10

Companies developing compliance with

# Guide 10. Robustness

## European Artificial Intelligence Act

Financiado por la Unión Europea
NextGenerationEU

GOBIERNO DE ESPAÑA
MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL

Plan de Recuperación, Transformación y Resiliencia

España | digital 2026

This guide has been developed within the framework of the development of the Spanish pilot for the regulatory AI Sandbox, through collaboration among participants, technical assistance providers, potential competent national authorities, and the sandbox's expert advisory group.

The aim of the guide is to serve as an introductory support to the European Regulation on Artificial Intelligence and its applicable obligations. Although **it is not legally binding and does not replace or develop the applicable legislation, it provides practical recommendations** aligned with regulatory requirements, pending the approval of the harmonised implementing standards for all Member States.

This document **is subject to an ongoing process of evaluation and review, with periodic updates** in line with the development of standards and the various guidelines published by the European Commission, and it will be updated once the Digital Omnibus amending the Artificial Intelligence Act is approved.

Among the relevant technical references currently under development, particular emphasis is placed on **prEN 18229-2 "AI Trustworthiness Framework – Part 2: Accuracy and Robustness"**, which will constitute the foundational standard for the assessment of accuracy and robustness in AI systems, once formally adopted as a harmonized standard within the framework of compliance with the European Artificial Intelligence Act.

**Revision date:** 10 December 2025

# General Content

1. Preamble ..... 5
2. Introduction ..... 9
3 European Regulation on Artificial Intelligence ..... 13
4. How to approach the requirements? ..... 16
5. Technical documentation ..... 35
6. Self-assessment questionnaire ..... 37
7. Annexes ..... 38
8. References, Standards and Norms ..... 57

# Detailed Index

1. Preamble ..... 5
1.1 Purpose of the document ..... 5
1.2 How to read this guide? ..... 6
1.3 Who is it for? ..... 7
1.4 Use cases used in the guide ..... 8
2. Introduction ..... 9
2.1 What is Robustness for AI? ..... 9
3. European Regulation on Artificial Intelligence ..... 13
3.1 Previous analysis and relationship of the articles ..... 13
3.2 Content of the articles in the AI Act ..... 14
3.3 Correspondence of the articles with the sections of the guide ..... 15
4. How to approach the requirements? ..... 16
4.1 Assessment of robustness ..... 16
4.1.1 Life cycle ..... 18
4.1.2 Selecting Metrics ..... 19
4.1.3 Validation and Verification ..... 20
4.1.4 Efficiency in AI systems ..... 21
4.1.5 Performance in AI systems ..... 22
4.1.6 Robustness monitoring ..... 24
4.2 Robustness and Resistance to Errors ..... 25
4.3 Redundancy and backup plans ..... 28
4.4 Robustness for systems that continue to learn ..... 30
4.4.1 Types of AI system degradation and mitigation plans, strategies and tools ..... 31
4.4.2 Strategies to mitigate changes in degradation of model and/or data accuracy and robustness ..... 32
5. Technical documentation ..... 35
5.1 Certification of model robustness ..... 36
6. Self-assessment questionnaire ..... 37
7. Annexes ..... 38
7.1 Annex I: Metrics and Tests for AI systems ..... 38
7.1.1 Metrics to Establish Robustness ..... 38
7.1.2 Test Methods ..... 39

7.2 Annex II: Robustness and uncertainty ..... 42
7.2.1 Robustness as a capacity to deal with and minimise uncertainty ..... 43
7.2.2 Methods for measuring and quantifying the uncertainty of the output of an AI model ... 43
7.3 Glossary ..... 44
7.4 Appendix: Critical Translations (English–Spanish) ..... 57
8. References, Standards and Norms ..... 58
8.1. References ..... 58
8.2 Standards and Norms ..... 65

# 1. Preamble

## 1.1 Purpose of the document

A high-risk AI system must be prepared to minimize and prevent harmful and undesirable behaviours and be able to detect when its operation takes place outside the domain of entry and execution established by its intended purpose. Likewise, it should be designed and implemented to avoid making wrong decisions or generating erroneous outputs information, in order to prevent adverse consequences for safety and fundamental right.

The objective of this guide is to establish the necessary measures that, in our opinion, serve to meet the requirements of the European Regulation on Artificial Intelligence, in relation to the robustness of the high-risk system.

The European Regulation on Artificial Intelligence considers the concept of **technical robustness** as a **key mechanism** for high-risk AI systems. This is developed through its recital (75), where it establishes that:

**AI Act**

Recital (75)

> Technical robustness is a key requirement for high-risk AI systems. They should be resilient in relation to harmful or otherwise undesirable behaviour that may result from limitations within the systems or the environment in which the systems operate (e.g. errors, faults, inconsistencies, unexpected situations).

The measures proposed in this guide establish that the AI system must consider in its design, development and validation, the technical and organisational measures to prevent such situations, including the possibility that when such technical robustness does not operate within safe parameters, the system may interrupt its operation. So much so that it establishes it in the same recital (75):

**AI Act**

Recital (75)

> […] high-risk AI systems, for example by designing and developing appropriate technical solutions to prevent or minimise harmful or otherwise undesirable behaviour. Those technical solution may include for instance mechanisms enabling the system to safely interrupt its operation (fail-safe plans) in the presence of certain anomalies or when operation takes place outside certain predetermined boundaries. […]

In other words, the robustness of the AI system is so relevant that in cases where it cannot be adequately guaranteed, the system must be prepared to **interrupt** its operation in a **controlled manner.**

It is the responsibility of the provider of the high-risk AI system to take appropriate measures (both organizational and technical) to ensure that the system's robustness requirements are met. Likewise, within its scope of application, the deployer of the system also has responsibilities that will materialize in specific measures (again organizational and technical).

We will develop throughout the guide a series of organizational and technical measures aimed first at validating the artificial intelligence system and then applying the quality controls of the model that help verify and validate the reasons that lead to the selection of such metrics.

As for complementary aspects that are key to implementing the system for its intended purpose, these metrics go beyond the immediate model output for a given evaluation benchmark and may have implications related to discrimination, bias, or inaccuracy.

Throughout the guide we will address the relationship with these complementary aspects, and their inclusion in the concept of robustness.

## 1.2 How to read this guide?

This guide is closely linked to the Article 15 guide on accuracy, robustness and cybersecurity.

The robustness of the AI system has, among other objectives, that of preserving the accuracy obtained during the training, testing and validation of the system throughout its life cycle; especially in those inputs that are not in the training and validation set, but that are within the domain of the AI system.

The relationship with cybersecurity is also close. On the one hand, cybersecurity contributes to the robustness of the system by protecting it against adversarial attacks, which could alter its response (and therefore its accuracy). On the other hand, the robustness mechanisms must guarantee that cybersecurity measures are maintained over time.

For a correct application of this guide, it has been structured following the article of the European Regulation on Artificial Intelligence, and at the same time providing a series of measures that allow covering the requirements established in the previously analysed article 15, in relation to robustness.

The guide addresses first in the Section 4, such as establishing robustness metrics for the system, validation, and verification. This is an important section in relation to the responsibilities of the AI system provider, because it establishes the basis for defining the robustness of their AI system. An enlarged detail is attached in the Section 7.1 Annex I: Metrics and Tests for AI systems which can be consulted to broaden the theoretical basis of what is intended to be explained in the aforementioned section. In this section robustness, efficiency, performance and monitoring are related to offer a complete vision.

Following the recommendations of the European Regulation on Artificial Intelligence in Article 15, measures are presented in section 4.1 to maintain the robustness when the system is exposed to errors.

In section 4, it is established that for robustness, once defined and protected against failures, it can be guaranteed thanks to a series of measures aimed at establishing its continuity over time.

For those systems that continue to learn over time, either because they do so automatically, or in a programmed manner through data collection (or user feedback) and progressive updates, specific technical measures are provided in section 5.

The provider should consider that this guide provides recommendations on the coverage of the requirements set out in Article 15 of the European Regulation on Artificial Intelligence, but that it is not an exhaustive compendium, and that it may, in some cases, necessarily have to consider adapting the specific details (especially those referred to in the Annexes) to the nature and intended purpose of the system and the risks to the health and rights and freedoms of natural persons.

Instructions on how to obtain the robustness associated with the system, how to use and interpret it, its thresholds, and the metrics associated with it should be documented according to the Technical Documentation Guid. Below, throughout the guide we will detail dimensions to take into account for this.

## 1.3 Who is it for?

To accommodate all the issues detailed in this guide it is the responsibility of the provider of the high-risk artificial intelligence system by taking the appropriate measures proposed here, both organisational and technical. All this with the aim of ensuring that the accuracy requirements of the system are met.

Within its scope of application, the person responsible for deploying the system also has responsibilities that will materialize in concrete measures, again organizational and technical. The guide will indicate, in each case, which are applicable and the scope of these.

## 1.4 Use cases used in the guide

Throughout the guide, two **use cases** will be used as **examples** of how **to develop** the technical documentation. Examples will focus only on the provider who is responsible for generating and maintaining documentation. A detailed description of the use cases used can be found in the **Concepts and Cross-Cutting Information Guide**.

Note: Whenever an **example** is given, it will be done **in an illustrative way**. **Provider and deployer** must consider the application of **all the measures** indicated in this guide.

The use cases have been selected according to their ability to explain the information and procedures detailed in this guide, for continuity with the accuracy guide, the same cases have been selected.

The cases selected in this case for the preparation of the guide are:

- **Detection of false reports**
- **Employee promotion system.**

# 2. Introduction

## 2.1 What is Robustness for AI?

Robustness can be expressed from a series of properties associated with the AI system that will allow different facets of the AI system to be clearly defined and established.

The **highlighted** words and descriptions correspond to terms that are developed in the glossary (see 7.4 Glossary).

We can understand that the relationship between these properties, robustness and the AI system cannot be understood in isolation from one another. These properties that we have considered desirable are:

- **Reliability**: Refers to the consistency between the initially estimated values and subsequent estimated values, or the overall consistency of a measure in statistics. Robustness is often considered a sub characteristic of reliability [ISO/IEC 25059:2023]. Within the framework of the Square method [75], reliability and robustness are integrated into the quality assessment of software systems, allowing for systematic measurement aligned with quality standards [ISO/IEC 25010].
- **Stability:** In AI systems, stability expresses the degree to which the output of the system is allowed to change in a controlled manner when its inputs vary within a specific domain. This property is crucial in applications that require consistency in responses to small fluctuations in input data, such as in classification, identification, or automated decision-making tasks. Stability is based on the availability of prior information about the expected output of the system, whether known to the user, derived from simulations or defined by optimization systems. This feature is relevant in neural networks, where it helps ensure that variations in inputs don't lead to drastically different outputs. However, it is also applicable to other AI models, such as rule systems, decision trees, or statistical learning methods, as long as the system is expected to maintain a predictable and consistent response within a range of input variability.
- **Sensitivity**: In artificial intelligence systems, sensitivity expresses the degree to which the output of the system is allowed to vary when its inputs undergo changes. Similar to stability, sensitivity analysis requires verifying that the system is constrained within certain response limits and is particularly relevant in applications that demand consistency under specific input conditions, such as in interpolation or regression tasks. This criterion is typically set in quantifiable terms (such as a threshold of variation allowed within a specific range of inputs), making it easier to compare different AI architectures or approaches. Sensitivity is crucial in systems that require direct accuracy against a known baseline (ground truth), such as AI systems that control medical devices. For example, in the case of an AI-controlled insulin pump, the sensitivity of the system is a determining factor for its safety and reliability, as insulin delivery must be kept within the limits of the manufacturer's recommended doses, without exceeding those parameters.

- **Relevance**: Expresses the magnitude of the impact that different inputs have on outputs in an artificial intelligence system. Since different AI models, including neural networks, can yield different criteria of relevance and still be both valid, any protocol for comparing models must include a mechanism to resolve potential discrepancies (e.g., through a voting system). This criterion is especially useful in applications where the AI performs a task that a human could perform and therefore must be able to understand and verify the outputs. Assessing relevance is essential to confirm that the accuracy of the system is supported by the correct factors, and this verification process can be performed either by a human operator or automatically using previously validated references.
- **Reachability**: This property refers to the multi-step accuracy of an AI system in interaction with its operating environment. It applies to systems that work under the agent paradigm, where there is a perception-action-environment cycle in which the agent perceives the state of the environment, executes actions, and evaluates the impact of these until certain objectives are achieved. The reachability property assesses whether the system, by applying an AI model (such as a neural network or other type of algorithm), can achieve a set of defined states, whether they are goals to be achieved or failure states to avoid, based on its decisions. This property is useful in various AI approaches, providing a metric of accuracy and performance in closed-loop systems, i.e. those where the system continuously responds to real-time environmental conditions.

