial examples* | This threat is already included as an attack technique. |
| | *ML model confidentiality* | This threat is already covered by the attack tactics: *Oracle - Extraction and Oracle - Inversion.* |
| | *ML Model integrity manipulation* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Model backdoors* | This threat is already included as an attack technique. |
| | *Model poisoning* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Model Sabotage* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Online system manipulation* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Overloading/conf using labelled dataset* | This threat is already covered by the attack tactic called as *Direct poisoning - Data Injection*. |
| | *Reducing data accuracy* | This threat is already covered by the attack tactic called as *Direct poisoning - Data Injection*. |
| | *Reduce effectiveness of AI/ML results* | Out of scope since threats to *Actors* and *Processes* are not the focus of SAFAIR AI Threat model. |
| | *Sabotage* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Scarce data* | Out of scope since concepts related to methodological flaws are considered out of scope. |
| | *Transferability of adversarial attacks* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Unauthorized access to data sets and data transfer process* | This threat will be included as a new attack tactic and their corresponding attack techniques. |

| Threat Category | Threat | SAFAIR AI Threat |
|---|---|---|
| | *Unauthorized access to models' code* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *White-box, targeted or non-targeted* | Out of scope since threats to *Actors* and *Processes* are not the focus of SAFAIR AI Threat model. |
| *Unintentional Damage* | *Bias introduced by data owners* | Threats that may introduce bias will be added as a new attack tactics. |
| | *Compromising AI inference's correctness - data* | See same entry under the threat category *Nefarious Activity/Abuse*. |
| | *Compromise and limit AI results* | See same entry under the threat category *Nefarious Activity/Abuse*. |
| | *Compromising ML inference's correctness – algorithms* | See same entry under the threat category *Nefarious Activity/Abuse*. |
| | *Compromising ML training – augmented data* | See same entry under the threat category *Nefarious Activity/Abuse*. |
| | *Compromising feature selection* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Compromise of data brokers/providers* | See same entry under the threat category *Nefarious Activity/Abuse*. |
| | *Compromise of model frameworks* | See same entry under the threat category *Nefarious Activity/Abuse*. |
| | *Compromise privacy during data operations* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Disclosure of personal information* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Erroneous configuration of models* | Out of scope since *Processes* and their related concepts are not the focus of SAFAIR AI Threat model. |
| | *Label manipulation or weak labelling* | See same entry under the threat category *Nefarious Activity/Abuse*. |
| | *Lack of sufficient representation in data* | This threat will be included mainly the aspects related to the introduction of some biases. |
| | *Manipulation of labelled data* | See same entry under the threat category *Nefarious Activity/Abuse*. |
| | *Misconfiguration or mishandling of AI system* | Out of scope since *Processes* and their related concepts are not the focus of SAFAIR AI Threat model. |

| Threat Category | Threat | SAFAIR AI Threat |
|---|---|---|
| | *Mishandling of statistical data* | Out of scope since this threat is related to methodological flaws. |
| | *ML Model Performance Degradation* | Out of scope since this threat is related to methodological flaws. |
| | *Online system manipulation* | See same entry under the threat category *Nefarious Activity/Abuse*. |
| | *Reducing data accuracy* | See same entry under the threat category *Nefarious Activity/Abuse*. |
| *Legal* | *Compromise privacy during data operations* | See same entry under the threat category *Nefarious Activity/Abuse*. |
| | *Corruption of data indexes* | See same entry under the threat category *Nefarious Activity/Abuse*. |
| | *Disclosure of personal information* | See same entry under the threat category *Unintentional Damage.* |
| | *Lack of data governance policies* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Lack of data protection compliance of 3rd parties* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Profiling of end users* | Out of scope since this threat is related to methodological flaws. |
| | *SLA breach* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Vendor lock-in* | Out of scope since this threat is related to methodological flaws. |
| | *Weak requirements analysis* | Out of scope since this threat is related to methodological flaws. |
| *Failures or malfunction* | *Compromising AI application viability* | Out of scope since this threat is related to methodological flaws. |
| | *Compromising ML pre- processing* | See same entry under the threat category *Nefarious Activity/Abuse*. |
| | *Corruption of data indexes* | See same entry under the threat category *Nefarious Activity/Abuse*. |
| | *Compromise of model frameworks* | See same entry under the threat category *Nefarious Activity/Abuse*. |

| Threat Category | Threat | SAFAIR AI Threat |
|---|---|---|
| | *Errors or timely restrictions due to non-reliable data infrastructures* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Inadequate/absen t data quality checks* | Out of scope since this threat is related to methodological flaws. |
| | *Label manipulation or weak labelling* | See same entry under the threat category *Nefarious Activity/Abuse*. |
| | *Lack of documentation* | Out of scope since this threat is related to methodological flaws. |
| | *ML Model Performance Degradation* | Out of scope since this threat is related to methodological flaws. |
| | *Poor resource planning* | Out of scope since this threat is related to methodological flaws. |
| | *Scarce data* | See same entry under the threat category *Nefarious Activity/Abuse*. |
| | *Stream interruption* | Out of scope since *Processes* and their related concepts are not the focus of SAFAIR AI Threat model. |
| | *Weak data governance policies* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| | *Weak requirements analysis* | Out of scope since this threat is related to methodological flaws. |
| | *3rd party provider failure* | This threat will be included as a new attack tactic and their corresponding attack techniques. |
| *Eavesdropping*<br>*Interception*<br>*Hijacking* | *Data inference* | This threat is already included as attack *tactic Oracle – Inversion and Oracle - Membership Inference*. |
| | *Data theft* | This threat is already included as *Data Access* attack tactic |
| | *Model Disclosure* | This threat is already included as attack tactic *Oracle – Inversion and Oracle – Extraction*. |
| | *Stream interruption* | See same entry under the threat category *Failures or malfunctions.* |
| | *Weak encryption* | Out of scope since this threat is related to methodological flaws. |
| *Physical attacks* | Not in the scope of the SAFAIR Threat model. | |

| Threat Category | Threat | SAFAIR AI Threat |
|---|---|---|
| *Outages* | Not in the scope of the SAFAIR Threat model. | |
| *Disasters* | Not in the scope of the SAFAIR Threat model. | |

### 6.3.2 *Updates to asset countermeasures*

Several updates to countermeasure instances gathered in the SAFAIR AI Threat Knowledge Base were also necessary so as to enrich the contents with new researched techniques and methods. Most relevant for this deliverable are the ones resulted from the work in SAFAIR that was described in previous chapters. The following table summarizes the techniques introduced in the Knowledge Base together with the corresponding reference.

Table 20: SAFAIR countermeasures added in the SAFAIR AI Threat Knowledge Base.

| Countermeasure name | Description | Attack Technique/Weaknesss countered | ML Algorithm | Source |
|---|---|---|---|---|
| Adversarial training | Inject adversarial examples into the training dataset so as to increase model robustness | Evasion attacks<br>o FGSM (Fast gradient sign method)<br>o iter-FGSM (iterative Fast gradient sign method)<br>o C&W (Carlini and Wagner)<br>o CIA (Centered Initial Attack) | SVM | SAFAIR D7.4 Chapter 3 and D7.5 chapter 2 |
| Feature scattering | The technique aims to maximize the distance between the outputs relative to clean and perturbed images while respecting a norm constraint during the training of the neural network, while keeping the good labels on the perturbed data. The optimal transport (OT) distance between the distributions of the sets of the extracted features of the clean and perturbed sets of images is used. | Evasion attacks | NN | SAFAIR D7.4 Chapter 3 and D7.5 chapter 2 |
| Hybrid approach com- | The technique applies feature scattering on the robustification of a model | Evasion attacks | Random Forest | SAFAIR D7.4 Chapter 3 and D7.5 chapter 2 |

| Countermeasure name | Description | Attack Technique/Weaknesss countered | ML Algorithm | Source |
|---|---|---|---|---|
| bining Features scattering and Adversarial Training | through adversarial examples generated through a proxy model trained using feature scattering. | | | |
| Neuron Behaviour descriptors | The objective is to analyse the behaviour of each neuron that compounds the Area of interest of a DL model. The neurons' behaviours are grouped according to the sample that has generated them. Hence, each sample is associated with a group of labels that describe the behaviour of the model in the area of interest. These descriptions can be used to to detect malicious input samples that are adversarial examples. | Evasion attacks | DL models | SAFAIR D7.4 Chapter 3 and D7.5 chapter 2 |
| Additive local explanation | The technique explains the reasoning of the model for one given instance by explaining the deviation of its prediction from the prediction of an average instance of a reference population by the sum of contribution of features. | Explainability weakness | Regression, binary classification and multi-label classification | SAFAIR D7.4 Chapter 4 and D7.5 chapter 3 |
| Fair adversarial network | The technique aims at rendering independent the outputs of a classifier regarding given sensitive features. The method is able to constraint the optimization objective via a zero-sum game which is classically used for training Generative Adversarial Networks (GANs). | Fairness weakness | Artificial Neural Networks | SAFAIR D7.4 Chapter 5 and D7.5 chapter 4 |
| Fair random forest | In the fair random forest, for each node of the tree, the split is chosen based on a criterion which takes into account the target and | Fairness weakness | Random forest | SAFAIR D7.4 Chapter 5 and D7.5 chapter 4 |

| Countermeasure name | Description | Attack Technique/Weaknesss countered | ML Algorithm | Source |
|---|---|---|---|---|
| | also the protected attribute(s). Indeed, the algorithm chooses the best split according to the information gain which is a function of the set of training examples and the attribute to split on. The method introduces a penalization parameter to control more finely the trade-off between accuracy and fairness. | | | |

## 6.4 SAFAIR AI Threat model evaluation

This section describes the methodology and the work plan for the evaluation of the SAFAIR AI Threat model. SAFAIR AI Threat model organises cyber risk knowledge to support the design, development, operation and maintenance of secure and privacy-aware ML systems.