### Example – False Reporting Detection

The risks analysis for the false reporting system states that the system must be able to maintain a uniform response for small variations in the reporting texts, to prevent certain data or keywords from triggering the category of a report, causing it to be incorrectly classified. This incorrect classification is especially serious, in the case of a **classification** of a true complaint **as false**, causing it to not be treated properly.

For this reason, the sensitivity of the system is considered especially relevant. To verify the **sensitivity of the AI system**, with respect to variations in inputs, its local valuation is established, considering its local valuation, to measure the variation **around** complaints of the datasets, modified in their environment, but not very far from each other.

The local evaluation of the system will be carried out on the verification dataset and then extended to the validation process (see section 4.1.3).

The procedure followed for this verification consists of the following steps:

- The same number of reports labelled false and true are selected in the verification set.
- A generic, sufficiently broad bag of words is used in Spanish.
- For each complaint with the bag of words, new complaints are generated, with random word changes of 5%, that is, complaints are created, *close* (local) to those selected.

- It is established that, during the verification phase, the **sensitivity** of the system to these complaints generated and those that have been originally selected as the basis for these will be studied

Robustness properties can be **local or global**. It is more common to verify local than global robustness, as the former is easier to specify. By local we mean verifying the robustness properties within the system's input and output domains, and always in relation to their intended purpose. For example, an image classifier correctly predicts "a car." The local robustness property can specify that all images generated by rotating the original image 5 degrees must be classified as a car.

A drawback of verifying local robustness properties is that the guarantees are local to the tested sample and do not extend to other samples in the database, this drawback can be circumvented, increasing and consolidating the local robustness tests to be as extensive as possible, and cover variations, within the expected domain. This will make it possible to find points in the entry domain that are much less robust than their neighbours, or to find areas of domain border where the system is less robust, and to address its gradual improvement throughout the design and implementation process.

On the other hand, global robustness properties define guarantees that are deterministically fulfilled over all possible inputs (these must be specified by defining a valid range for input features, which is more difficult in configurations where individual features have no semantic significance).

Both robust environments (global and local) are equally important for a high-risk AI system and according to the state of the art of the type of model, the intended purpose and the risks to health, fundamental rights and freedoms, both must be verified. Local robustness, for allowing less robust points to be found and acted upon and improving the AI system in the process of its design, testing and validation. The global one for offering guarantees of the behaviour of the model in the input ranges of the expected domain. It is of special interest (as detailed in the Technical Documentation guide) to consider within the properties of robustness, the reasonably foreseeable misuses of the AI system, so that they can be considered in the global and local analysis of the same.

Desirable properties when designing AI systems include robustness, resilience, reliability, accuracy, security, safety, privacy, etc. Since robustness is a crucial property that poses new challenges in the context of AI systems: data quality, model degradation, stability of input features, accuracy vs. recall. And disturbances in the input data.

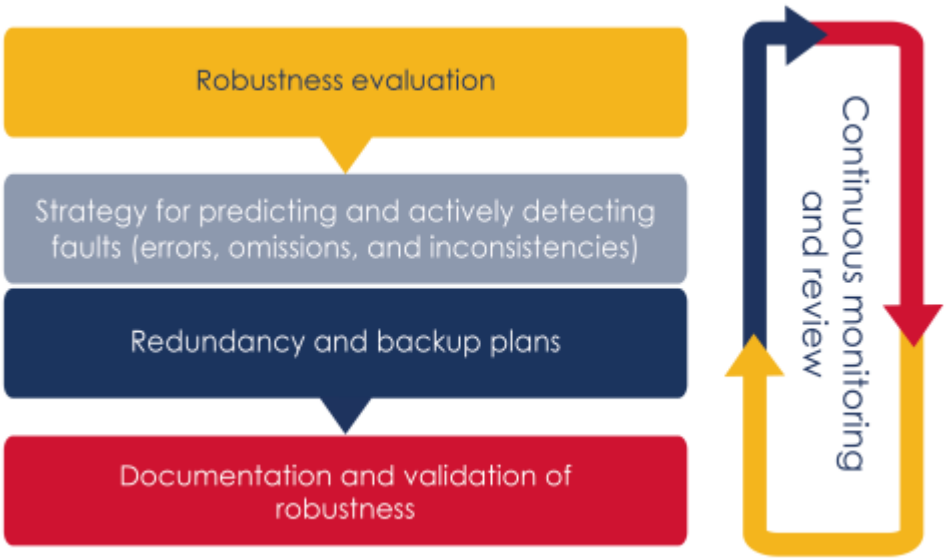
Understanding the risks especially related to the robustness of the AI system is essential for its adoption, so the risks especially linked to the robustness of the AI systems in relation to those located in the risks plan should be assessed, especially those related to damage to the health and rights and freedoms of natural persons.

Although software verification is an essential part of any industrial process, the goal of ensuring the safety and accuracy of the software used in all parts of the system makes the software verification process an important part of system certification. Therefore, the techniques used in AI systems are also subject to validation. As not all techniques that

validate software are directly applicable to AI systems (such as neural network models), new approaches are necessary for the verification of AI systems, we will address those approaches in this guide on robustness.

The changes that digital transformation entails at the organizational level must contemplate culture **change management** processes to achieve the principles of trustworthy AI. An example of game theory methodologies that can inspire such changes includes *mechanism design* theory [1] [113].

Moreover, a summarized and visual way to see the changes associated with meeting the robustness requirements is:




[1] deserving of the Nobel Prize in Economics in 2007, in which the objective is to obtain desired results according to a specific solution concept, when we are given an objective function and we do not know the mechanism.

# 3. European Regulation on Artificial Intelligence

## 3.1 Previous analysis and relationship of the articles

This article establishes the requirements that must be met in terms of three fundamental aspects *"Accuracy, robustness and cybersecurity"*. Cybersecurity and accuracy are specifically addressed in their guides.

In this section we are going to emphasize the paragraphs of this article that are specifically oriented to the robustness of high-risk AI systems, which are concentrated in sections 1 and 4.

**Article 15 Accuracy, robustness and cybersecurity paragraph 1 →** establishes the need to maintain adequate levels of accuracy throughout the life cycle of the AI system, and we will therefore indicate a series of measures aimed at ensuring that AI systems **do not degrade** their performance and robustness specifications once they are put into operation, throughout their life cycle.

AI systems must not present problems of operation (compatibility with old libraries they use or data they process) or quality problems in terms of their robustness as they are used over time. To this end, they must **achieve** and **respect** minimum reasonable levels of **robustness** (and associated metrics) pre-established in their requirements and specific to the task they solve, which guarantee their safety and proper functioning.

**Article 15 Accuracy, robustness and cybersecurity section 4 →** establishes the need for measures to protect against errors, failures or inconsistencies.

These aspects described in these sections broaden the focus of the European Regulation on Artificial Intelligence's requirements related to robustness by addressing failures and redundancy systems as an important requirement. In the same way, the article reflects that the robustness for those systems that continue to learn after their implementation must be considered in a specific way and with measures.

The concept and interpretation of robustness, in terms of technical measures and evaluation metrics, is inseparable and dependent on the domain of application and, therefore, its intended purpose.

## 3.2 Content of the articles in the AI Act

### AI Act

#### Art.15 – Accuracy, robustness and cybersecurity

1. High-risk AI systems shall be designed and developed in such a way that they achieve an appropriate level of accuracy, robustness, and cybersecurity, and that they perform consistently in those respects throughout their lifecycle.

2. To address the technical aspects of how to measure the appropriate levels of accuracy and robustness set out in paragraph 1 and any other relevant performance metrics, the Commission shall, in cooperation with relevant stakeholders and organisations such as metrology and benchmarking authorities, encourage, as appropriate, the development of benchmarks and measurement methodologies.

3. The levels of accuracy and the relevant accuracy metrics of high-risk AI systems shall be declared in the accompanying instructions for use.

4. High-risk AI systems shall be as resilient as possible regarding errors, faults or inconsistencies that may occur within the system or the environment in which the system operates, in particular due to their interaction with natural persons or other systems. Technical and organisational measures shall be taken in this regard.

The robustness of high-risk AI systems may be achieved through technical redundancy solutions, which may include backup or fail-safe plans.

High-risk AI systems that continue to learn after being placed on the market or put into service shall be developed in such a way as to eliminate or reduce as far as possible the risk of possibly biased outputs influencing input for future operations (feedback loops), and as to ensure that any such feedback loops are duly addressed with appropriate mitigation measures.

5. High-risk AI systems shall be resilient against attempts by unauthorised third parties to alter their use, outputs or performance by exploiting system vulnerabilities.

The technical solutions aiming to ensure the cybersecurity of high-risk AI systems shall be appropriate to the relevant circumstances and the risks.

The technical solutions to address AI specific vulnerabilities shall include, where appropriate, measures to prevent, detect, respond to, resolve and control for attacks trying to manipulate the training data set (data poisoning), or pre-trained components used in training (model poisoning), inputs designed to cause the AI model to make a mistake (adversarial examples or model evasion), confidentiality attacks or model flaws.

## 3.3 Correspondence of the articles with the sections of the guide

The following table details the sections of this guide that address the different elements of this article:

| Article | Regulation requirement | Section |
|---|---|---|
| 15.1 paragraph 1 | Adequate and consistent level of robustness throughout the life cycle | Section 4.1 |
| 15.4 paragraph 1 | Resistance to errors, failures and inconsistencies | Section 4.2 |
| 15.4 paragraph 2 | Redundancy, backup, and fail-safe plans | Section 4.3 |
| 15.4 paragraph 3 | Systems that continue to learn after they are put into operation. | Section 4.4 |

# 4. How to approach the requirements?

## 4.1 Assessment of robustness

The applicable robustness metrics guaranteed by the provider in the documentation should reflect and be a sign of the quality of the system. Otherwise, when the requirements established in the documentation in terms of minimum established notions or metrics of responsible for the system mechanisms to notify a human operator that their supervision is required (according to human oversight guide), both for a temporary variation of these robustness metrics, as in the case of persistent variation in the form of a degradation of response.

The **providers** of the AI system must implement capabilities that provide it with the necessary robustness according to the intended purpose and with the aim of mitigating the risks found in the risks plan (see risks guide).

At any point in the valid life cycle of the system, the provider shall:

- Provide the deployer with mechanisms to **observe, monitor, and report** different types of model degradation (out-of-range model outputs, potential attacks by adversaries, or disturbances that data or model may suffer) that exceed the limits documented as reasonable for each robustness metric, in order to make them reproducible for remediation.
- When the guaranteed robustness requirements **change, correct them** to ensure the standards or metrics provided according to the documentation. When this is irremediably not possible, and documented guarantees are breached, both the documentation and the database card and the responsible AI model card must be updated and the deployers involved must be notified (see accuracy guide). If this adjustment is made with an update, the rationale for the change must be explained in the technical documentation, as indicated in the technical documentation for updates.

If conflicts arise between the application of the requirements of Chapter 3 Section III of the European Regulation on Artificial Intelligence (in short, the guides accompanying this sandbox), it must be demonstrated that these are formally the result of enforcing ethical principles, and report the conflict that results in serious damage to the robustness of the model, This type of consideration must be adequately documented, as established in the Technical Documentation Guide, indicating the reasons and detailed explanations of the preferences made. All parties involved must ensure that they are resolved. For example, using established compliance procedures as a standard, best practices and those processes guided by existing and upcoming certifications.

The measures to be taken by the provider in place will include those that measure the quality of a system based on its ability to maintain a certain level of robustness associated

with its accuracy under all circumstances [SC42 N483 ISO IEC TR 24029.1]. Statistical analyses can be done to demonstrate this capability, but to test it requires some form of formal analysis (See approaches in [ISO SC42 N938 ISO IEC CD 24029.2], ISO/IEC 25059:2023], Square method [75] and methods in [ISO/IEC TR 24029-1:2021 to assess robustness without the use of formal methods).

In this section we are going to develop, therefore, the measures that are the responsibility of the provider, and we are going to group them by specific aspects of the AI system, in order to put the robustness in perspective for each of them.

The following points represent an indication of the topics to be addressed, related to the assessment of robustness:

- The relationship of the life cycle and robustness, considered as a vertical for the AI system.
- An approach to selecting the AI system's robustness metrics is presented.
- The verification and validation process for the selected metrics is established.
- Both efficiency and performance are considered to be elements associated with robustness, so we provide indications on how to achieve them.
- Finally, the aspect of monitoring the robustness is addressed.

**Organizationally**, the **deployer must** have sufficient training to detect when a degradation of the system occurs and thus be able to provide feedback to the provider when required by the product documentation. Thus, the deployer must:

1. Know the basics associated with robustness and become familiar with the documentation that gives you access to:
    a. Visualize monitoring dashboards of all robustness metrics. These dashboards must be integrated into the AI system by the provider, they can be developed using commercial, open-source solutions or in-house developments.
    b. Interpret and use the output of the system in terms of its robustness, at the level of functionality, model and system.
2. Access available tutorials (online, tutorials, additional teaching material) beyond the product documentation so that you can fully understand the functionality of the AI product when necessary to achieve the above point. Given that these are high-risk systems, specific training plans for personnel should be considered when the qualification or training of operators may not be sufficient.

It should be understood that the technical means integrated into the high-risk AI system have been made available to the deployer by the system provider, providing knowledge of them during the commercialisation phase and throughout the life cycle of the system itself.

**Technically**, in order for the deployer to be able to assess the output of the AI system as a whole, and for a particular data entry, they must have a sufficient level of detail in the documentation to be able to act on the monitoring guidance, if robustness degrades below the minimum guaranteed levels. Thus, it is the responsibility of the deployer:

Know how to use and access the interface to obtain associated metrics of robustness and computational efficiency. In order to use the system, the deployer must be familiar with:

- The interface of the system and its outputs, for each functionality of the system, and for the system as a whole. Within its intended purpose and in accordance with its use.
- Statistics that serve to compare the operation and metrics of the system provided with other existing models of the state of the art or used as an algorithm/base model.
- Mechanisms for managing notifications of potential documented operating errors, inspection procedures, supervision, correction, notification of potential erroneous exits, or the absence thereof.
- Possible actions to take when the output is not endowed with explainability consistent with its robustness, or when you witness unacceptable levels of robustness or uncertainty in the output.

In addition, you can improve your user experience by studying the relevance of applicable and chosen robustness metrics, depending on the use case, to observe and understand differences in statistical significance of models.

> Know how to explore the robustness metrics and other quality metrics associated with the good functionality and supervision of the project.2. Know how to notify or correct potential operating errors (erroneous outputs, lack of outputs, etc.) to the provider and inspect input and output data.

### 4.1.1 Life cycle

For each element of the lifecycle of a high-risk AI system, a level of robustness is established in line with the intended purpose of the system, establishing appropriate measures. Depending on the phase of the life cycle, the measurements must be carried out by the provider (design, development) or deployer (during its production or operation), avoiding degradation, and any operating problem that reduces the robustness of the system during its life cycle.

The robustness of an AI system must be verified throughout the life cycle defined in this order:

- **Design and development**. Recognized characteristics should be identified, and separability checked. Experiments will be designed to evaluate the capabilities of the AI system in terms of a deliberate set of variables to which the model must show robustness. For example, ensemble learning approaches (a technique that combines multiple individual models to create a final ensemble learning model) will be considered to increase the robustness of the models, including resistance to adversarial or evasive attacks. In this aspect, and related to the data, we can consider the following example: in an automatic speech recognition system, the use of sufficiently large and diverse data will be demonstrated, which leads the model to support and adequately deal with accents, background noise and technical vocabulary.
- **Verification and validation (V&V).** The covered portions of the input domain should be verified (e.g., with sensitivity criteria, or using formal methods) and the impact of the disturbance should be measured by forcing the intentional, formal, and intended purpose-related disturbance into the verification and validation process. Section 4.1.3 presents different measures to address this aspect.

- **Implementation / production.** In this case, the impact of problems caused by numerical precision (compilers that rearrange or replace operations, reduce numerical precision, or change the rounding process) can be verified. In these scenarios, and as defined in the cybersecurity guide, to avoid failures caused by configuration problems, the validation and production environments of the AI system have to be identical, or equipped with the mechanisms that allow them to be identical (as indicated in the aforementioned guide, using container technologies or equivalent). In section 4.1.6, we address in detail aspects focused on robustness and systems in production.

### 4.1.2 Selecting Metrics

So far, we have seen the concept of robustness and its desirable properties, in the organizational framework of the AI system. This organizational framework is the necessary scaffolding for us to establish the detail of the technical operation on how to quantify robustness. In successive sections, we will address how robustness is approached, technically, with applicable methodologies and associated metrics.

The methodology for assessing the robustness of a model is based on the following steps:

1. **Establish robustness requirements or objectives and associated metrics**: To what extent does the system need to be robust? What are the robustness characteristics of interest? Robustness **properties** demonstrate the degree of **generalizability** of the model to new data, compared to its accuracy in the data with which it was trained, or with expected data in typical operations. The set of criteria for deciding on robustness properties must be constituted. Given these, the metrics that quantify the elements that demonstrate achieving robustness will be identified.
2. **Planning experiments that demonstrate robustness**. These will be based on statistical methods, formal or empirical, or in practice, on a combination of several.
3. **Conduct experiments:** The experiments defined in step 1 must be executed according to the established plan and the results, the datasets used, and the output values must be recorded, with which the metrics are calculated in an aggregate manner.
4. **Analyse results** against the established metrics selected in point 1.
5. **Interpret** the **results** to inform the decision, these must be interpreted not only in the value of the metric itself, but also in relation to its evaluation with all experiments, both progressively (evolution over time) and aggregate (statistically).
6. **Decide on the robustness** of the system given the criteria and interpretation identified above. If robustness is insufficient, return to the stage that alleviates its deficiencies: adding robustness objectives, metrics, other measures, rethink experiments or modify the data collection protocol for the experiment or correct the analysis. This iteration should be repeated until the established robustness goal is reached.

**Example – Employee Promotion System**

During the design phase of the AI system in order to establish the robustness criteria and how these are going to be applied to the system, the system provider follows the following considerations, aligned with the points highlighted in this section:

1. The following characteristics are selected for the AI system: reliability, stability, sensitivity, and relevance. According to the design team, the thresholds for each of them are established, and how they should be maintained throughout the life cycle of the system. Validation data sets are generated for each of the different features, data that are not used during training. In addition, a control group is available for all the others that is also evaluated against these characteristics.
2. Since it is not possible to develop a formal method suitable for the AI system (see section 7.2.1) that can build a system to which to apply a mathematical optimizer, a combination of statistical and empirical methods is chosen to perform the experiments.
3. The experimentation plan and the recording mechanisms are established. The provider chooses to keep within its version control system, associated with the versions of the model, the accumulation of experiments for each step of the validation and verification iterations.
4. The analysis of the results will be carried out using the **AUC** and **Lift techniques.**
5. The analysis interprets the results in relation to the specification values defined in point 1.
6. It will be iterated until the desired robustness is achieved in the validation and verification process.

Although specifications can be implemented in different ways, during development you can decide, for example, on specific threshold values to complete the optimization. This value is not necessarily specified in advance and therefore the system will be more or less accurate, depending on the data. This is why it is necessary to validate the model by testing on specific data to observe how it behaves (**performance validation**), and this data will depend on the intended purpose.

For metric selection, the 7.1 Annex I: Metrics and Tests for AI systems, a reference list that can be consulted is presented, with the consideration that the provider must select the most appropriate metric for the measurement of robustness, always considering the intended purpose and the risks.

### 4.1.3 Validation and Verification

Robustness assessment requires the intervention of two very important steps that can be clearly identified, **validation and verification**:

- **Verification**: Confirmation, by providing objective evidence, that the specified requirements (see ISO/IEC 25000:2014 - *Systems and software engineering* 4.43 and ISO/IEC 25030:2019 - *Systems and software engineering* 3.22) have been met in

accordance with the manner in which they were specified[2]. This phase does not require running the program with real data and is executed before validation.

- **Validation**: Confirmation, through the provision of objective evidence, that the requirements of the intended purpose of the AI system have been met (see ISO/IEC 25000:2014 - *Systems and software engineering* 4.41 and ISO/IEC 25030:2019 - *Systems and software engineering* 3.21). Validation includes testing the AI system, previously verified, using real datasets, running the code to validate that it does what it should do and works as expected.

Most types of data processed by different architectures, e.g. neural networks, can be analysed by at least one formal method (see ISO/IEC 24029-2:2023. *Artificial intelligence (AI) – Assessment of the robustness of neural networks*). Each method, with its advantages and disadvantages (mainly scalability, etc.) You can tackle one or more of the desired criteria in the robustness assessment model: stability, sensitivity, relevance or achievability. In the 7.1 Annex I: Metrics and Tests for AI systems, a series of mechanisms that can be used for the analysis of these types of data are offered.

**Example - False Reporting Detection**

To expand the mechanisms for verification and validation of the false complaint detection model, an incomplete formal, black-box and deterministic method is established (see Annex I).

To do this, between the complaints established as true and the complaints established as false, a control group is extracted. From the original total set, a list of words related to both types of complaints are extracted. With this list of words, for each complaint of this control group, the statistical count of frequency of occurrence is established.

With this data, a function is constructed that provides, in relation to the distribution of words, the exact result for each of the complaints of the control group.

In the validation and verification phase, the predictions of the AI system are compared with the predictions of the established function, both for specimens in the control group and outside it. In this way, this formal method **complements** the verification and validation process.

### 4.1.4 Efficiency in AI systems

High-risk AI systems can run in environments with different computing capabilities, both for those who are in the inference stage and for those who continue to learn over time, the available computing capacity will profoundly affect the robustness of the system.

It is therefore very important that the metrics chosen to validate the robustness characteristics are validated and verified (see section 4.1.3) in *hardware* environments that

[2] Meeting specified specifications, early checks for programming errors, checking programs and documents through a system walkthrough with a step-by-step tutorial, etc.

exactly replicate the final computing capacities (memory, CPU, numerical accuracy, processing speed, etc.) to which the AI system will have access. If the system is to be installed by the **deployer,** the installation instructions must be clearly indicated the needs of that *hardware* and its direct relationship with the accuracy of the system.

In those cases, in which there may also be calculation or memory restrictions, we provide examples of measures:

1. Machine learning at the edge, when accuracy and robustness are required with limited computational resources, such as running a model on a mobile device such as a phone in real time or even on IoT devices (smart sensors) or other devices (cameras).
2. **Distillation** [47], when there are limitations on the memory of the system that the AI system will run.
3. **Distilled privileged learning** [92], when not all the data modalities that were used during inference-time training are available.
4. **Continuous learning** [25], when the model cannot grow in architecture and number of parameters as learning tasks increase and must maintain minimum levels of accuracy for all the tasks for which it was trained.
5. Network compression and pruning techniques (when the memory occupied by the models exceeds a memory threshold) or others.

### 4.1.5 Performance in AI systems

The robustness metrics associated with *system hardware* and performance should be described in the system documentation before designing the accuracy and performance tests of the system execution stage. In this way, the target performance and efficiency will be established to, for example, deal with a specific number of queries or requests per hour.

- It should consider the robustness of the accuracy, performance, and strategies of their implementation within the AI system as a whole, and how they are constrained by limitations in the availability of resources, such as memory or power. For example, one metric can be performance per watt or performance per watt (6.6.5 ISO SC42 N1011), although joules per watt or nano joules per pixel can be more intuitive to calculate aggregated statistics [38]. At the *hardware* level, the required memory capacity (e.g., in Python, Memray can be used for memory reservation tracking and reporting).
- Computational complexity metrics: can measure degradation in the robustness part. For example, in machine learning models, they would translate to inference latency (e.g., classification latency, classification efficiency, classification performance, and power consumption).
- Performance metrics: such as throughput in short periods of time, e.g. in FLOPS (*Floating Point Operations Per Second*), or efficiency, throughput maintained over a representative unit over a long period of time, e.g. in FLOPY (*Floating Point Operations Per Year*). For example, while latency is often used to measure customer-facing interactions, and performance in server applications, in AI they must be tailored to the specific problem, for example, in an image classifier, efficiency can be measured in the number of images processed in one second for a batch size much

larger than 1, or in latency when the batch size is one (latency is inversely proportional to efficiency).

Additional considerations that contribute to achieving proper model robustness are based on hardware *and* its monitoring during different points in the AI model lifecycle:

- Hardware for the use of more complex architectures. To achieve the desirable values of the accuracy and robustness metrics specified in the system requirements, providers may use compute acceleration measures for specialized hardware, graphics processing units (GPUs), application-specific integrated circuits, or instruction sets embedded in central processing units (CPUs).[3]
- The hardware and architectures associated with a level of accuracy and robustness must be transparent, fixed, and updated, where appropriate, in the established documentation (according to the Technical Documentation guide).
- Possible inconsistencies in the system due to classification latencies should be considered[4].
- We once again reinforce an exact continuity between *development*, verification and validation hardware environments, which allows the metrics obtained to be understandable in contexts of identical computing capacity and architecture. In those situations, in which it is not possible to have the same environment between phases of the life cycle, **it will be necessary** to establish **specific robustness** conditions for each environment, in order to allow the passage from design to verification and from this to validation. In that case, the target values and considerations (defined as specifications) must be set out in the hardware and its final system architecture. Again, we remind you that, if the system is to be installed at the deployer's premises, in any possible marketing format, minimum (and ideally recommended) requirements must be clearly defined in which robustness requirements can be guaranteed.