The evaluation of the SAFAIR AI Threat Model will be carried out through the evaluation of the SAFAIR AI Threat Knowledge Base tool (a.k.a. Knowledge Base), an Open Source tool that was developed in the context of T7.1 “Threat Modelling for AI systems”. So, this evaluation process will allow Tecnalia to test and validate the results of T7.1 in the context of T7.5 “Testing and validation”.

### 6.4.1 *Evaluation Objectives*

The main objective of the evaluation process is to assess if the actual content of the Knowledge Base is correct and sufficient for the users when interacting with it, i.e. we will check the quality and completeness of its content.

### 6.4.2 *Evaluation Dimensions*

We will consider five dimensions in the evaluation process, being 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.

### 6.4.3 *Evaluation Means*

A **questionnaire** with dedicated questions to assess the different dimensions above will be designed to support the evaluation process. The questionnaire will be an online questionnaire to be filled by evaluators after using the Knowledge Base and will contain several questions to get feedback on different aspects of the tool:

- General questions about the evaluation practices
- Technical questions related to the content of the Knowledge Base
- Questions on user satisfaction with the use of the tool

Both the questionnaire editing and the questionnaire processing procedures carried out by Tecnalia will follow GDPR rules with regards to protection of personal information of the responder data subjects.

The final content of the questionnaire, the statistics of questionnaire responses gathered and the analysis of the results will be provided in D7.6 “Validation and evaluation report” (M36).

### 6.4.4 *Evaluators*

The **profile** of the evaluator is an AI designer or AI developer, that is responsible for developing AI based systems, and that aims to use the Knowledge database to search information of certain threats against AI systems. Therefore, the evaluation process will allow to get feedback from actual AI designers and developers using the Knowledge Base.

All the evaluators will be volunteers. At least ten people from SAFAIR program partners that work in the design and development of AI systems from potentially different sectors (health, energy, etc.) will be involved.

### 6.4.5 *Evaluation process*

The evaluation of the SAFAIR AI Threat Model and Knowledge Base will be carried out as part of the task T7.5 where evaluators will work hands on with the SAFAIR AI Threat Knowledge Base. Figure 62 shows the evaluation process that will be followed.

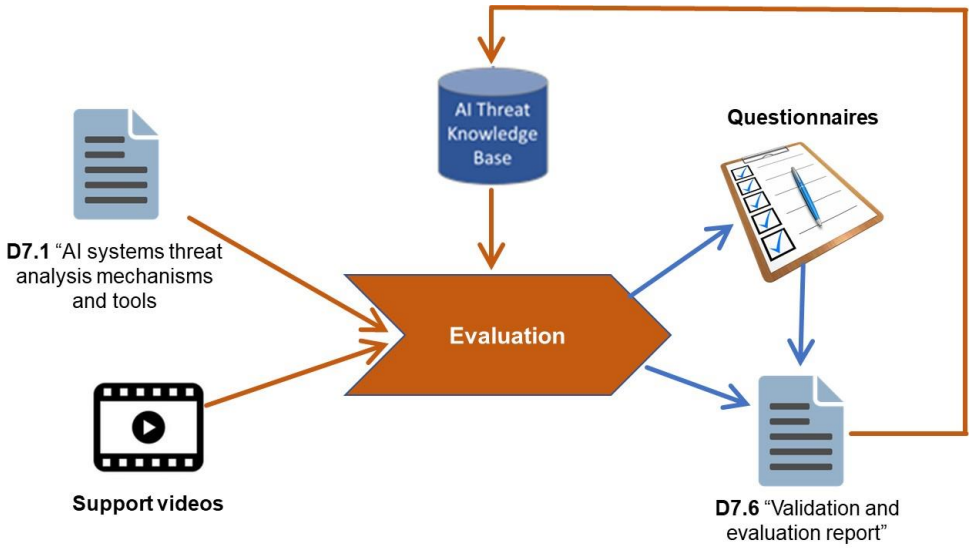



Figure 62: Knowledge Base Evaluation process

Before the validation starts, the evaluators will be provided with the following **inputs** that will be available in a git repository:

- D7.1 “AI systems threat analysis mechanisms and tools” document, which describes the AI Threat model and summarizes the literature review performed in SAFAIR.
- Support videos. Evaluators will have the possibility to watch some videos illustrating two example use cases of the SAFAIR AI Threat Knowledge Base:
  - UC1: Threat Analysis on AI-based Healthcare system, where the AI system is a Healthcare system for disease detection implemented with AI, and a Poisoning attack is launched, that alters the training process, typically by modifying the training dataset.
  - UC2: AI-based network traffic classification system, where the AI system it a Network traffic classification system realized by Support Vector Machines (SVM), and an attack is launched to hack a trained classifier to obtain information that was implicitly absorbed from the elements the classifier received as input.
- SAFAIR AI Threat Knowledge Base tool (part of deliverable D7.1 too).

During the evaluation, participants will be asked to perform a **threat analysis** of the AI systems under test using the SAFAIR AI Threat Knowledge Base. The information about the threats and associated potential countermeasures is organised in a form easily searchable by AI system designers and operators.

Evaluators will consult the Knowledge Base to search for information of threats against AI systems, including attack techniques that can be performed against confidentiality, integrity and availability of both data and learning models of AI systems. The interaction with the tool will involve three main steps:

- Step 1: Identify the AI asset
- 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

The evaluation process will be performed before D7.6 due date, so the results of the evaluation can be reported therein. The evaluation will allow to obtain the first results and also will provide some relevant feedback to improve the database and the questionnaire design, so the questionnaire can be improved and made available in git together with the Knowledge Base.

Both questionnaire and Knowledge Base will be available in this git repository so outsiders from SPARTA can also use and evaluate the tool, and we can get feedback and statistics about its use beyond the project end.

The answers to the questionnaire and the details and results of the conducted evaluation and validation will be documented in D7.6 “Validation and evaluation report” (M36).

The results of the evaluation will therefore allow to update the SAFAIR AI Threat Knowledge Base, mainly by updating and extending its content when necessary. The final version of the Knowledge Base will also be provided in D7.6.

# Chapter 7 Legal aspects

## 7.1 Introduction

Since AI methods are data-driven, biases towards the data used for training must be overcome to guarantee appropriate functionality that does not lead to discrimination. Decisions based on AI systems must be made accountable and correct in accordance with EU regulations (such as the GDPR), but also other emerging documents and guidelines such as those of the European Data Protection Council (former Article 29 Working Party) and the Toronto Declaration (May 2018) on preventing the use of machine learning to sustain discrimination. The methodology developed in these tasks should make fairness a feature of AI systems that does not compromise performance, but truly enhances the AI system due to the reduction of conscious or unconscious bias.

Recently, the European High-level Expert Group released a new version of the guidelines for trustworthy AI. Special attention will be given to this new documentation.

After explaining the methodology and the main findings from the previous deliverables (D7.2 and D7.4), the first part of this chapter aims at establishing a practical checklist for AI software developers in order to respect the equity criteria throughout the development process. The second part will link the different elements of the fairness principle with the algorithms proposed by our technical partners in this deliverable.

This analysis does not include algorithms or technologies to enhance the protection of personal data but focuses on the fairness principle to avoid bias in AI system.

Through the different graphs (see part 4 of this chapter), taking up the main considerations of the High-level Expert Group on AI, we'll see that the notion of fairness does not only concern the result given by the artificial intelligence tool (and the absence of bias) but also the process of realisation of this technology and the choice of data and the justification of the processing (see Table 1).

This is reinforced in the General Data Protection Regulation. Indeed, the notion of fairness, which could apply both to the processing itself and to the result obtained, reflects the primary consideration of the EU legislator when drafting Article 22: not to leave the processing of personal data solely and entirely to a machine and to make it understandable for human beings. Fairness in the GDPR seems to be linked with the transparency requirement for the data controller and the expectations in each circumstance of data subjects[14].

[14] See MALGIERI, G., The Concept of Fairness in the GDPR: A Linguistic and Contextual Interpretation (January 10, 2020). Proceedings of FAT* '20, January 27–30, 2020. ACM, New York, NY, USA, 14 pages. DOI: 10.1145/3351095.3372868. Available at SSRN: https://ssrn.com/abstract=3517264; M. Knockaert, "GDPR and Automated individual decision-making: Fair processing v. Fair result", *in Deep diving into data protection: 1979-2019: celebrating 40 years of research on privacy data protection at the CRIDS*, Bruxelles, Larcier, 2021, p. 251.

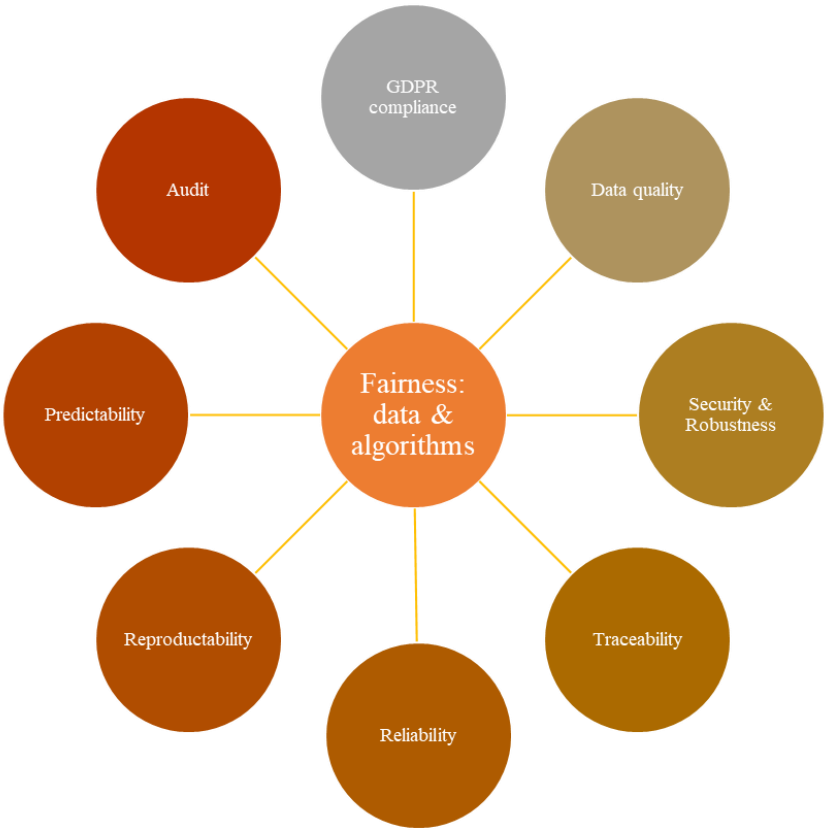



Figure 63: Criteria for fairness in data and algorithms used for AI.

## 7.2 Methodology

The overall goal of the Task 7.4 and 7.5 is the development of a systematic method to prevent machine learning and the decisions based on AI systems from being used to create discrimination.

Knowing that the guidelines for trustworthy AI published by the European High-Level Expert Group on AI constitute the most recent and accomplished document on the subject, we have partly taken the list of different questions they propose, taking care to rework and complete it.

In the first part of this chapter, we propose a practical checklist for AI software designers. This list, mainly based on the guidelines from the High-level Expert Group on AI from the European Commission as explained before, is divided into three steps: Before the design - During the development - At the end of the project or if someone wants to withdraw from the database. Each step has three dimensions: organizational, legal and technical. The AI software developer should read and verify each statement in this list to minimize the risk of discrimination. The purpose of this restructuring is to have a clearer and more practical list for the AI software designer at each stage of project development. This will make it easier to verify compliance with legal fairness requirements.

The second part of the chapter aims to summarize the different algorithms proposed by our technical partners for this task and to verify if they meet the legal requirements for fairness. This will allow us to understand the usefulness of each of these algorithms. Finally, a few lines are devoted to the European Commission's new proposal to regulate artificial intelligence activities in the European Union.

## 7.3 Main findings of previous deliverables (D7.2 – D7.4)

Based on the most recent documents existing at the European level, the previous deliverables were an opportunity to analyse the European strategy in front of the increasing development of artificial intelligence.

- European Parliament, "Recommendations to the Commission on Civil Law Rules on Robotics"[15]
- EPRS, "EU guidelines on ethics in artificial intelligence: Context and implementation"[16]
- EPRS, "A governance framework for algorithmic accountability and transparency"[17]
- European Commission, "Artificial Intelligence for Europe"[18]
- European Commission, "Building Trust in Human-Centric Artificial Intelligence"[19]
- High-Level Expert Group on Artificial Intelligence, "A definition of AI: main capabilities and disciplines"[20]
- High-level Expert Group on AI, "Ethics Guidelines for Trustworthy AI"[21]
- High-level Expert Group on AI, "Policy and Investment Recommendations for Trustworthy AI"[22]
- European Consumer Organization, "AI Rights for Consumers"[23]

The European Parliamentary Research Service[24] observes that fairness is a multi-faceted notion: "*Fairness reflects the appreciation of a situation based on a set of social values, such as promoting equality in society. The assessment of fairness depends on facts, events, and goals, and therefore has to be understood as situation or task-specific and necessarily addressed within the scope of a practice (…). The concept of fairness in the context of algorithmic implementations appears as a balance between the mutual interests, need and values of different stakeholders affected by the algorithmic decisions*"[25].

On the basis of this observation, we then tried to provide initial solutions. See "Sources of unfairness and solutions" and the summary of the guidelines from the independent High-level Expert Group on AI in the D7.2 document. Finally, we have also adopted an approach based on the notion of fairness in the GDPR (e.g. transparency and quality of the information, integrity and confidentiality of the data, misuse of data, data quality and the right to obtain a human intervention and the right to object). A particular attention has been paid to the right not to be subject to an automated-decision.

In the D7.4, we saw that, at the international level, the major concerns about artificial intelligence are the requirement of transparency and explainability of the algorithms and the AI mechanism, accountability and a secured AI system. While the European Commission also takes up these three criteria, it adds the need to place people at the heart of technological development. This mainly includes compliance with the entire European regulatory framework (in particular compliance with the European framework on the protection of personal data at all stages of AI deployment and during the

[15] Available at: https://www.europarl.europa.eu/doceo/document/A-8-2017-0005_EN.html
[16] Available at: https://www.europarl.europa.eu/thinktank/en/document.html?reference=EPRS_BRI(2019)640163
[17] Available at: https://www.europarl.europa.eu/thinktank/en/document.html?reference=EPRS_STU(2019)624262
[18] Available at: https://ec.europa.eu/digital-single-market/en/news/communication-artificial-intelligence-europe
[19] Available at: https://ec.europa.eu/digital-single-market/en/news/communication-building-trust-human-centric-artificial-intelligence
[20] Available at: https://ec.europa.eu/digital-single-market/en/news/definition-artificial-intelligence-main-capabilities-and-scientific-disciplines
[21] Available at: https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai
[22] Available at: https://ec.europa.eu/digital-single-market/en/news/policy-and-investment-recommendations-trustworthy-artificial-intelligence
[23] Available at: https://www.beuc.eu/publications/ai-rights-consumers/html
[24] EPRS, "A governance framework for algorithmic accountability and transparency", 2019. Available at: https://www.europarl.europa.eu/thinktank/en/document.html?reference=EPRS_STU(2019)624262
[25] EPRS, "A governance framework for algorithmic accountability and transparency", 2019, p. 10.

processing of this information by the machine itself) and safeguarding the fundamental rights of individuals. In parallel, the European Commission is also focusing on the use of data, putting in place a framework to promote data sharing while protecting personal data against illegal processing.

It seems undeniable that the latest tool made available to stakeholders by the European Commission, namely the ALTAI guidelines, will become increasingly important in the future. In essence, in order to strengthen trust in AI, the European Commission focuses on three main aspects, namely a transparent governance and the respect of personal data, a high level of security and the possibility of human control in artificial intelligence activities.

A prototype web-based tool to help developers and deployers of AI from a practical point of view through an accessible and dynamic checklist has been developed: https://futurium.ec.europa.eu/en/european-ai-alliance/pages/altai-assessment-list-trustworthy-artificial-intelligence. The objective of this approach is twofold. First, the High-level Expert Group on AI and the European Commission wanted to provide a visualisation of the self-assessed level of adherence of the AI system. Second, some recommendations to enhance the system will be provided. Additionally, having the same basis for evaluating the system also allows for a harmonised approach to the notion of fairness.

## 7.4 Check-list

Figure below shows the structure used in this deliverable. Indeed, we divide the realisation of an AI tool into three main steps. Our objective is to divide the recommendations for a fair artificial intelligence tool from the High-level Expert Group into these three phases: 1) before any technical development 2) during the development of an AI system and 3) at the end of the AI service (see Table 2). At each of these stages, the recommendations are divided into elements relating to the operation and management of an artificial intelligence system. Each phase of the project must therefore meet organisational, technical and legal aspects.

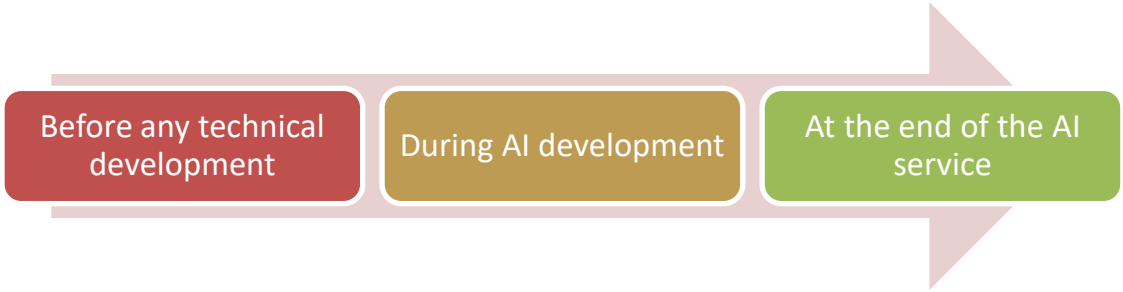



Figure 64: Timeline for the consideration of fairness

### 7.4.1 *Before starting developing AI*

#### 7.4.1.1 Introduction

Before even beginning to develop any IA project, the people in charge of this development must ask themselves a number of organizational, legal and technical questions in order to determine a number of processes related to the respect of the principle of fairness. Indeed, fairness is not only relevant for the duration of the development or as a final result, it must be considered from the idea stage to the finished product. The table below shows the common structure for each phase of AI development. We suggest a reading grid for the guidelines developed by the High-level Expert Group on AI[26]. We have tried to divide the recommendations according to the different stages of a project to develop an artificial intelligence tool.