**Example – Employee Promotion System**

The employee promotion system in one of its marketing formats is delivered to the deployer so that he or she can install it. The provider decides to establish a series of additional metrics to the robustness: latency of an individual evaluation and processing capacity per second of evaluations, considering in the design of the system that these two elements are key to prevent too slow processing from harming the position in the classification of a person due to delay in the calculation of their score.

[3] Other accelerators can be applied to simple functions (such as matrix multiplication) or complex functions (a ResNet function). In addition, due to restrictions on energy saving measures, the extension of the training of certain models could favour accuracy over others trained for a shorter time.

[4] Classification latency measures the duration between the user's input of data to the model and the output of the model's inference, and determines the quality (in this case, the accuracy and robustness) of the service provided, which must fall within an umbrella range that defines the operational constraints of the model's implementation. and reporting when the execution of the model inference exceeds unreasonable limits (ISO SC42 N1011).

Since these two points are strongly dependent on the relationship with the *hardware*, and during the design and validation/verification phases you will not have access to the same *hardware* resources as during production:

- Minimum values are set for *development* and pre-production hardware, which must always be achieved in any iteration of the algorithm.
- Minimum hardware specifications are established *on* which the expected values are achieved in a production environment. These tests are executed whenever a version is considered ready for production and are a necessary condition for the version to be considered successful (always linked to other significant elements of robustness, not as a single condition).
- The system includes a special recording of the two characteristics (latency and evaluations per second) during its execution to inform of their possible degradation, both due to changes in the data and modifications in the initial volumes.
- The indications of the minimum requirements and the volumetry considered are generated in the system instructions, so that the user can establish hardware requirements in their facilities that guarantee the operation of the AI system.

### 4.1.6 Robustness monitoring

AI systems are software artifacts with a very special nature, which means that their behaviour over time can vary, not only in those cases in which the system continues to learn over time, but in general in all scenarios.

The appearance of unexpected domain spaces in the input data, or even the existence of undetected errors or failures that can affect the accuracy, robustness or even the intended purpose causing harm to people or affecting rights and freedoms, are aspects that cannot be ruled out.

This type of monitoring fits into the scenario of post-market monitoring (see that guide).

Focused on robustness, proper monitoring involves looking at statistics, data distribution, and changes in business usage. Other notions to keep in mind are:

- Feedback mechanisms should be provided to distinguish between errors of the deployer misusing the system from those of the system itself being inflexible to meet the needs of the deployer (system errors), or the system making erroneous assumptions about the deployer (Context errors).
- The types of errors in a troubleshooting process that includes a record of problems, their reasoning or cause and type of fix, permanent or short-term and their effectiveness (and according to the records guide) and documented following the indications for changes and updates as indicated in the technical documentation.
- Providers should implement the ability to audit models based on ethical principles (e.g., [57]), so that the key tensions responsible under ethical consequences are identified, and recognize these as commitments. To this end, robustness assessment

methodologies (based on the assessment of Stability, Sensitivity, Relevance or Achievability [ISO/IEC 24029-2:2023 SC42 N938 ISO IEC CD 24029-2]) or conformity assessment should be used to guide the processes that preserve CapAI's principles of responsible AI [57].

There are different platforms that allow this control or monitoring, both commercial and open source, the provider can select any of them to formalize this monitoring or opt for its own development as long as the chosen solution meets the requirements of the European Regulation on Artificial Intelligence.

Robustness cannot be understood without being related to the supervision and transparency that establish how information on the status of the system is communicated and the impact it has on decision-making, it is recommended to consult or revisit the respective guides on these aspects. It is important to consider that **an** expert in the domain **should be informed** of the intended purpose of the system, who may not necessarily have knowledge of AI.

## 4.2 Robustness and Resistance to Errors

In the introductory section of the guide we have seen, supported by the considerations provided by the European Regulation on Artificial Intelligence, that the system must have **technical robustness** (as indicated in recital (75) of the Regulation) to prevent failures, errors or inconsistencies from having consequences for safety, or negatively affecting fundamental rights.

Likewise, throughout section 4.1, the organisational and technical measures necessary to establish the robustness of the system and how to measure it over time have been developed.

On this aspect, it is Article 15 of the European Regulation on Artificial Intelligence in paragraph 4, which indicates that:

**AI Act**

### Art.15.4 – Accuracy, robustness and cybersecurity

High-risk AI systems shall be as resilient as possible regarding errors, faults or inconsistencies that may occur within the system or the environment in which the system operates, in particular due to their interaction with natural persons or other systems. Technical and organisational measures shall be taken in this regard.

Therefore, it is clear that it is necessary, associated with the concept of robustness, to establish strategies for predicting failures, unintended consequences of negative effect and

effects that affect the safety of people or that negatively affect fundamental rights will have to be studied and assessed to avoid their scope in possible misexecutions of the AI system.

The provider is responsible for implementing measures to mitigate the lack of robustness in AI systems against errors, failures and inconsistencies.

**Organizationally**, the provider:

- It will facilitate the implementation of alerting techniques (pointing out which special cases need changes in the rules to correct those decisions in the future) to the responsible parties or, in the last case, if no response is received from them within a time established as safe, functionality will be implemented that allows the automatic shutdown of the system. In the specific case of robustness that concerns us, the system must be prepared to launch such alerts when robustness is compromised. AI-based automatic deployment must use safe fault protocols, such as allowing human intervention at any time. This aspect is detailed in the human supervision guide.
- Introduce safe fault protocols. The development of AI technologies with safe failure protocols must allow humans to predict catastrophic mishaps even if their occurrence is rare, it must reserve a design planning that will entail a dedicated effort to try to predict future risks and traps into which the system may fall.

**Technical** measures to preserve robustness such as resistance to failures, errors or technical inconsistencies include the following aspects.

Prior to the development of the AI system, dialogues must be established with different stakeholders, maximizing the diversity and inclusion of different domain profiles during the design, development, maintenance, application, monitoring and use of the system. The aim is to prevent catastrophic consequences from the subsequent use of the system when it has been implemented.

One approximation to this approach is to develop different model design evaluation committees that will be involved, with different points of view from experts with different backgrounds, in order to predict (and tackle, in the design phase), inconsistencies in the design or implementation of the system that may lead to unintended consequences. In SMEs and start-ups, these committees can focus on working groups throughout the company where a cross-cutting operation of the process can be evaluated.

The thinking to be established consists of the idea of "*safe AI based on suffering*, in favor of safe failure": It is easier to strive to avoid very tragic triggers that could occur when deploying a given AI system, than to try to make sure that all aspects of the system will always work correctly and as expected [40].

Another approach is the classic one in secure AI of the trade-off between the probability of success and controllability. In this case, if it is difficult to control the values of the resulting artificial agent, then the risk that is supposed to be a dead end, whereas in the AI of certain failure it does not require worrying about the controllability of the system at the same level, as long as there is enough controllability to predict that they can be avoided.

In safe AI focused on *suffering*, intervention strategies are proposed such as:

- Influencing system outputs with controlled AI, through the propagation of the system's core values, or targeted safe AI strategies such as **differential progress** to differentially move the system away from the worst-case scenario. Also identify safe AI where it is needed most, identifying the worst event that could occur from a controlled failure, and work towards identifying an overall control or system shutdown mechanisms.
- Designing ways to make the AI system fail with graceful or "benign" failures, so that there would be even worse failures than the design of this graceful failure[5].

### Example – Employee Promotion System

During the risk analysis of the AI system, it has been considered a risk that the system does not have the same amount of information for all employees, which can cause the evaluation not to be carried out correctly, causing those employees who have less information within the system to be discriminated against. The errors that can cause information not to be incorporated into the system for a given employee are very varied, and difficult to inventory because the system integrates with several sources.

To remedy this, **the possibility of missing information** (i.e. failure in data entry) is considered possible for all employees. To do this, two ways are created to deal with this *potential error*:

- The system creates a list of all employees who have less information than the average and creates alerts about them so that it can be reviewed by the staff and the lack of information corrected, for any of the origins.
- The system is capable of providing an **alternative** employee evaluation list, in which an evaluation is added on the missing information for those employees that is not complete, based on an analysis of the data of those who, if they have complete information, to have an approximation of the promotion status for those employees with failure to incorporate information.

**Organizationally**, deployers should familiarize themselves with past cases and lessons learned in similar use cases reported, documented, and kept up to date by the provider in the updated system documentation.

The deployer is responsible for using the channels provided by the provider to properly report the presence of errors, since it is their responsibility to monitor them. This implies, therefore, that the provider must provide means to manage the receipt of notification of the presence of errors, by the person responsible for the deployer.

Technically, the deployer will need to be prepared to:

[5]The implementation of secure system values can be done by asking whether correctability reduces the risks of the system causing suffering [41].

1. Know or have – and train, if necessary – personnel who could comply with the provider's established concepts and possible implementations of "safe failure".
2. To act in the face of the lack of robustness as a concept of degradation of the model.

## 4.3 Redundancy and backup plans

The path of knowing, guaranteeing and preserving the robustness of the system as a characteristic of the system, has the additional derivative of having redundancy techniques for the operation of the AI system, in such a way that it can be guaranteed.

**AI Act**

### Art.15.4 – Accuracy, robustness and cybersecurity

The robustness of high-risk AI systems may be achieved through technical **redundancy** solutions, which may include **backups** or **fail-safe plans.**

Therefore, part of the mechanisms to achieve robustness are based on redundancy techniques, which include backup or fail-safe plans.

The **provider** is primarily responsible for establishing redundancy techniques in the AI system and ensuring that they are in line with the intended purpose of the system and also paying attention to foreseeable undesired outcomes.

The provider must document and facilitate the monitoring of the level of coverage of the quality tests carried out on the system (see cybersecurity guides, accuracy and cybersecurity, as well as the quality management system guide), correlation analysis and redundancy that may reveal potential attacks on the system or degradation of its quality. Examples of techniques to be used are: **correlation of controls and measures**, e.g. data restores, System restores, KPI measures (number of incorrect data restores), degree of coverage of preventive defences, coverage testing, quality testing and data restoration.

The provider will provide actions for the continuous design of the system based on risks safeguarding, to support the AI system with resilience to attacks, robustness to changes in the operating environment, minimizing unexpected unintended damage, and reproducibility.

The provider will need to provide:

1. **Automatic data copy mechanisms** (according to records guide) that allow the redundancy of models, algorithms, logs or records provided by the MLOps and DevOps libraries, data and model executions.
2. Mechanisms for the **safe failure** of the same components throughout the life cycle of the AI system.
3. Tools that allow **an action plan** to be executed when the failure has occurred, for the effective recovery of data, models, algorithms and the AI system.

4. **Tools for reproducibility**, tracing (according to the records guide), provenance or lineage of the data that guarantee the robustness of the model used and allow detecting modifications in the data that lead to significant changes in the robustness or performance of the model.

Both the redundancy and the action plan of the same components indicated above must be local and global; ideally, using a second device or mechanism, such as secure cloud hosting.

To **technically execute** the previous points 1, 2, and 3, it is recommended to follow the following MLOps methodologies and tools:

- Geographic redundancy **measures.**
- **Systems scalable in capacity** according to demand, availability and performance monitoring system.
- Implementations of **backup systems and** data version tracking.
- Models and algorithms that are included in **DevOps platforms** such as all those existing in the market, both commercial and open source (free). They provide key features such as automation of development and distribution operations, centralized cloud computing, continuous error monitoring, and container technology to run microservices or resource-constrained applications across multiple environments, which would allow orchestration to achieve geo-redundancy scenarios and scalability according to demand.

## Example – False Reporting Detection

The processing of complaints at police stations is a process that requires continuity and availability over time, which allows a citizen to file a complaint at any police station at any time.

To ensure that the AI system can perform the operations of evaluating the complaints, with high availability the AI system is distributed in data centers (data centers), with redundancy systems in different physical locations, both in storage and in application servers, to prevent a failure in one of the locations of the data center from suspending the execution of the service.

They introduce an auto-scalable response system, which allows the system to increase the performance of the database and applications according to demand.