[26] High-level Expert Group on Artificial Intelligence, "Assessment list for Trustworthy AI", 2020. Available at: https://ec.europa.eu/digital-single-market/en/news/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment

| Organisational dimension | Legal dimension | Technical dimension |
|---|---|---|
| Human monitoring | Personal data protection | Avoidance of unfair bias |
| Data quality and data governance | | Security, robustness and accuracy |
| Transparency | | Traceability |
| Accountability and risk management | | Reliability and reproducibility |
| Accessibility | | Explainability |

Table 21: Structure of the guidelines for the notion of fairness

#### 7.4.1.2 Organisational dimension

Based on the guidelines from the High-Level Expert Group on AI[27], Figure below lists the main components to assess fairness in AI from an organisational dimension. The purpose of this section is to encourage the project leader to adopt certain practices before actually starting to develop an artificial intelligence tool. For example, ensuring the quality of the input data, monitoring from the outset and throughout the process, and considering how best to ensure the transparency (transparency by design) of the system are crucial. In addition, the project manager should, from the outset, think about the risks that artificial intelligence can create, make the developers aware of the risks and set up a risk monitoring procedure.

[27] High-level Expert Group on Artificial Intelligence, “Assessment list for Trustworthy AI”, 2020. Available at: https://ec.europa.eu/digital-single-market/en/news/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment

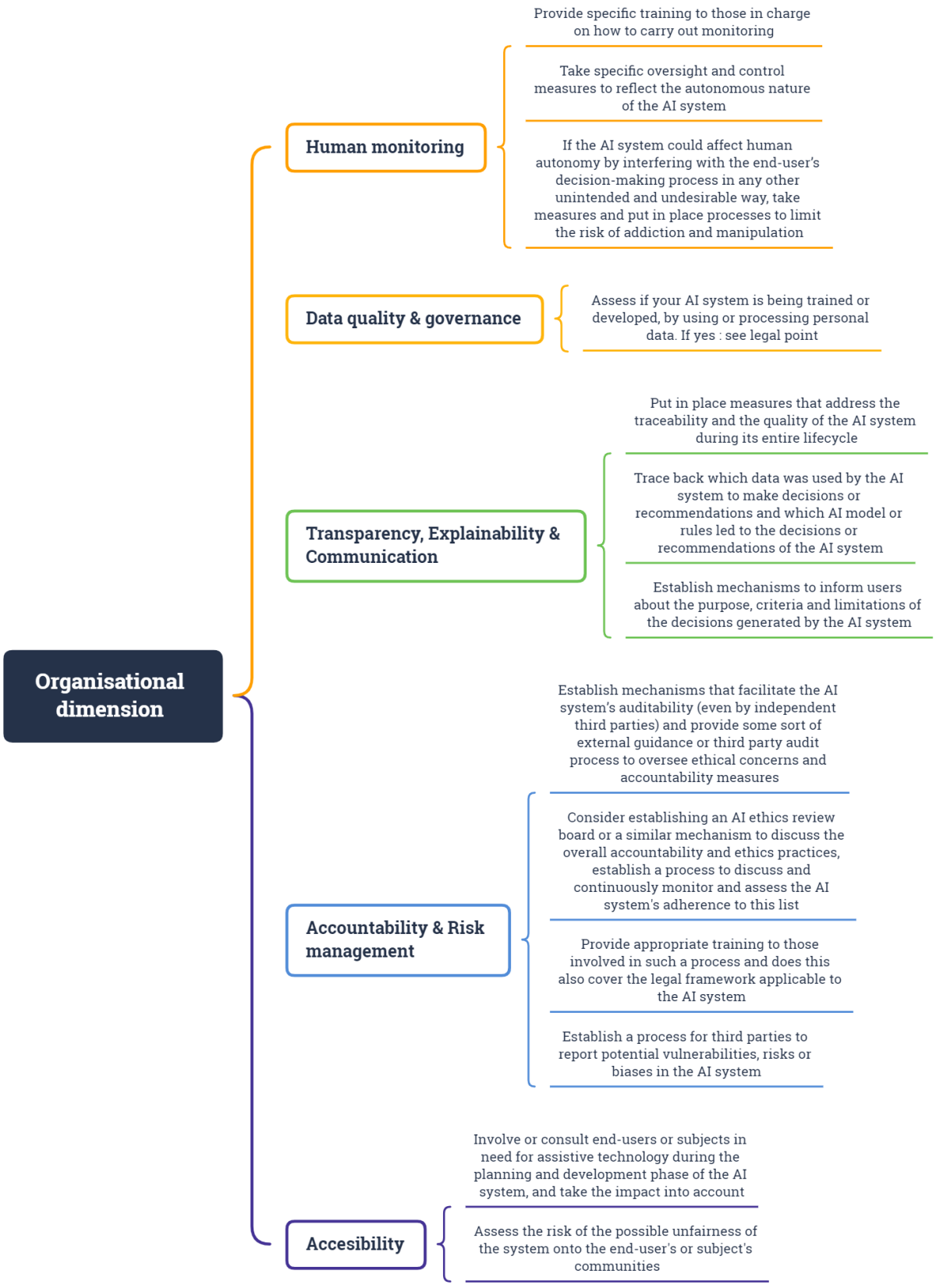



Figure 65: Organisational dimension for fairness

#### 7.4.1.3 Legal dimension

### Personal data protection

Figure below lists the main components of the legal dimension encompassed in the fairness criteria[28]. The main challenge from a legal perspective is notably the protection of personal data. Indeed, it is essential to verify whether personal data will be used to develop and train the artificial intelligence tool. It is also imperative to ensure that the result obtained by the AI could not have consequences on the privacy of the users or influence their behaviour. This verification is necessary because, if the GDPR applies, it is important to determine who is the data controller[29] (and the data processor[30] if applicable) and to put in place a data protection policy by design and by default. On this point, see D7.2.


Figure 66: Personal data protection and fairness.

#### 7.4.1.4 Technical dimension

Based on the guidelines from the High-Level Expert Group on AI[31], Figure below lists the main technical components to assess fairness in an AI tool or service. Indeed, the organisational and legal dimensions are complemented by a technical dimension. Some technical issues need to be considered before actually starting the development of an artificial intelligence tool. In addition to an organisational approach, the project manager must be aware of certain criteria that AI must meet in order to be considered fair. These are mainly the avoidance of unfair bias and to ensure the robustness and security of the IA tool and components. Additionally, the project manager must incorporate techniques to enable the traceability of the AI system and its reliability/reproducibility.

[28] This dimension is also reflected in the guidelines from the High-level Expert Group on AI: https://ec.europa.eu/digital-single-market/en/news/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment
[29] To determine who is the data controller, please see article 4.7 of the GDPR.
[30] To determine who is the data processor, please see article 4.8 of the GDPR.
[31] High-level Expert Group on Artificial Intelligence, “Assessment list for Trustworthy AI”, 2020. Available at: https://ec.europa.eu/digital-single-market/en/news/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment

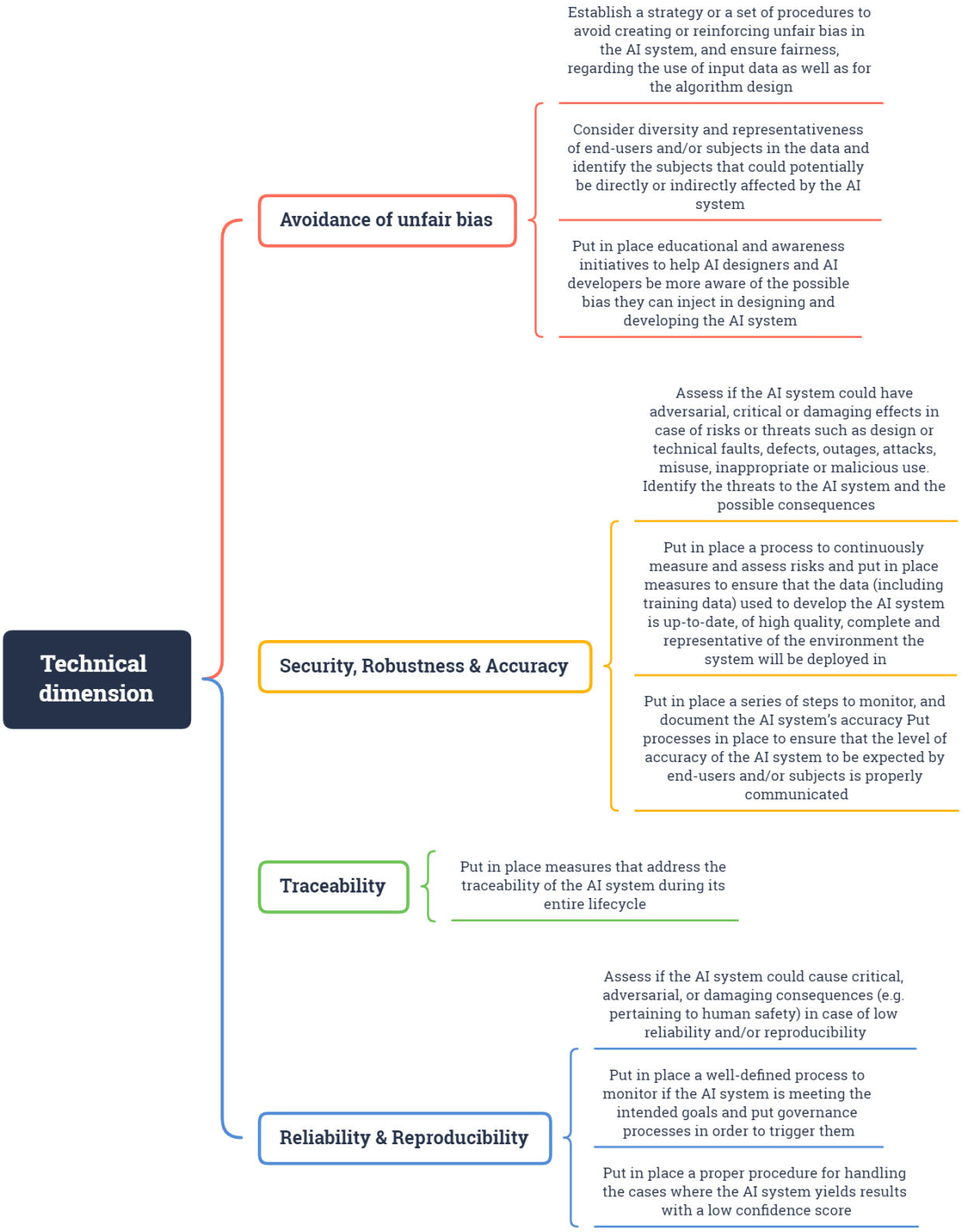



Figure 67: Technical components for fairness

### 7.4.2 *During AI development and operation/deployment*

#### 7.4.2.1 Introduction

The AI development, testing and deployment phase is clearly the part requiring the most important analysis and questioning to avoid creating a discriminatory tool. The following statements will help in verifying your compliance with the principle of fairness during this phase.

#### 7.4.2.2 Organisational dimension

Figure below lists the main components to assess fairness in AI from an organisational dimension during the deployment of the tool or service[32]. As is the case before any technical service is provided to deploy an artificial intelligence tool, fair development requires considering an organisational, legal and technical aspect. In this section, the criteria are the same as those to be considered before the technical realisation of an artificial intelligence tool. Indeed, we find the need for human monitoring, the transparency of the system, a risk management system and the accessibility of the AI system. Each of these categories is divided into several recommendations that will need to be technically and organisationally developed.

[32] This figure is also based on the guidelines from the High-level Expert Group on AI: https://ec.europa.eu/digital-single-market/en/news/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment

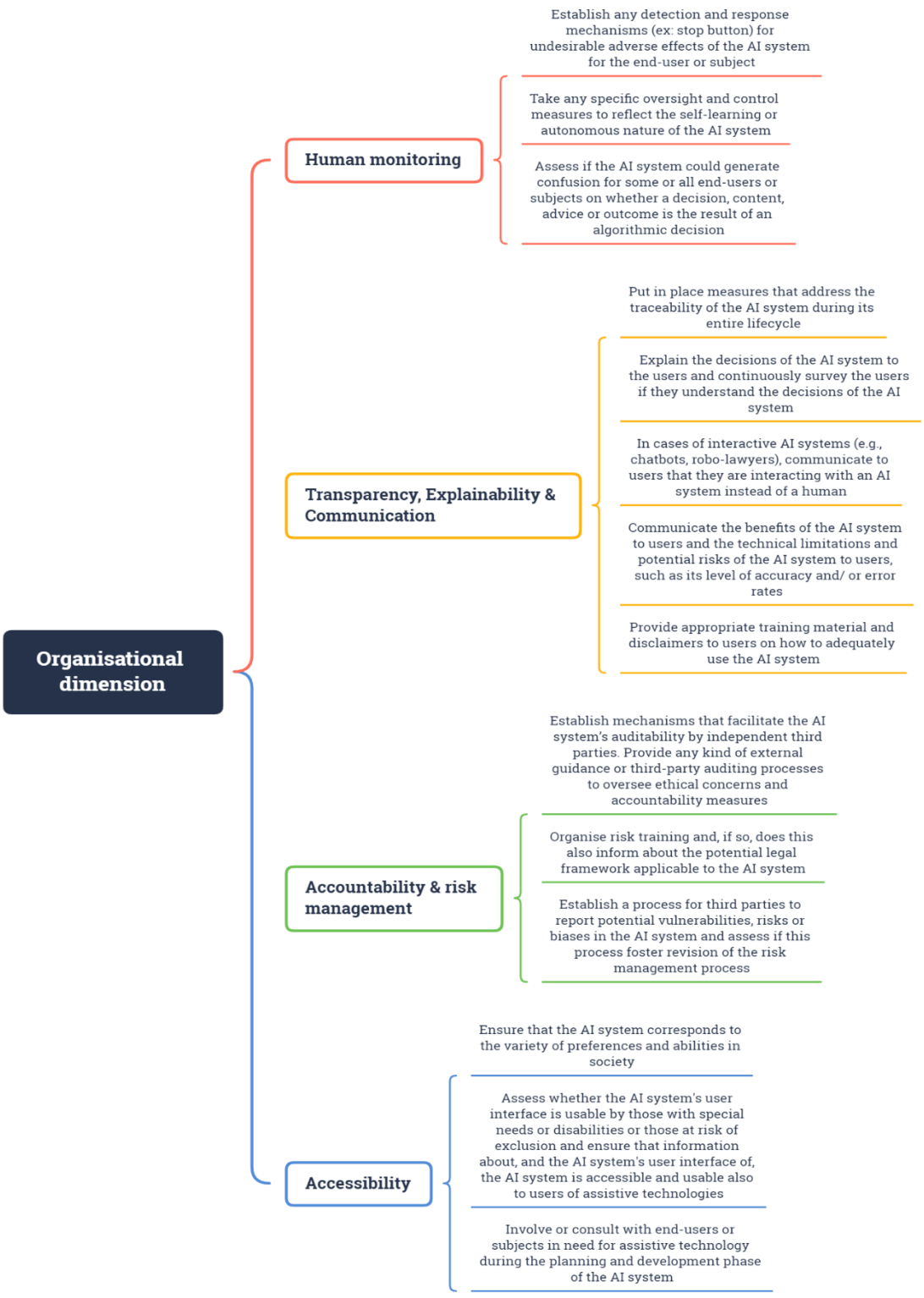



Figure 68: Organisational dimension for fairness during the deployment of AI

#### 7.4.2.3 Legal dimension

### Personal data protection

Based on the guidelines from the High-Level Expert Group on AI[33], Figure below completes the legal requirements in relation to those that must be in place before any IT development can begin.


Figure 69: Personal data protection for a fair IT deployment of AI

#### 7.4.2.4 Technical dimension

Based on the guidelines from the High-Level Expert Group on AI[34], Figure below completes the technical considerations in relation to those that must be in place before any IT development can begin. In this section, we focus on the technical implementation of the considerations studied in the preparatory phase (see Figure 66). Some technical components or functionalities of artificial intelligence need to be considered and studied before actually starting the development work of the artificial intelligence system. This section aims to list the recommendations of the High-level Expert Group on AI to effectively implement the preparatory work in this second phase.

[33] High-level Expert Group on Artificial Intelligence, "Assessment list for Trustworthy AI", 2020. Available at: https://ec.europa.eu/digital-single-market/en/news/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment .
[34] High-level Expert Group on Artificial Intelligence, "Assessment list for Trustworthy AI", 2020. Available at: https://ec.europa.eu/digital-single-market/en/news/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment

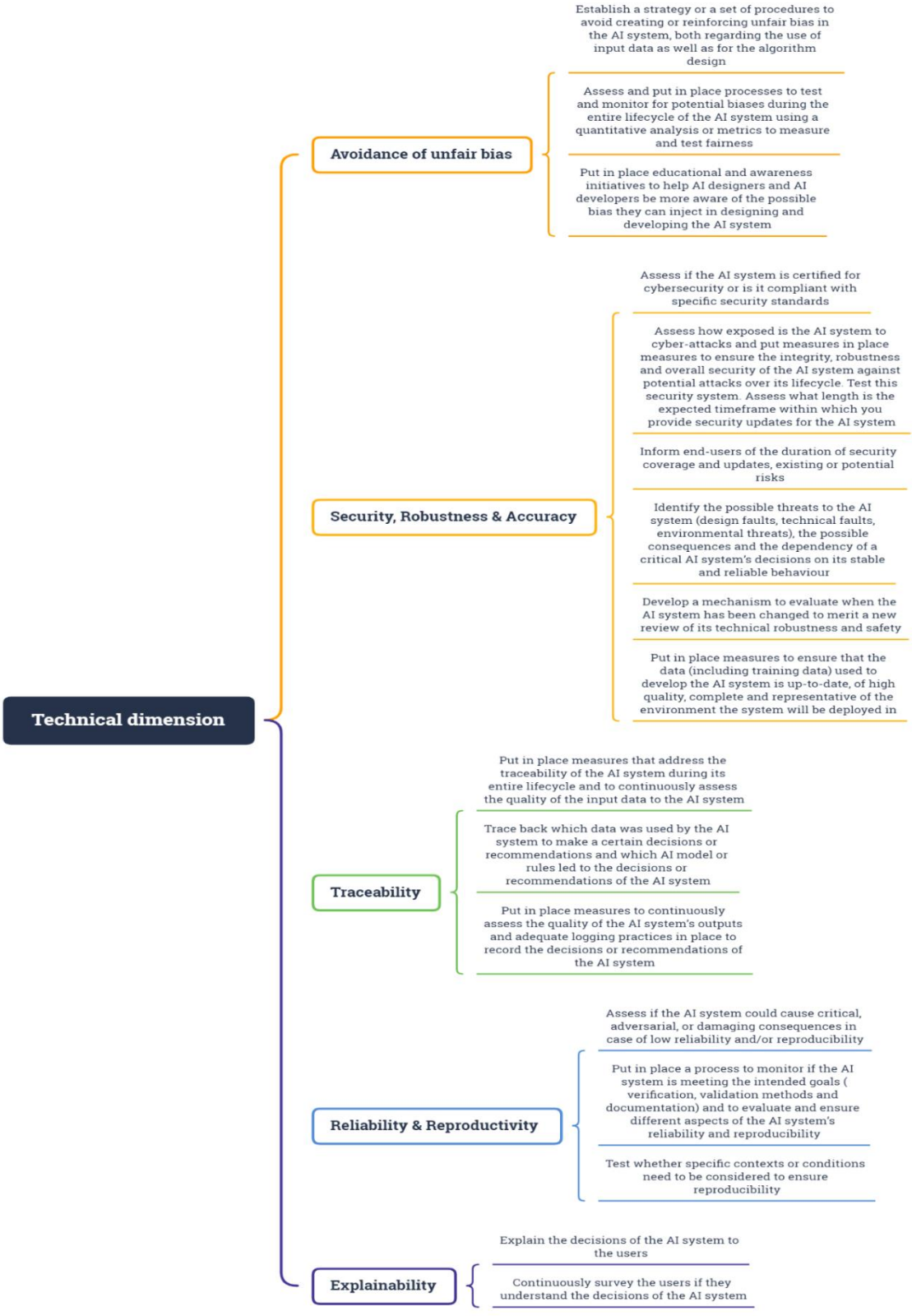



Figure 70: Technical dimension to ensure fairness during the IT deployment

### 7.4.3 *When a person withdraws from the service offered by AI or the project is over*

#### 7.4.3.1 Introduction

An AI project may well end, be abandoned, or not work. It is also important to think about the notion of fairness at this stage of the process. In addition, the question of personal data handling and deletion needs to be addressed.