For the integration of all this with the process of distributing new versions, an open-source DevOps tool is connected, which before releasing a new verified and validated version, performs a battery of unit tests, integration and checks the accuracy of the system and if everything is correct distributes the update over the infrastructure.

With these considerations, the AI system has an infrastructure that guarantees redundancy and a deployment mechanism that ensures that the necessary requirements for its implementation established from the design and through risks analysis are met.

As for the data (point 4 above), configuration parameters or the model learned, they must be downloadable to promote their reproducibility. To promote the publication of results and the lineage of models, when the weights of a model, pre-trained models or datasets are not possible to store them in the main code repository that stores it, the use of open access tools and trusted repositories will be promoted.

To keep track and record of the databases used and the modifications they undergo, free database-specific version control tools can be used.

For their part, the **deployer** in this regard must know the redundancy measures that are provided, as well as the backup and fail-safe plans.

It will be especially so if the AI system is installed by the deployer in their own facilities. In this case, it is the responsibility of the deployer to accurately replicate the installation and maintenance instructions provided by the provider, associated with the measures described in the previous section.

**Organizationally**, the user's responsibility will be to know, apply and keep an eye on the associated accuracy metrics and statistics, which may reflect changes between the distributions of the training data and inference (being able to exhibit distribution shift or data shift (*dataset shift* or *shift distribution*)).

In terms of **technical** training, the deployer is responsible for knowing, applying and keeping an eye on the fault backup system in the event of a failure, if specified in the provider's documentation.

## 4.4 Robustness for systems that continue to learn

The behaviour of AI systems that continue to learn after being commercialized, requires special focus and treatment to ensure their robustness, especially in the possibility of the appearance of biases.

### AI Act

#### Art.15.4 – Accuracy, robustness and cybersecurity

High-risk AI systems that continue **to learn after being placed on the market** or put into service shall be developed in such a way as to eliminate or reduce as far as possible the risk of possibly biased outputs influencing input for future operations (feedback loops), and as to ensure that any such feedback loops are duly addressed with **appropriate** mitigation measures.

It is important to consider that continuous learning should be extended, conceptually in terms of coverage of the proposed measures, when it is **updated** by having been retrained following an update process, and not only the systems that continue to learn automatically during their use.

Provider and deployer must ensure that the constant and continuous training of the AI system over time does not deteriorate its robustness, accuracy and performance metrics.

The provider, as responsible for the design, implementation, verification and validation of the AI system, is primarily responsible for meeting the requirements established in this section. At the **organizational level**, they must consider:

- The AI system provider should have experts who can cover monitoring the degradation of the system during each and every one of the different stages by which the model can suffer degradation at the level of data scientist, data engineer, as well as changes in the operation of MLOps and DevOps.
- At the level of design and conception of the model, continuous learning strategies, compression of deep learning models must be considered to tackle the possible catastrophic forgetting. Other measures of robustness to be studied in these stages of system inception are the design of mechanisms (*Mechanism Design* [113, 53, 51]).
- It must be ensured that the constant and continuous training of the AI system over time does not deteriorate its KPIs and metrics of accuracy, robustness, performance, and other metrics that reach the principles of reliable AI.
- The provider shall indicate formal annotation techniques in the system documentation on specific cases of execution of the AI system when the must or has the option of interacting, interoperating with other systems or humans, reusing or integrating with other systems. In particular, **feedback** or improvements **provided** by **humans** should be noted so that the future version of the updated model takes them into account, within the paradigm and human-centered AI, Human *on the Loop* (HOTL) and *Human in the Loop* (HITL).

The provider, in relation to such reported data, shall maintain a public database of lessons learned from failures, which shall serve to maintain a continuously updated knowledge base, and which integrates safety into the design and the paradigm of the safeguard design. It can evoke situations or visualizations of examples that can help discern and prevent unfavourable HRAIS scenarios to avoid.

The deployer must have access to and be familiar with the functionality of the system in order to be able to visualize and report, at any time, if necessary, to the provider, anomalies that do not ensure that the constant and continuous training of the AI system over time it does not deteriorate its KPIs and metrics of quality, robustness, efficiency, performance and uncertainty, among others.

### 4.4.1 Types of AI system degradation and mitigation plans, strategies and tools

The quality in terms of robustness of an AI system is only as good 1) as good as the data is (see Data Guide), and 2) as solid as the model and its infrastructure are once in production.

Once the model is trained, the provider may be forced to find a trade-off between accuracy and stability. However, a solid system must avoid its degradation, for this it is necessary

that for the degradation taxonomies that may exist, we can transfer correction or detection mechanisms.

There are three main types of potential degradation of an AI model, as it continues to be trained or used for inference; All of them shall be monitored and evaluated from the development phase to the implementation phase during the life cycle:

1. The **deviation of the model**: This refers to the difference between the average predictions of the model and the actual values. It represents the systematic error that indicates the model's ability to capture the underlying relationships in the data. A high deviation suggests that the model is too simple and fails to capture complex patterns, leading to underfitting. To mitigate drift, it is advisable to increase the complexity of the model or improve the quality and quantity of training data. It is essential to balance deviation and variance to avoid both underfitting and overfitting the model.
2. The deviation or **drift of the model over time**: it is a phenomenon in which the predictions of the learned model are degraded due to alterations in the environment. It occurs when P(Y|X) changes: the same inputs produce different outputs. Strategies to mitigate and monitor this type of drift: the battery of tests must be responsible for detecting model and system failures silently, the impact of the change in the data with respect to the original training data of the model and its impact on the accuracy and associated metrics.
3. Data **drift over time**: occurs when the input data (or independent variable) of a model (the joint distribution, P(X)) changes. It's one of the biggest reasons why a model degrades its accuracy over time. It occurs when P(X) changes, or P_t1 (X) ≠ P_t2 (X)). When P(X) changes, a **domain shift occurs**.

### 4.4.2 Strategies to mitigate changes in degradation of model and/or data accuracy and robustness

To mitigate **both model drift and data drift,** network retraining is often a solution, but not always feasible, especially in continuous learning contexts.

Characterize model deviation and adapt the model to the model [58], use dataset deviation strategies [60], or **out-of-distribution** (OOD) generalization or **zero-shot generalization** [61]. In order for the model to continue learning and adapting to the change in distribution, continuous learning strategies can be useful for data deflection [62] and for continuous learning with models that do not stop training. For example, the use of ***Elastic Weight Consolidation (EWC),*** *Piggyback* [112], ***Experience Replay models*** *(replay buffer, prototype experience replay, rehearsal), replay* with generative models (*pseudo rehearsal*), or *Memory networks*, among many others [25].

## Example - False Reporting Detection

In the design of the high-risk AI system, the provider has considered that, although the training base is broad (see examples in the accuracy guide), the model can be degraded by changes in the data present in the complaints over time and may suffer: data drift and domain displacement. The concept that is established is not that they can occur, but that they will occur, so it is necessary to mitigate them (as far as possible) and provide a feedback mechanism that expands the training base with the final evaluation of a human agent.

In order to mitigate data drift and domain displacement in complaints (both from their structured content: provinces, addresses, and from textual content), the domain *adaptation* technique proposed in this guide on training and verification/test complaint sets is used.

Since retraining the system is not feasible on an ongoing basis. The AI system is equipped with a process that allows agents to add the real veracity/falsity of a report in which the system has made an error, with this information the AI system prepares a depersonalized and anonymized training copy that can be incorporated into the training suites in subsequent future updates. This information is incorporated into the MLOps system described in this guide for this same AI system.

To mitigate data drift and domain shift, domain *adaptation* techniques [59] or *transfer learning can be used*.

When the data sets are visual, the potential occurrence of general, selection, frame, or label bias and all its subtypes should also be studied (Table 1 in [88] and associated list of requirements in the appendix).

To detect **causal relationships** that may be a **source of bias**, classifiers can be used to discern the presence of certain objects or attributes in a classifier, to detect selection bias [93], through the *Neural Causation Coefficient* (NCC, [92]).

- When you don't have access to the actual target: ***Confidence-based Performance estimation (CBPE)*** for classification models, and ***Direct Loss Estimation (DLE)*** *for regression problems.*
- Efforts should be made to establish mechanisms to detect silent failure of the model, for example, with warning systems. For example: NannyML claims to be the only open-source algorithm today capable of fully capturing the impact of data drift on accuracy, and Microsoft's AzureML with MLOps-level error diagnosis.
- To estimate causal relationships that may compromise robustness, the omitted variable bias that explains the dependent and independent variables leads to biased estimates. To do this, methods that reduce the risk of this bias can be used: variable instrument or *least squares estimator*, or *regression discontinuity design*.

To monitor the well-known phenomenon of **catastrophic forgetting** and the inability to learn continuously [25], it is recommended to use *continuous learning* or *lifelong learning* models, consider and report metrics for forgetting control and continuous learning monitoring, for example, ***Backward transfer, forward transfer, model size efficiency, samples storage size efficiency, and computational efficiency*** [28].

**Organizationally**, the AI system must implement functionality that allows the automatic computation, monitoring and notification by the **deployer** of the degradations to which the system may be exposed. Therefore, it will be the user's responsibility to know the system provided by the provider, and to access its technical documentation.

Regarding **technical** measures:

- The deployer is responsible for knowing, applying and keeping an eye on the model through the interfaces provided to transmit its accuracy, biases (according to the Data Guide) and uncertainty.
- Thus, you should use the instruction manuals of the interfaces of the tools that provide all the related metrics, to be in a position to interpret (see Transparency guide), verify that the output of the model corresponds to its accuracy and associated metrics, and that the explanation points out the expected correct reasons [113], in order to evaluate if the system is working correctly, without bias (avoiding for example hallucination of elements [114,115], see Biases in the Data Guide). Once the biases have been identified, it will act accordingly according to human oversight guide).

More information on measures to monitor and control the robustness of AI systems at ISOS:

- [SC42_N483_ISO_IEC_TR_24029-1] ISO/IEC TR 24029-1, Artificial Intelligence (AI) – Assessment of the robustness of neural networks - Part 1: Overview
- [ISO/IEC 24029-2:2023 SC42 N938 ISO IEC CD 24029-2] Artificial Intelligence (AI) – Assessment of the robustness of neural networks – Part 2: Methodology for the use of formal methods.
- (ISO/IEC 24029-2:2023 ISO /IEC JTC 1/SC 42/WG 3 Secretariat: ANSI Date: 2021-09-06).

# 5. Technical documentation

Just as important as robustness itself, its associated metrics and the mechanisms necessary to guarantee it, is its documentation. The technical documentation, as detailed in the corresponding guide, must contain at least the following information related to robustness:

- Instructions on how to obtain and interpret robustness measures and metrics, including their explanation and monitoring, shall be documented according to the Technical Documentation Guide for all potential users of the system who are required to monitor, develop, or intervene in the system.
- For reported robustness to be actionable, such documentation shall include the possible ranges of configurable model parameters, ranges of input and output data, as well as estimated latency and efficiency measures to achieve the established robustness.
- The minimum acceptable values for the applicable robustness metrics (and guaranteed by the provider in the documentation) shall reflect and serve as an indicator of the system's quality. When these values cannot be guaranteed, the provider shall make available mechanisms for the deployer to notify a human operator (according to the Human Oversight Guide) and/or enable possible interruption or shutdown of the system.
- Since robustness must be maintained throughout the entire life cycle of the system, the documentation shall be updated after each relevant modification of the model, retraining, or change in the operational context, reflecting the evidence and results of the new evaluations carried out.
- In addition, robustness documentation shall explain to the different deployers how the system operates in relation to common challenges among the principles of trustworthy AI. For example, by indicating the minimum guarantees the system provides regarding known trade-offs: between global group fairness and individual privacy (already demonstrated), or the potential trade-off between robustness and privacy. As these trade-offs between requirements have shown the possibility of being in conflict, the AI model card shall state the minimum expected guarantees, as well as those trade-offs that have been tested and may or may not be guaranteed.
- To test the model as a sandbox case, methodologies are recommended with which the user should become familiar, especially if they become a provider by reusing or modifying the product – such as canary deployment, where the model is put into production for a small segment of users, being monitored and controlled for possible issues identified through integration and regression testing.
- The Responsible AI Scorecard shall be reported together with the dataset sheets and model cards as a unifying basis for the trustworthy AI principles that affect the robustness of the system.

In any case, the Technical Documentation guide establishes how to organise this documentation and how to relate it to the measures described in this guide. To have a complete scenario and therefore a correct framework, it is recommended to consult both together to establish how to document the robustness of the system.