#### 7.4.3.2 Organisational dimension

Figure 71 completes the timeline by considering the organisational dimension when a person no longer wants to benefit from the services or tools offered by AI or in case the project is over.


Figure 71: Organisational criteria at the end of the project or the service

It is also imperative, when processing personal data, that a policy for deletion or anonymisation of such an information is put in place. Indeed, when the purpose no longer justifies the retention of personal data or when the individual indicates his or her wish to no longer have his or her personal data retained and/or used, anonymisation or deletion must be guaranteed. Furthermore, the controller must notify the other controllers or his processor so that they also delete any copies of the personal data (Articles 5 € and 17 of the GDPR).

## 7.5 Linking legal and technical aspects for fairness

### 7.5.1 *Introduction*

This fifth section is devoted to technical implementations of the notion of fairness. In collaboration with THALES, based on the results of the D7.4 deliverable, we have identified the main ethical and legal components of the notion of fairness and identified potential algorithms that could meet these criteria.

### 7.5.2 *Results*

| Legal aspects | Technical aspects/Algorithms |
|---|---|
| Data quality:<br>Accuracy exact and kept up to date.<br>Avoidance of unfair bias[35]. | Fair Adversarial Network<br>Fair Random Forest<br>Model inspection tool to study fairness |
| Transparency:<br>Processed lawfully, fairly and in a transparent manner in relation to the data subject.<br>Explainability:<br>The questions refer to the ability to explain both the technical processes of the AI system and the reasoning behind the decisions or predictions that the AI system makes[36]. | Shapley Values<br>Decision trees of limited size as a surrogate model for a black box model |
| Accountability:<br>To be able to demonstrate compliance with the GDPR.<br>The principle of accountability necessitates that mechanisms be put in place to ensure responsibility for the development, deployment and/or use of AI systems"[37]. | |
| Reproducibility:<br>The state of or capacity for being reproductive[38]. | |
| Traceability:<br>The quality of having an origin or course of development that may be found or followed[39]. | |
| Security & Robustness[40]:<br>Secure and strong in constitution.<br>Resilience to attack.<br>General Safety. | Evasion attacks (Fast Gradient Sign Method, Basic Iterative method, Carlini and Wagner approach, Centered Initial Attack) |

[35] See the list of questions established by the High-Level Expert Group on AI: https://digital-strategy.ec.europa.eu/en/library/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment (p.16).
[36] https://digital-strategy.ec.europa.eu/en/library/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment (p.14).
[37] https://digital-strategy.ec.europa.eu/en/library/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment (p.21).
[38] See the list of questions established by the High-Level Expert Group on AI: https://digital-strategy.ec.europa.eu/en/library/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment (p.11).
[39] According to the High-Level Expert Group on AI traceability must be understood as a self-assessment "whether the processes of the development of the AI system, i.e. the data and processes that yield the AI system's decisions, is properly documented to allow for traceability, increase transparency and, ultimately, build trust in AI in society"; https://digital-strategy.ec.europa.eu/en/library/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment (p.14).
[40] See the list of questions established by the High-Level Expert Group on AI: https://digital-strategy.ec.europa.eu/en/library/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment (p. 9-10).

| Legal aspects | Technical aspects/Algorithms |
|---|---|
| | Defence mechanisms (Adversarial Training, Dimensionality Reduction, Prediction Similarity, Feature scattering, Double-ended monitoring for adversarial detection) |
| Reliability:<br>Being trustworthy or performing consistently well[41]. | |
| Predictability:<br>Always behaving or occurring in the way expected. | |
| Confidentiality & Integrity:<br>Processed in a manner that ensures appropriate security of the personal data, including protection against unauthorized or unlawful processing and against accidental loss, destruction or damage, using appropriate technical or organizational measures. | |

Table 22: Results Linking legal and technical aspects for fairness.

## 7.6 A step forward: The Artificial Intelligence Act

In April 2021, the European Commission releases a proposal to regulate the AI activities at the European level[42].

The European Commission insists on the need for a legal framework to respect the fundamental rights of European citizens. In order to ensure a high level of protection of these rights, the proposal adopts, like the GDPR, a risk-based approach and accountability of all stakeholders[43]. Furthermore, the Commission hopes that an ex-ante testing, risk management and human oversight procedure will minimise the risk of bias in the results rendered by artificial intelligence tools.

Additionally, the Commission recognises that, if the proposal impacts on the freedom to conduct business and the freedom of art and science, this is justified for imperious reasons such as health, safety or consumer protection.

The European Commission is creating a real transparency duty and foresees that this will not infringe intellectual property rights as it will be limited to the information necessary to allow effective redress for individuals and control by the authorities. Confidentiality obligations are also put in place to ensure compliance with Directive 2016/943 on the protection of know-how and commercial information[44].

Hereafter, we draw attention to certain provisions that may have an impact on the notion of fairness by providing, for example, access to certain information or by insisting on a transparency requirement by the service provider.

[41] See the list of questions established by the High-Level Expert Group on AI: https://digital-strategy.ec.europa.eu/en/library/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment (p.11).

[42] Proposal for a Regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act), 21.4.2021, COM(2021) 206 final. Available at: https://ec.europa.eu/newsroom/dae/items/709090. We draw the attention to the fact that this document is, for the moment, at the stage of a proposal and might be subject to modifications.

[43] See the definitions contained in the Proposal for the notion of "importer" (article 3.6 of the Proposal), "distributor" (article 3.7 of the Proposal) and "operator" (article 3.8 of the Proposal).

[44] Explanatory Memorandum of the Artificial Intelligence Act, p.11.

As a preliminary remark, two definitions are crucial:

- "'**artificial intelligence system'** (AI system) means software that is developed with one or more of the techniques and approaches listed in Annex I and can, for a given set of human-defined objectives, generate outputs such as content, predictions, recommendations, or decisions influencing the environments they interact with"[45]
- "**provider'** means a natural or legal person, public authority, agency or other body that develops an AI system or that has an AI system developed with a view to placing it on the market or putting it into service under its own name or trademark, whether for payment or free of charge"[46]

Even if the criteria "fair" does not appear as such in the text of the proposal from the European Commission, we highlight some proposed provisions that are important for the notion of fairness:

a) Prohibition of certain practices

The proposal of Article 5 listing the artificial intelligence activities that should be prohibited[47].

b) High risk AI system

Several requirements are set to ensure the legality of artificial intelligence tools considered high risk. First, the proposal states that a risk management system must be put in place.

> Article 9.2 of the Proposal:
>
> *"The risk management system shall consist of a continuous iterative process run throughout the entire lifecycle of a high-risk AI system, requiring regular systematic updating. It shall comprise the following steps: EN 47 EN (a) identification and analysis of the known and foreseeable risks associated with each high-risk AI system; (b) estimation and evaluation of the risks that may emerge when the high-risk AI system is used in accordance with its intended purpose and under conditions of reasonably foreseeable misuse; (c) evaluation of other possibly arising risks based on the analysis of data gathered from the post-market monitoring system referred to in Article 61; (d) adoption of suitable risk management measures in accordance with the provisions of the following paragraphs."*

Secondly, a governance policy for training data[48], validation data[49] and testing data[50] must be put in place. This governance must itself meet certain conditions to be valid (article 10 of the Proposal).

Thirdly, a technical documentation shall be established before is AI system is placed on the market or enter into service (Article 11 of the Proposal).

Fourth, the European Commission's proposal provides for a record-keeping system (Article 12 of the Proposal)[51] .

[45] Article 3.1 of the Proposal.

[46] Article 3.2 of the Proposal.

[48] See the definition in article 3.29 of the Proposal.

[49] See the definition in article 3.30 of the Proposal.

[50] See the definition in article 3.31 of the Proposal.

[51] According to Article 12 of the Proposal; "For high-risk AI systems referred to in paragraph 1, point (a) of Annex III, the logging capabilities shall provide, at a minimum: (a) recording of the period of each use of the system (start date and time and end date and time of each use); (b) the reference database against which input data has been checked by the system; (c) the input data for which the search has led to a match; (d) the identification of the natural persons involved in the verification of the results, as referred to in Article 14 (3)".

Fifth, a particular requirement for transparency to users is envisaged. The main objective is to enable users to understand and interpret the results provided by the artificial intelligence tool. In addition, instructions to assist the user must be provided. The European Commission foresees the list of information to be provided (Article 13 of the Proposal).

Finally, we find several criteria already mentioned by the High-level Expert Group on AI: human oversight[52], accuracy, robustness and cybersecurity[53].

c) Transparency obligations for certain AI systems

Because of the sensitivity of these techniques, the European Commission insists on and reinforces transparency for certain artificial intelligence tools. In particular, individuals must be informed that they are interacting with an artificial intelligence tool or that they are exposed to an "emotion recognition tool"[54] or a "biometric categorisation system"[55]. An increased transparency is required to combat *deep fakes* in order to alert individuals that content has been artificially generated or manipulated (Article 52 of the Proposal)[56].

[52] Article 14 of the proposal.

[53] Article 15 of the proposal.

[54] See the definition in article 3.34 of the Proposal.

[55] See the definition in article 3.35 of the Proposal.

[56] In particular, Article 64 may be of major importance as it provides access to data and various information by the service provider to the market surveillance authority. Proposal of article 64.1: "Access to data and documentation in the context of their activities, the market surveillance authorities shall be granted full access to the training, validation and testing datasets used by the provider, including through application programming interfaces ('API') or other appropriate technical means and tools enabling remote access." The notion of "market surveillance authority" is defined in the article 3.26 of the Proposal.

# Chapter 8 Summary and Conclusion

This document detailed the approaches implemented in the final demonstrator of SPARTA WP7 SAFAIR program. The approaches cover three of the identified challenges: security and robustness (Chapter 2), explainability (Chapter 3) and fairness (Chapter 4).

About ***security and robustness***, we implemented several attacks and defence mechanisms. We proposed in the demonstrator classical methods of evasion attack, as well as original ones. The techniques were applied for two different use cases, face reidentification and PDF malware detection. The results of a contest organized to evaluate some evasion attacks and the defence strategies adopted by the participants were also presented in Chapter 5.

Here are a few takeaways on the mechanisms to achieve security and robustness:

- Applying a sequence of specific preprocessing steps to the data can significantly improve the robustness of a model to adversarial attacks (Section 2.7). The results suggest that this approach can be used if a moderate performance loss is acceptable.
- Adversarial Training can significantly improve the robustness of a model to adversarial attacks, including in a setting where the target is not known in advance (Section 2.8). The results suggest that using a mix of different attacks in the training set produces a more robust defence.

Concerning ***explainability and fairness***, we presented a component based on ShapKit, a Python module dedicated to local explanation of machine learning models described in D7.4. We then illustrated the use of ShapKit on a case dedicated to denial of service attack detection. We also presented some supplemental explorations of surrogate-type methods for explainable artificial intelligence. We also described a tool that is dedicated to both interpretability and fairness inspection. We also presented its usage on an example.

Here are a few takeaways on the mechanisms to achieve explainability and fairness:

- Surrogate-explanation is a promising, model-agnostic concept for addressing explainability (Section 3.2). The results suggest however to use an ensemble of model-agnostic techniques to provide optimal insights on the outputs of a given model.
- A combination of distribution stressing and feature importance can be used to design a model-agnostic method for estimating fairness (Section 4.1). The results suggest that this method may be effectively used to evaluate and eventually improve a model's fairness.

Finally, this document was extended with a report on additional progresses done by the SPARTA team. SAFAIR AI Threat model and Knowledge Base of SPARTA was extended and improved to capture the new results from ENISA and other relevant initiatives on AI threat landscape (Chapter 6). Also, we presented the legal aspects of AI, a practical checklist for software developers in order to respect the equity criteria throughout the development process (Chapter 7). The link between the different elements of the fairness principle and the algorithms proposed by the partners in the current deliverable was established.

# Chapter 9 List of Abbreviations

| Abbreviation | Translation |
|---|---|
| DPIA | Data Protection Impact Assessment |
| DPO | Data Protection Officer |
| EC | European Commission |
| EU | European Union |
| GDPR | General Data Protection Regulation |
| DoS | Denial of Service |
| AI | Artificial Intelligence |
| ML | Machine Learning |
| TCP | Transmission Control Protocol |
| ReLu | Rectified Linear Unit |
| ICMP | Internet Control Message Protocol |
| IP | Internet Protocol |
| 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 |
| CIA | Centered Initial Attack |
| SVM | Support Vector Machine |
| AI | Artificial Intelligence |
| ML | Machine Learning |
| NN | Neural Network |

# Chapter 10 Bibliography

[1] Z. Liu, P. Luo, X. Wang, et X. Tang, « Deep Learning Face Attributes in the Wild », nov. 2014.

[2] Z. Liu, P. Luo, X. Wang, et X. Tang, « Large-scale CelebFaces Attributes (CelebA) Dataset », *Multimedia Laboratory, The Chinese University of Hong Kong*, 2016.

[3] S. Srinivas, R. K. Sarvadevabhatla, K. R. Mopuri, N. Prabhu, S. S. S. Kruthiventi, et R. V. Babu, « An Introduction to Deep Convolutional Neural Nets for Computer Vision », in *Deep Learning for Medical Image Analysis*, Elsevier, 2017, p. 25-52. doi: 10.1016/B978-0-12-810408-8.00003-1.

[4] A. Anwar, « Difference between AlexNet, VGGNet, ResNet, and Inception », *Towards Data Science*, 2019.

[5] C. Szegedy *et al.*, « Going Deeper with Convolutions », sept. 2014, doi: https://arxiv.org/abs/1409.4842.

[6] K. He, X. Zhang, S. Ren, et J. Sun, « Deep residual learning for image recognition », in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, p. 770-778.

[7] A. Krizhevsky, I. Sutskever, et G. E. Hinton, « ImageNet classification with deep convolutional neural networks », *Commun. ACM*, vol. 60, nº 6, p. 84-90, mai 2017, doi: 10.1145/3065386.

[8] K. Simonyan et A. Zisserman, « Very Deep Convolutional Networks for Large-Scale Image Recognition », sept. 2014.

[9] N. C. Thompson, K. Greenewald, K. Lee, et G. F. Manso, « The Computational Limits of Deep Learning », juill. 2020.

[10] A. Kaya, A. S. Keceli, C. Catal, H. Y. Yalic, H. Temucin, et B. Tekinerdogan, « Analysis of transfer learning for deep neural network based plant classification models », *Comput. Electron. Agric.*, vol. 158, p. 20-29, mars 2019, doi: 10.1016/j.compag.2019.01.041.

[11] O. M. Parkhi, A. Vedaldi, et A. Zisserman, « Deep Face Recognition », in *Procedings of the British Machine Vision Conference 2015*, 2015, p. 41.1-41.12. doi: 10.5244/C.29.41.

[12] « ResNet-50; ResNet-50 Pre-trained Model for Keras », 2017. https://www.kaggle.com/keras/resnet50

[13] K. Zhang, Z. Zhang, Z. Li, et Y. Qiao, « Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks », avr. 2016, doi: 10.1109/LSP.2016.2603342.

[14] J. Du, « High-Precision Portrait Classification Based on MTCNN and Its Application on Similarity Judgement », *J. Phys. Conf. Ser.*, vol. 1518, p. 12066, 2020, doi: 10.1088/1742-6596/1518/1/012066.

[15] J. Xiang et G. Zhu, « Joint Face Detection and Facial Expression Recognition with MTCNN », in *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*, juill. 2017, p. 424-427. doi: 10.1109/ICISCE.2017.95.

[16] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, et A. Vladu, « Towards Deep Learning Models Resistant to Adversarial Attacks », juin 2017.

[17] S. Qiu, Q. Liu, S. Zhou, et C. Wu, « Review of Artificial Intelligence Adversarial Attack and Defense Technologies », *Appl. Sci.*, vol. 9, nº 5, p. 909, mars 2019, doi: 10.3390/app9050909.

[18] T. Bai, J. Luo, J. Zhao, B. Wen, et Q. Wang, « Recent Advances in Adversarial Training for Adversarial Robustness », févr. 2021.

[19] G. K. Dziugaite, Z. Ghahramani, et D. M. Roy, « A study of the effect of JPG compression on adversarial images », août 2016.

[20] N. Das *et al.*, « Keeping the Bad Guys Out: Protecting and Vaccinating Deep Learning with JPEG Compression », mai 2017.

[21] V. Zantedeschi, M.-I. Nicolae, et A. Rawat, « Efficient Defenses Against Adversarial Attacks », juill. 2017.

[22] W. Xu, D. Evans, et Y. Qi, « Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks », avr. 2017, doi: 10.14722/ndss.2018.23198.

[23] C. Guo, M. Rana, M. Cisse, et L. van der Maaten, « Countering Adversarial Images using Input Transformations », oct. 2017.

[24] B. Ahn, Y. Kim, G. Park, et N. I. Cho, « Block-Matching Convolutional Neural Network (BMCNN): Improving CNN-Based Denoising by Block-Matched Inputs », in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, nov. 2018, p. 516-525. doi: 10.23919/APSIPA.2018.8659548.

[25] A. Creswell et A. A. Bharath, « Denoising Adversarial Autoencoders », mars 2017.

[26] J. Xu, L. Zhang, W. Zuo, D. Zhang, et X. Feng, « Patch Group Based Nonlocal Self-Similarity Prior Learning for Image Denoising », in *2015 IEEE International Conference on Computer Vision (ICCV)*, déc. 2015, p. 244-252. doi: 10.1109/ICCV.2015.36.

[27] K. Zhang, W. Zuo, Y. Chen, D. Meng, et L. Zhang, « Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising », août 2016, doi: 10.1109/TIP.2017.2662206.

[28] A. Blanco-Justicia et J. Domingo-Ferrer, « Machine Learning Explainability Through Comprehensible Decision Trees », in *Machine Learning and Knowledge Extraction*, Springer International Publishing, 2019, p. 15-26.