## 5.1 Certification of model robustness

Prior to commissioning and after the implementation of the model, the robustness of the model can be certified. Given the intended purpose of the AI system and the type of data it uses, it is recommended to look at the type of attacks and defences (in this, the cybersecurity guide reviews the applicable scenarios).

In terms of methodologies, and which can therefore help the provider to properly document the robustness of the AI system, mechanisms can be established to establish certain guarantees of robustness. To this end, new licenses and those under preparation can be considered:

- Hippocratic License https://firstdonoharm.dev/
- RAIL (https://www.licenses.ai)
- Ethicas foundation (certification/seal)

Examples:

- Stability Training *(79)* can be used *to stabilize the robustness of image processing models against distortions such as compression, re-scaling, or* cropping.
- To evaluate and strengthen natural language processing models, data augmentation and adversarial training can be used with *TextAttack* [12] [13].
- For objective-soft functions such as *population loss*, adversarial training can be used on principle, for example, considering a Lagrange penalty formulation perturbing the distribution of the data on a *Wasserstein ball*.

Depending on the stage of the life cycle, the certification of robustness requires different organizational measures.

1. **Operation and monitoring**. It is essential to assess the robustness of artificial intelligence (AI) systems in their operational scope and monitor their evolution. When the required level of robustness is not reached, corrective actions must be implemented, such as alerting the operator or activating safety modes to prevent failures. Formal methods can be applied in this context; however, they tend to demand greater processing, memory, and power resources compared to other approaches.
    - To make the explanations over time more robust, counterfactual data augmentation can be used to train the model with data samples and their factual counterfactuals and thus avoid unfortunate counterfactual ***events*** *(*UCEs [82]) or use plausible counterfactual events [85, 86], which not only provide better robustness than counterfactual explanations closer to the example to be explained, but they also imply greater individual fairness [85].
    - To explain the drift of input data in the sense of data differentials, ontologies and induction of concepts on background knowledge can be used [87].

2. **Re-evaluation**: Monitoring of validation tests and verification of EU ethical principles could be done through the requirements set by regularly updated licenses.

# 6. Self-assessment questionnaire

To carry out a self-assessment of compliance with the requirements of the Artificial Intelligence Act referred to in this guide, a global self-assessment questionnaire has been generated with a series of questions with the key points to be considered with respect to the obligations dictated by the articles of the AI Act mentioned in this guide.

It will be necessary to refer to this document in order to carry out the section of the self-assessment questionnaire corresponding to this guide

# 7. Annexes

## 7.1 Annex I: Metrics and Tests for AI systems

### 7.1.1 Metrics to Establish Robustness

In addition to those identified in the accuracy guide, specific metrics on the robustness of artificial intelligence models can be reported:

**Multiclass classification**:

- **Cohen's Kappa coefficient**: Measures agreement between scorers, adjusting for the probability of random agreement.
- **Confusion Matrix and Derived Metrics**: Provides a detailed view of model performance, including true positives, false positives, true negatives, and false negatives.
- **Hinge Loss**: Mainly used in Support Vector Machines (SVMs), it evaluates loss based on the classification margin.
- **Matthews Correlation Coefficient (MCC):** Provides a balanced measure of model performance, especially useful in imbalanced datasets.

Importantly, in imbalanced datasets, metrics such as Area Under the ROC Curve (AUROC) may not be adequate, as they do not correctly reflect the ratio of false positives to true positives.

**Binary classification**:

- **Precision-Recall Curve:** Useful when classes are unbalanced, as it focuses on the model's ability to correctly identify positive instances.
- **ROC curve**: Shows the relationship between the true positive rate and the false positive rate at different thresholds.
- **Balanced accuracy**: Averages sensitivity and specificity, providing a more balanced measure in cases of unbalanced classes.

**Multi-tag classification:**

- **F1 Score**: Combines accuracy and recall into a single metric, being especially useful when handling multiple tags per instance.

**Scoring systems:**

- **Area Under the Curve (AUC):** Assesses the model's ability to distinguish between classes.

- **Lift:** Measures the improvement of model predictions compared to a random strategy.

**Statistical methods:**

- **Mean Squared Error (RMSE):** Calculates the square root of the average of the squared errors, providing a measure of the magnitude of the error.
- **Maximum Error (MaxError):** Indicates the largest difference between the predicted and actual values.
- **Correlation between current and predicted values:** Evaluates the linear relationship between model predictions and actual values.
- **Cross-validation:** Includes techniques such as K-fold, Group K-fold, Leave-One-Out, Leave-One-Group-Out, and Monte Carlo cross-validation, which help assess the generalizability of the model.

**In addition, to measure the robustness associated with the accuracy of the model, the following can be used:**

- **AUROC:** Although it should be employed with caution on imbalanced datasets. [ISO SC42 N1011]
- **Cumulative Response Curve (CRC) or Gain Curve:** Alternatives to the ROC curve when the target is focused on a specific proportion of the database. [7]
- **Hamming Loss:** Measures the fraction of labels incorrectly predicted in multilabel classification problems; lower values indicate greater robustness.
- **5x2 cross-validation schemes:** Provide a more robust evaluation of model performance.

The selection of metrics should align with the specific goals of the model and the characteristics of the dataset, ensuring a fair and accurate assessment of its performance and robustness.

### 7.1.2 Test Methods

Statistical methods rely on mathematical testing procedures that can illustrate a certain level of confidence in the results. They allow engineers to verify whether system properties reach a target desired threshold. For example, how many predictions produced are flawed or inconsistent?

For this there are the following test methods:

- Formal methods rely on firm formal proofs to demonstrate a mathematical property over a domain. Formal methods include uncertainty analysis to test the stability of interpolation, using optimizers (**solvers**, e.g. an **SMT optimizer** to test the absence or existence of adversarial examples and the ability to exhibit them), optimization techniques (such as **Branch and Bound** or **constraint programming**) or abstract interpretations to prove a maximum **stable space property**.

- Empirical methods rely on experimentation, observation, and expert judgment. to evaluate the extent to which the system properties are true in the tested scenarios. Field trials that may use software testing standards (e.g.ej. ISO/IEC/IEEE 29119-3:2021, where the residual risk is indicated in relation to what was evaluated by the tests in order to find defects in the tested element), tested a posteriori (with input-output evaluation, or validation by testing).

Performing input perturbation tests, to consolidate the robustness of the system, is a testing mechanism that allows evaluating the behaviour of the AI system in production conditions, where inputs are less controlled. Here are three examples of disturbances according to the types of AI systems:

- To assess the robustness of a model against a specific disturbance (e.g., presence of liquid droplets causing image defects in lenses) using a formal method we must have a function describing the process of application of the disturbance that is based on a limited parameter that can be varied to present acceptable variations of perturbation, and it is preferable to have a commutative composition of functions. Other disturbances could be vibration, rotation, brightness, atmospheric turbulence, homogeneous noise, blurring, overexposure *(***blooming bright spots***)* or smear.
- In sound data, frequency ranges audible to humans or not (such as ultrasound-based disturbances) can be studied.
- In reinforcement learning (RL) you can: applying disturbing forces to the agent makes the agent more solid, influencing an agent to always take the worst possible action can be seen as an attack that leads to better strength of the trained agent in environments with different physical properties. Reducing an agent's accuracy or efficiency is also possible even if the adversary is only part of the victim's environment and therefore part of his or her observations. An example of technique in this case is **league training** [81].

#### 7.1.2.1 Formal methods

Formal methods are mathematical techniques for the rigorous specification and verification of software and hardware systems that aim to demonstrate or verify the correctness of the system. Formal methods allow the AI system to be matched to the intended purpose with a mathematical model. They can complement other methods and increase confidence in the robustness of models, reason about them and demonstrate whether they satisfy the relevant robustness properties.

Generally, they establish a link in the model (e.g., neural network) through the inputs to the outputs on a domain, which allows us to know how much each input impacts each output. There are several types of formal methods. A distinction is generally made between:

- **Complete** (provide exact answers) and **incomplete**. The latter may be more realistic for application to the verification of neural networks that achieve high accuracy in complex data tasks, as they use abstraction techniques that scale to them. However, they may fail to demonstrate a property of robustness that is satisfied.
- **Deterministic or not**. The latter can verify models such as **mixture density networks** or **variational autoencoders** that do not produce a deterministic output but

probability distribution. To demonstrate formal guarantees of its robustness with high probability, non-deterministic methods can establish the parameters of this output distribution deterministically.

- **White box** (internal-access) or **black box** (model agnostic). White-box verifiers require access to the network's internal representation, architecture, and learned parameters, but not to the training data or the algorithm used to train the neural network. Black box verifiers can be used when the model is encrypted and not accessible. However, they require the ability to run the model for specific inputs, which can make them less accurate than white-box ones.
- **Computer or real arithmetic**. Although in many cases this is not the case, verifiers can take into account the rounding errors made with computer arithmetic and reason, under the semantics established for that particular arithmetic, about the outputs of a neural network[6].

#### 7.1.2.2 Formal methods and their applicability for verifying the robustness of artificial neural network-based AI systems

The application of formal methods, as a mechanism for verifying the robustness, allows two formal properties of the robustness of a system to be established in neural networks:

- **Interpolation stability** calculates the uncertainty of a neural network as a way to detect when the network will have insufficient interpolation capacity and therefore insufficient robustness.
- Maximum stable space: Calculates the size of the domain where the neural network will have a stable classification prediction. Formal methods can be divided into the type of model to which they are applied:
    - Piecewise linear neural networks (PLNN) verification methods: The robustness of these neural networks can be verified with **satisfiability modulo theories** (SMT) solver optimizers, with mixed integer **linear program (MILP**) or others reflected in ISO /IEC DTR 24029:2021 such as **Fast-Lin – Fast-Lip, CROWN and formal safety analysis.**
    - Methods for verifying binarized neural networks (**BNNs**): Create an exact representation of the BNN with Boolean formulas such that all valid input and output pairs are solutions of this formula, and verification is achieved using Boolean satisfiability, **Integer Linear Programming (ILP**).
    - Verification methods through optimizers (**solvers**): **MILP**, **SMT** (Satisfiability Modulo Theories): All are **deterministic**, **white box**. They encode every computation of a neural network as a constraint and, depending on the architecture, can be complete or incomplete verifiers[7]. If the elements at the

[6] To prevent robustness assurances from being met for computations actually done with floating-point arithmetic or otherwise, formal verifiers consider the semantics of the type of computational arithmetic, standard or not, and ensure that the output captures the possible outputs of the network under that semantics. Only when operators round correctly as set forth in the IEEE 754 standard can verifiers account for changes in the order of the computations and can approximate the rounding done on each operator.

[7] For example, hyperbolic activation functions such as sigmoid and tanh are too complex to encode precisely, so optimizers approximate them with formal abstractions.

boundaries of the strength constraint satisfy the constraints, the property is demonstrated.

- Verifiers of Abstract Interpretation. Frameworks for the scalable analysis of large and complex probabilistic and deterministic systems that, in the context of neural networks, provide a deterministic, incomplete white-box method that can verify the robustness of large architectures through the abstraction of model inputs by means of geometric shapes, boxes, zonotopes, polyhedrons, or others, so there's an inherent trade-off between accuracy and scalability. For example, using boxes to constrain inputs scales to millions of neurons per second, but it is imprecise to verify robustness properties; while semi-defined relaxations are more precise, but do not scale to large architectures.

#### 7.1.2.3 Methods for formally verifying the robustness of other particular neural network models

Transformer **models** can be verified by decomposing layers into sublayers where limits are calculated that act as a guarantee from the first sublayer (of linear transformations, nonlinear functions and *self-attention* operations) to the last [99]. Verifying and validating the robustness therefore in each of the established stages.

In the case of recurrent neural networks (**RNNs**), these can be considered as infinite-state machines or finite-state automata, so the local robustness of the RNNs can be measured for classification with **model checking** and **abstract interpretation**. Abstract interpretation is a framework for the scalable analysis of large-scale, complex, and probabilistic deterministic systems that provides an incomplete, deterministic, white-box method that can verify the robustness of large neural networks.

In reinforcement learning **(RL)-based AI systems:** Stability and robustness should be considered when documenting the robustness of systems, testing both in simulation and in the real world [68], and using metrics adapted to the task and context [69, 105].

For other types of data such as graph-based neural networks they will require specific specialized methods[8].

## 7.2 Annex II: Robustness and uncertainty

In this Appendix we present the relationship between uncertainty and robustness for the AI system. The provider should consider this Annex in connection with the robustness assessment, as a detail and extent of the measures referred to in paragraph 2.

[8] For example, when you have tabular data that combine data of various types (numerical, symbolic, textual) and express relationships between them, this may prevent the use of formal methods due to the large number of rows and the difficulty of estimating the variance within each row.

### 7.2.1 Robustness as a capacity to deal with and minimise uncertainty

The reliability of the reported accuracy and robustness for an AI model depends on the reliability of the data and the estimation of the factors generating the data, its hypothesis in relation to the intended purpose, etc. [1]. If the data degrades (when the data it processes no longer follows the distribution of the data used when it was trained), the model will too. (see Data Guide).

A reliable representation of a model's uncertainty is key for any AI model. At least **two** main types of **uncertainty** are associated with a predictive model: 1) **Random**, and 2) **Epistemic**. Randomness is not reducible, as it is due to the inherent stochastic nature of the dependence between instances x and outputs y. Epistemics is reducible by the model: you could eliminate it by collecting more data for the worksets [1].

The provider shall provide infrastructure to monitor and report, throughout the life cycle of the AI system, potential:

1. Changes in the distribution of training data and test (***dataset shift***).
2. Disparities in domain change or machine learning task requiring adjustment to the new data domain by fine-tuning with weight freezing in neural networks, or with transfer learning processes.
3. **Changes** in the mechanism of acquiring **new data**, involving, for example, the introduction of unwanted artifacts, data that include indicators that may introduce a bias indirectly, or a simple imbalance of these.
4. Catastrophic forgetfulness and the inability to learn continuously [25].

In terms of **technical** measures, we detail in the following sections, different measures to deal with different dimensions that lead to the robustness of an AI system.

### 7.2.2 Methods for measuring and quantifying the uncertainty of the output of an AI model

Techniques for quantifying the uncertainty associated with a model include:

- Calibrate the classification models (with isotonic regression or **Sigmoid / Platt scaling**), i.e. use a post-processing technique in order to improve the estimation of probability or improve the error of the distribution.
- Assess uncertainty indirectly (assessing its usefulness for improved prediction and decision-making), e.g. with *accuracy-rejection curves* or other strategies [1].
- Using other calibration measures: **Macro-average accuracy, proportion of classes, MSE, LogLoss, CalBin, CalLoss, Avg error, relative error, Anderson-Darlin (A2) test** [2].

Tools for estimating and monitoring (see Human Supervision guide) of the degradation of model accuracy and robustness through the uncertainty associated with it:

- IBM Uncertainty Quantification 360 Toolkit (to estimate, communicate and use uncertainty in predictive models to improve both model accuracy and robustness).

- NannyML (to estimate model uncertainty after model is put into production, before and after the target output data is no longer available, detects data and model drift that possibly cause it).

## 7.3 Glossary

| Term | Definition |
|---|---|
| **Abstract interpretation** | It is a technique that simplifies program analysis by creating abstract representations that capture essential properties. It helps to understand and reason about the behaviour of programs without actually running them, making it easier to detect errors and optimize in artificial intelligence. |
| **Anderson-Darling (A2) test** | The Anderson-Darling test is a statistical test used to determine whether a dataset follows a specific distribution. It is used to evaluate whether the data fits into a theoretical distribution, such as the normal, exponential, or uniform distribution.<br><br>The test is based on the comparison of the observed values with the expected values under the theoretical distribution. It calculates an adjustment measure called the Anderson-Darling statistic, which takes into account the differences between observed and expected values, giving more weight to deviations at the tails of the distribution.<br><br>The null hypothesis of the test is that the data follow the theoretical distribution, while the alternative hypothesis is that the data do not conform to the theoretical distribution. If the value of the Anderson-Darling statistic is greater than a given critical value, the null hypothesis is rejected, and it is concluded that the data do not follow the theoretical distribution. |
| **Distilled Privileged Learning** | It is a machine learning technique that uses additional information or privileged knowledge available during the training stage to improve the performance of the final model. A secondary model is trained to learn how to predict this privileged information and is used to teach the base model to take advantage of it and improve its predictions. |
| **AUROC** | The AUROC is a metric that summarizes the quality of the ROC curve and provides a quantitative measure of model performance. The higher the AUROC value, the better the performance of the model. |

| Term | Definition |
|---|---|
| **Variational autoencoders** | They are machine learning models that generate latent representations following a specific distribution. These models allow you to reconstruct the original data and also generate new data similar to the originals. They are useful in data generation tasks and other applications where capturing the uncertainty associated with data generation is required. |
| **Mean error / relative error** | Mean error (ME) is a measure used to assess the systematic bias or trend of a model or estimate in predicting values. It is calculated by taking the average difference between the predictions and the actual values.<br>The ME is a measure that takes into account the direction of error. If the predictions are consistently higher than the actual values, the EM will be a positive number, indicating a positive bias. On the other hand, if the predictions are systematically lower, the EM will be a negative number, indicating a negative bias. Relative error (RE), on the other hand, provides a measure of error relative to actual value. Expressed as a percentage, it is useful for assessing the relative accuracy of the model or estimate at different data scales. The ER allows for a more direct comparison of accuracy between different models or estimates, regardless of the magnitude of the values. |
| **Backward transfer, forward transfer** | In the context of machine learning and model training, backward transfer and forward transfer refer to the influence that learning one task can have on the performance of other related tasks. Backward transfer occurs when learning from a previous task improves performance on subsequent tasks. That is, the knowledge acquired by learning a previous task helps to improve performance in subsequent tasks. On the other hand, forward transfer occurs when learning a subsequent task improves performance on previous tasks. In this case, the knowledge gained from learning a subsequent task influences the improvement of performance in previous tasks. These transfers can occur when tasks have a certain relationship or similarity in terms of characteristics or underlying structure. If the knowledge learned in a task can be effectively applied in related tasks, a transfer of learning occurs, either backwards or forwards. |
| **Blooming bright spots** | It refers to a visual phenomenon that can occur in digital images, especially in images captured by digital cameras or CCD (charge-coupled devices) sensors. It is related to the presence of noise in the images. |

| Term | Definition |
|---|---|
| **Branch and Bound** | It is a technique used to solve optimization problems by systematically exploring all possible solutions. It uses branching strategies to divide the search space and bounding techniques to eliminate unpromising solutions. The goal is to find an optimal solution or approach it efficiently. |
| **CalBin** | It is a calibration measure based on overlapping clustering. For each class, all cases must be ordered according to the prediction *p(i, j)* by reassigning the indices. The first 100 elements will be taken as the first bin, the percentage of cases of class *j* in this box will be calculated as the true probability, *fj*. $$\Sigma_{i\epsilon\lvert\ldots 100}\lvert p(i,j) - fj\rvert$$ The error for this bin is, Later the error will be calculated for the rest of the bins. Finally, the average of the errors will be calculated. |
| **CalLoss** | A perfectly calibrated classifier always gives a convex ROC curve. However, a classifier can produce very high AUC scores, but the probabilities may differ from the actual probabilities. One method of calibrating a classifier is to calculate the convex hull or, in other words, to use isotonic regression. Flach and Matsubara (2007) break down Brier's score into loss of calibration and loss of refinement. CalLoss is defined as the mean squared deviation of the empirical probabilities derived from the slope of the ROC segments. |
| **Cohen Kappa** | It is a measure of agreement, used to evaluate the agreement between two evaluators or classifiers. It provides a more robust measure than simple match in ranking and takes into account the expected agreement by chance. A kappa value close to 1 indicates a high degree of agreement, while a value close to 0 indicates agreement similar to that expected by chance. |
| **Computational efficiency** | Computational efficiency in machine learning refers to the ability of machine learning algorithms and models to perform tasks quickly and efficiently in terms of computational resources, such as runtime and storage capacity. In the context of machine learning, computational efficiency is an important aspect due to the increasing complexity of datasets and models. The aim is to optimize the use of computational resources to achieve a balance between the performance of the model and the resources required. |

| Term | Definition |
|---|---|
| **Confidence-based Performance estimation (CBPE)** | It is a technique that uses the confidence or probability measures assigned by a classification model to evaluate its performance. Setting a confidence threshold calculates the accuracy of the model only for predictions with a confidence measure above that threshold, allowing for a more accurate estimate of model performance in critical situations. |
| **Correlating controls and measures** | Control-measurement correlation refers to the relationship or dependence that exists between the values of the controls used in a system and the resulting measurements. In the context of control and monitoring systems, it is important to understand how changes in controls affect measurements and vice versa.<br><br>When there is a strong correlation between controls and measurements, it means that changes in control values have a significant impact on the measurements obtained. This may be desirable in some cases, as it indicates that controls have a predictable effect and can be used to influence measurements in a controlled manner.<br><br>However, there may also be situations where a strong correlation between controls and measurements is not desirable. For example, if there are external factors that can affect measurements independently of controls, a strong correlation can make it difficult to identify the true cause of variations in measurements.<br><br>Understanding the correlation between controls and measurements is essential in many fields, including engineering, physics, and data science. It allows systems to be properly analysed and modeled, causal relationships to be identified, controls to be optimized to achieve desired results, and to understand the behaviour of systems under different conditions. |
| **Coverage tests** | These are techniques used to evaluate what part of the source code has been executed during testing. There are different types of coverage, such as line coverage, branch coverage, condition coverage, and road coverage, which focus on different aspects of the code. Coverage testing is useful for measuring and improving software quality, although it does not guarantee the absence of errors. |

| Term | Definition |
|---|---|
| **Crammer-Singer** | It is an iterative method used to solve the multiclass classification problem. Through a series of transformations and optimizations, it seeks to maximize separability between classes in a feature space. This approach avoids the need to decompose the problem into independent binary classifiers and can improve classification performance by considering additional information. |
| **CROWN** | Acronym for "Convex Relaxation of Nonlinear layer", it is an approach used in the field of machine learning and neural networks. It is an approach that uses convex mathematical programming to approximate the forward propagation function of a nonlinear neural network using a convex formulation. It provides upper and lower limits for neural network output based on input boundaries, allowing robustness analysis to be performed and safety properties to be certified for the model. |
| **Data differentials** | Differences or changes in data between different points or moments in time. They are a measure used to analyse how data varies or differs in different situations and can provide valuable insights into trends, patterns, and changes in data. |
| **Data shift** | Change in the distribution of data between the training set and the test set or between different points in time. It can negatively impact model performance and requires specific approaches to address the issue and mitigate its impact. |
| **Dataset shift** | It occurs when there is a discrepancy or change in the distribution of data between related datasets. It can have a negative impact on model performance and requires specific techniques to mitigate its effect and improve model generalization. |
| **Distillation** | It refers to transferring knowledge from a larger, more complex master model to a smaller, more simplified student model. It is used to train the learner model more efficiently and effectively, using soft labels generated by the master model rather than categorical labels. This allows the student model to capture the knowledge and underlying structure of the data and generalize better. |
| **Direct Loss Estimation (DLE)** | It is an approach in machine learning that seeks to directly estimate the loss function associated with a problem rather than using the conventional loss function. The goal is to obtain more accurate estimates of loss and improve the performance of the model in specific tasks, taking into account particular aspects of the problem. |

| Term | Definition |
|---|---|
| **Elastic Weight Consolidation (EWC)** | It is a method used in machine learning to mitigate the catastrophic forgetting of previously learned knowledge when training models on new tasks. It selectively protects weights important for previous tasks by means of a penalty function that prevents drastic changes in those weights during training on new tasks. This allows the model to preserve and efficiently use the previous knowledge acquired. |
| **Ensemble Learning** | It is a focus on machine learning that combines multiple learning models to improve accuracy and overall performance. By taking advantage of the diversity of the models and combining their individual predictions, a more robust and reliable final prediction is obtained. This approach is based on the premise that the combination of independent models can overcome individual constraints and improve generalizability. |
| **Experience Replay (replay buffer, prototype experience replay, rehearsal)** | Experience Replay, also known as replay buffer, prototype experience replay, or rehearsal, is a technique used in training deep learning models, especially in reinforcement learning algorithms.<br><br>The core idea of Experience Replay is to store and reuse past experiences during training rather than using only the most recent experiences. This is accomplished by storing experience transitions, which consist of pairs of states and actions taken in an environment, in a storage memory called a replay buffer. During training, instead of using only the most recent transitions, the transitions stored in the replay buffer are randomly sampled. This allows the deep learning model to re-experience a variety of past situations, helping to avoid over-reliance on the most recent experiences and improve training efficiency and stability. There are several benefits to using Experience Replay. First, it helps to decorate experience transitions, as adjacent experiences in time can be highly correlated. This prevents the model from being trained only in similar situations and improves its ability to generalize to different situations. In addition, the replay buffer allows for more effective exploration, as transitions are randomly sampled, and suboptimal behaviours can be avoided. There are variations of Experience Replay, such as Prototype Experience Replay, which focuses on storing and using representative examples or prototypes to improve model performance. |