[29] T. Laugel, X. Renard, M.-J. Lesot, C. Marsala, et M. Detyniecki, « Defining Locality for Surrogates in Post-hoc Interpretablity », juin 2018, doi: abs/1806.07498.

[30] M. Szczepanski, M. Choraś, M. Pawlicki, et R. Kozik, « Achieving Explainability of Intrusion Detection System by Hybrid Oracle-Explainer Approach », 2020.

[31] M. Ribeiro, S. Singh, et C. Guestrin, « Why Should {I} Trust You?: Explaining the Predictions of Any Classifier », 2016.

[32] N. Zhang, A. Gupta, C. Kauten, A. V. Deokar, et X. Qin, « Detecting fake news for reducing misinformation risks using analytics approaches », *Eur. J. Oper. Res.*, vol. 279, nº 3, p. 1036-1052, 2019.

[33] M. Ribeiro, S. Sing, et C. Guestrin, « Anchors: High-Precision Model-Agnostic Explanations », 2018.

[34] A. Richardson et A. Rosenfeld, « A survey of interpretability and explainability in human-agent systems », 2018.

[35] T. Parr et P. Grover, « How to visualize decision trees », *Explained.ai*. https://explained.ai/decision-tree-viz/index.html.

[36] « Plotly JavaScript Open Source Graphing Library », *plotly*. https://plotly.com/javascript/

[37] L. Hulstaert, « Understanding model predictions with LIME », *Towards Data Science*, 2018.

[38] M. Pawlicki, M. Choraś, R. Kozik, et W. Hołubowicz, « On the Impact of Network Data Balancing in Cybersecurity Applications », *Comput. Sci. – ICCS 202020th Int. Conf. Amst. Neth. June 3–5 2020 Proc. Part IV*, vol. 12140, p. 196—210, mai 2020.

[39] A. Hirabayashi et L. Condat, « Torwards a general formulation for over-sampling and under-sampling », 2007.

[40] NK. W. Bowyer, N. V. Chavla, L. O. Hall, et W. P. Kegelmeyer, « SMOTE: Synthetic Minority Over-sampling Technique », *J Artif Int Res*, vol. 16, nº 1, p. 321-357, 2002.

[41] T. Greene, « AI Now: Predictive policing systems are flawed because they replicate and amplify racism », *TNW News*, 2020.

[42] S. Garcia, A. Parmisano, et M. J. Erquiaga, « IoT-23: A labeled dataset with malicious and benign IoT network traffic (Version 1.0.0) [Data set] », *Zenodo*, 2020.

[43] G. E. A. P. A. Batista, A. L. C. Bazzan, et M. C. Monard, « Balancing training data for automated annotation of keywords: a case study », in *Proceedings of the Second Brazilian Workshop on Bioinformatics*, 2003, p. 35-43.

[44] G. L. Ciampaglia, « Fighting fake news: a role for computational social science in the fight against digital misinformation », *J. Comput. Soc. Sci.*, vol. 1, nº 1, p. 147-153, 2018.

[45] R. Goldman, « Reading Fake News, Pakistani Minister Directs Nuclear Threat at Israe », *The New York Times*, 2016.

[46] T. Quandt, L. Frischlich, S. Boberg, et T. Schatto-Eckrodt, « Fake News », in *The International Encyclopedia of Journalism Studies*, Wiley, 2019, p. 1-6. doi: 10.1002/9781118841570.iejs0128.

[47] E. C. Tandoc, Z. W. Lim, et R. Ling, « Defining “Fake News” », *Digit. Journal.*, vol. 6, nº 2, p. 137-153, févr. 2018, doi: 10.1080/21670811.2017.1360143.

[48] H. Allcott et M. Gentzkow, « Social Media and Fake News in the 2016 Election », *J. Econ. Perspect.*, vol. 31, nº 2, p. 211-236, mai 2017, doi: 10.1257/jep.31.2.211.

[49] N. K. Conroy, V. L. Rubin, et Y. Chen, « Automatic deception detection: Methods for finding fake news », *Proc. Assoc. Inf. Sci. Technol.*, vol. 52, nº 1, p. 1-4, janv. 2015, doi: 10.1002/pra2.2015.145052010082.

[50] J. Devlin, M.-W. Chang, K. Lee, et K. Toutanova, « No Title », in *Proceedings of the 2019 Conference of the North*, 2019, p. 4171-4186. doi: 10.18653/v1/N19-1423.

[51] R. Horev, « BERT Explained: State of the art language model for NLP », *Towards Data Science*, 2018.

[52] H. Jwa, D. Oh, K. Park, J. M. Kang, et H. Lim, « exBAKE: Automatic Fake News Detection Model Based on Bidirectional Encoder Representations from Transformers (BERT) », *Applied Sciences* , vol. 9, nº 19. 2019. doi: 10.3390/app9194062.

[53] C. Bisaillon, « Kaggle: Fake and real news dataset [dataset] », 2020. https://www.kaggle.com/clmentbisaillon/fake-and-real-news-dataset

[54] H. Ahmed, I. Traore, et S. Saad, « Detecting opinion spams and fake news using text classification », *Secur. Priv.*, vol. 1, nº 1, p. e9, janv. 2018, doi: 10.1002/spy2.9.

[55] V. Sanh, L. Debut, J. Chaumond, et T. Wolf, « DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter », oct. 2019.

[56] Explosion, « Spacy: Industrial-Strength Natural Language Processing in Python », 2016.

[57] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and Harnessing Adversarial Examples. CoRR abs/1412.6572 (2014). http://arxiv.org/abs/1412.6572.

[58] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial examples in the physical world, arXiv preprint arXiv:1607.02533 (2016)

[59] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In Security and Privacy (SP), 2017 IEEE Symposium on. IEEE, 39–57.

[60] Uri Shaham, Yutaro Yamada, and Sahand Negahban. 2015. Understanding Adversarial Training: Increasing Local Stability of Neural Nets through Robust Optimization. CoRR abs/1511.05432 (2015).

[61] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus, Intriguing properties of neural networks. CoRR abs/1312.6199 (2013). http://arxiv.org/abs/1312.6199

[62] B. Addad, J. Kodjabachian and C. Meyer, Clipping free attacks against Artificial Neural Networks, https://arxiv.org/abs/1803.09468, 2017.

[63] Haichao Zhang, Jianyu Wang, Defense Against Adversarial Attacks Using Feature Scattering-based Adversarial Training, https://arxiv.org/abs/1907.10764, 2019.

[64] Contagio Dump (2013). Contagio: Malware dump. http://contagiodump.blogspot.fr/2013/03/16800-clean-and-11960-malicious-files.

[65] Ateniese, G., Felici, G., Mancini, L. V., Spognardi, A., Villani, A., and Vitali, D. (2013). Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. CoRR.

[66] Biggio, B., Corona, I., Maiorca, D., Nelson, B., Srndi ˇ c, N., ´ Laskov, P., Giacinto, G., and Roli, F. (2013). Evasion Attacks against Machine Learning at Test Time.

[67] Borg, K. (2013). Real time detection and analysis of pdf files. Master's thesis.

[68] Alexander Jordan, Francois Gauthier, Behnaz Hassanshahi, and David Zhao, SAFE-PDF: Robust Detection of JavaScript PDF Malware With Abstract Interpretation,

[69] D. Maiorca, G. Giacinto, and I. Corona. A pattern recognition system for malicious PDF files detection. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7376 LNAI:510–524, 2012.

[70] Bonan Cuan, Aliénor Damien, Claire Delaplace, Mathieu Valois. Malware Detection in PDF Files Using Machine Learning. SECRYPT 2018 - 15th International Conference on Security and Cryptography, Jul 2018, Porto, Portugal. 8p.

[71] CVEDetails (2017). Adobe vulnerabilities statistics. https://www.cvedetails.com/product/497/Adobe-Acrobat-Reader.html.

[72] Kittilsen, J. (2011). Detecting malicious pdf documents. Master's thesis.

[73] Stevens, D. (2006), Didier stevens blog. https://blog. didierstevens.com/.

[74] Torres, J. and De Los Santos, J. (2018). Malicious pdf documents detection using machine learning techniques.

[75] Demontis, A., Melis, M., Pintor, M., Jagielski, M., Biggio, B., Oprea, A., Nita-Rotaru, C. and Roli, F., 2019. Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks. In 28th Usenix Security Symposium, Santa Clara, California, USA.

[76] Aas, Jullum and Lland. Explaining individual predictions when features are dependent:More accurate approximations to Shapley values. 2019

[77] Belsley and Welsch, 'Regression diagnostics: Identifying Influential Data and Sources of Collinearity', Wiley, 1980. 244-261.

[78] Fisher, R.A. "The use of multiple measurements in taxonomic problems" Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contributions to Mathematical Statistics" (John Wiley, NY, 1950).

[79] Grah and Thouvenot. A Projected SGD algorithm for estimating Shapley Value applied in attribute importance. Machine Learning and Knowledge Extraction (pp.97-115), 2020.

[80] Lundberg, Scott M., and Su-In Lee. A unified approach to interpreting model predictions. Advances in NeuralInformation Processing Systems. 2017.

[81] Merrick and Taly. The Explanation Game: Explaining Machine Learning Models with CooperativeGame Theory. 2019.

[82] Shapley. A value for n-person games. In Contributions to the Theory of Games. 2.28 (1953), pp. 307- 317

[83] W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology, volume 1905, pages 861-870, San Jose, CA, 1993

[84] Strumbelj and Kononenko. Explaining prediction models and individual predictions with feature con-tributions. Knowledge and information systems 41.3, 647-665, 2014.

[85] Bachoc, Gamboa, Halford, Loubes and Risser, Explaining Machine Learning Models using Entropic Variable Projection, 2020

[87] 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.