| Term | Definition |
|---|---|
| **F1-Score** | The F1 score is a statistical measure that combines the accuracy and recall of a classification model. It is useful when you have unbalanced classes in your data. It provides a single metric that represents accuracy and recall in a balanced manner, and its value ranges from 0 to 1, with 1 being the best possible result. A high F1 score indicates a balance between accuracy and model recall in the class rankings. |
| **Fast-Lin** | Computationally efficient algorithm. It calculates a certified lower bound on the input disturbances allowed for ReLU networks using a layer-by-layer approach and binary search in the input domain. |
| **Fast-Lip** | Computationally efficient algorithm. It relies on Fast-Lin to calculate the limits of the activation functions and, in addition, estimates the local Lipschitz constant of the network. In general, Fast-Lin is more scalable than Fast-Lip, while Fast-Lip provides better solutions for l1 limits. |
| **Formal safety analysis** | A formal analysis of neural networks is required due to the risks associated with erroneous predictions in adverse situations. This approach makes it possible to verify security properties and find concrete counterexamples in larger networks, which significantly improves the existing analysis capacity. In addition, this approach can contribute to improving the explainability and robustness of neural networks. |
| **Group K fold cross validation** | It is a model evaluation technique that takes into account the group structure of the data. It divides the data into predefined groups and evaluates the performance of the model in test and training sets that are representative of different groups. This allows for a more accurate assessment of model performance in real-world situations where the data has a group structure. |
| **K-fold cross validation** | It is a technique that divides data into k folds and performs k iterations to evaluate and select models. It provides a more accurate estimate of model performance and helps avoid over- or under-fitting issues by using all data more effectively. |
| **League training** | It is a training technique in which multiple agents are trained in parallel and compete against each other in a game environment. Through internal competition, agents learn and improve their strategies with the goal of achieving optimal performance in the league. |

| Term | Definition |
|---|---|
| **Leave-one-group-out cross validation** | It's a cross-validation technique that leaves out an entire group in each iteration as a test set. It is useful when the data has a structure of related groups and allows you to evaluate the performance of the model taking into account the specific characteristics of each group. |
| **Leave-one-out cross validation** | It is a cross-validation technique in which a single sample is left out as a test set in each iteration. It allows for a thorough evaluation of the model using all available data, although it can be computationally expensive. |
| **LogLoss** | Also known as logarithmic loss or cross-entropy loss, it is a common evaluation metric for binary classification models. It measures the performance of a model by quantifying the difference between the predicted probabilities and the actual values. Logarithmic loss is indicative of how close the prediction probability is to the corresponding true/true value (0 or 1 in the case of binary classification), penalizing inaccurate predictions with higher values. Lower record loss indicates better model performance. Log Loss is the most important probability-based ranking metric. It's difficult to interpret raw record loss values, but record loss is still a good metric for comparing models. For any given problem, a lower logarithmic loss value means better predictions. |
| **Macro-average accuracy** | It is a metric used to evaluate the performance of a classification model in multiclass problems. Instead of calculating the accuracy for each class individually and averaging the results, the macro average accuracy calculates the accuracy per class and then performs an unweighted average of these accuracies. It takes into account the performance of all classes equally and is not affected by class imbalance in the data. Each class contributes equally to the calculation of the average macro accuracy, making it suitable when you want to evaluate the overall performance of the model across all classes equally. |
| **Matthew's correlation coefficient** | It is a metric used to evaluate the performance of a binary classification model. It provides a measure of the balance between true positives, true negatives, false positives, and false negatives. It's a useful metric when working with imbalanced datasets, where positive and negative classes have different proportions. Unlike accuracy or recall, MCC is not affected by class imbalance and provides a more balanced assessment of model performance. |
| **MaxError** | ME or Maximum Error is the absolute value of the most significant difference between a predicted variable and its actual value. |

| Term | Definition |
|---|---|
| **Maximum stable space** | It is the set of data points that are robust against disturbances and do not undergo significant changes in their labels or classifications. This concept is useful for understanding the stability and robustness of machine learning models and can be used to improve the generalizability and adaptability of machine learning algorithms. |
| **Maximum stable space property** | The property of Maximum Stable Space states that an algorithm is capable of maintaining data stability by preserving the classifications or labels of data points even in the presence of disturbances. This property is important to ensure the robustness and generalizability of machine learning models under different scenarios and conditions. |
| **SQuaRE Method** | It is known Risks as a Semi-Quantitative Risk Assessment. It uses a combination of qualitative and quantitative analysis to assess and classify risks. It provides a structured way to identify, analyse and prioritise risks, taking into account both their impact and their likelihood. This approach helps organizations make informed decisions about how to manage and mitigate identified risks. |
| **Mixture density networks** | They are a neural network architecture used in probability distribution modelling problems. Instead of predicting a single value or classification, MDNs are capable of modelling and predicting complex probability distributions. They are useful in applications such as text generation, speech synthesis, time series prediction, and regression problems where complex probability distributions need to be modelled and estimated. They provide a flexible and powerful way to represent uncertainty in neural network predictions. |
| **Model checking** | It is a formal verification technique used in computer science to analyse and verify the correctness of concurrent systems or software systems. It consists of automatically verifying whether a formal model of the system complies with certain specified properties. This technique is particularly useful for critical systems where correctness and verification are of paramount importance, such as in the design of electronic circuits, communication protocols, embedded systems, and safety-critical software. Model checking provides a rigorous and automatic way to ensure the correctness of systems and detect potential problems before they are deployed or deployed. |

| Term | Definition |
|---|---|
| **Model size efficiency** | It refers to the ability to achieve optimal or acceptable performance using a small size model. In other words, it is about obtaining results comparable to or close to those obtained with larger models, but with fewer parameters or computational resources. |
| **Monte Carlo cross-validation** | It is an approach that combines traditional cross-validation with repeated random sampling to evaluate the performance of a model. It helps to get a more reliable estimate of model performance by considering multiple random partitions of the data. |
| **MSE** | It is an evaluation metric that measures the average of the squared differences between predictions and actual values in a regression problem. It is widely used and penalizes large errors more but can be sensitive to the scale of the data. |
| **ONNX** | ONNX (Open Neural Network Exchange) is an open-source machine learning model exchange format, which provides a standard format for describing the structure of the model and associated parameters, allowing developers to train models in one framework and use them in another without having to recreate the model from scratch. It was developed collaboratively by Microsoft and Facebook to facilitate interoperability between different machine learning frameworks and tools. |
| **SMT Optimizer** | It is a software tool used in the realm of formal verification and constraint satisfaction troubleshooting. It combines the capabilities of constraint satisfaction solvers (SATs) with the ability to reason about specific theories, such as arithmetic theories, set theories, list theories, array theories, among others. SMT optimizers are widely used in hardware and software verification, driver synthesis, automated planning, program optimization, and other domains where it is necessary to reason about constraints and combined theories. |
| **Out-of-distribution** | Discovery of out-of-distribution data in deep neural networks is an area of research that focuses on identifying and detecting data that is significantly different from a model's training data. This is important because out-of-distribution data can lead to incorrect or unreliable predictions. |

| Term | Definition |
|---|---|
| **Piecewise linear neural networks -PLNN** | They are an architecture of neural networks that use linear neurons in each linear segment of the input space. Although they are simpler and easier to train, they can struggle to model complex nonlinear relationships. They are suitable for applications where a linear approximation is sufficient or when a simpler and more efficient model is desired. |
| **PLEM o MILP (mixed integer linear program)** | It is a mathematical optimization approach that combines continuous and integer variables in a linear programming model. It allows you to address discrete decision-making and optimization problems, where some variables must be integer. |
| **Differential Progress** | It is a neural network training approach that uses small, incremental adjustments to model parameters to achieve faster and more stable convergence. It is an efficient and effective technique for optimizing machine learning models. |
| **Proportion of classes** | Indicates the relative distribution of different classes relative to the total samples or instances in a dataset. |
| **RMSE** | It is a metric that quantifies the average error between the predictions of a regression model and the actual values of the dataset. It is used to evaluate and compare the accuracy of different models, with the lowest possible RMSE value being desirable. |
| **Safeguard based design** | A design approach that focuses on embedding safeguards and safety mechanisms early on in AI systems, with the primary goal of ensuring that AI systems are safe, ethical, and reliable, and minimizing the risks associated with their implementation. This design involves identifying potential problems, risks, and vulnerabilities inherent in systems and taking steps to mitigate them. This can include implementing security controls, privacy measures, risk assessments, and rigorous testing. In addition, ethical and legal aspects must be considered when designing these systems, such as fairness, transparency, and accountability. The safeguard-based design approach is based on the principle that it is more effective and less expensive to address safety and ethics issues in the early stages of development, rather than trying to fix them once the system is up and running. By integrating safeguards from the initial design, the aim is to prevent possible unintended or harmful consequences and ensure that the systems are safe and reliable for use in different applications. |

| Term | Definition |
|---|---|
| **Samples storage size efficiency** | It refers to the ability to reduce the space needed to store samples without significantly compromising the performance or quality of the model. This is important because datasets can be massive, take up a lot of storage space, and require significant computational resources to process. |
| **Satisfiability modulo theories (solver)** | Satisfaction modulus theories (SMT) refer to the problem of determining whether a first-order formula is satisfactory with respect to some logical theory.<br><br>SMT-based solvers are used as back-end engines in model verification applications, such as bounded model verification, interpolation-based, and predicate abstraction. |
| **Solvers** | Solvers are specialized programs or algorithms that solve mathematical optimization problems by finding optimal or suboptimal solutions through specific techniques and algorithms. They are fundamental tools to address optimization problems in various fields, including economics, logistics, engineering, among others. |
| **Domain adaptation techniques** | Deep domain adaptation allows us to transfer the knowledge learned by a particular DNN in a source task to a new objective-related task. It has been successfully applied in tasks such as image classification or style transfer. In a sense, deep domain adaptation allows us to get closer to human-level performance in terms of the amount of training data needed for a particular new machine vision task. Therefore, I believe that progress in this area will be crucial for the entire field of computer vision, and I hope that it will eventually lead us to an effective and simple reuse of knowledge through visual tasks. |

| Term | Definition |
|---|---|
| **Unfortunate counterfactual events** | It refers to events that could have happened but didn't. These events are called "counterfactual" because they are hypothetical or counterfactual situations that are raised to evaluate the impact of alternative decisions or actions.<br><br>In the context of machine learning and decision-making, "unfortunate counterfactual events" refer to situations in which a negative or undesired outcome occurs as a result of a decision made or an action taken. These unwanted events can be the result of various circumstances, such as a machine learning model that is not properly trained or has not captured all possibilities, insufficient or unbalanced training data, or simply the uncertainty inherent in the decision-making process.<br><br>The study of "unfortunate counterfactual events" is important in machine learning to evaluate and understand errors or unintended results of models, and to identify possible improvements or solutions. By analysing and considering counterfactual scenarios, one can learn about alternative actions or decisions that could have led to better outcomes and use this knowledge to improve models, algorithms, and decision-making strategies in the future. |
| **Zero-shot generalization** | Zero-shot generalization refers to the ability of a machine learning model to apply its prior knowledge and understand entirely new tasks or domains without having been specifically trained in those domains. It allows models to use the transferable knowledge gained in training to perform tasks beyond specific training examples. |

## 7.4 Appendix: Critical Translations (English-Spanish)

- Performance: *Precision*, given the closest contextual meaning typically used in ML/AI (it could also be interpreted as *accuracy*), although it is usually translated as *performance* or *efficiency* in more general computer science contexts.
- Accuracy: *Accuracy* or *hit rate*, defined as (TP+TN)/(TP+TN+FN+FP)
- Precisión: *Precision*, *positive predictive value*; the fraction of retrieved instances that are relevant.
- Recall (*hit rate* or *TPR*): *Sensitivity* or *recall*; the fraction of relevant instances that have been retrieved.
- Confidence, Trust, and Reliance (the latter sometimes translated as *dependence*): all are translated as *confidence*.
- Trade-off: *Compromise*.

# 8. References, Standards and Norms

## 8.1. References

[1] Hullermeier et al. 2019 Randomic and Epistemic Uncertainty in Machine Learning: A Tutorial Introduction.
[2] Calibration of Machine Learning Models. Antonio Bella, Cèsar Ferri, José Hernández-Orallo, and María José Ramírez-Quintana. Department of Computer Systems and Computing, Polytechnic University of Valencia 2010.
[3] Metrics for Deep Generative Models N Chen · 2018 ·
[4] Language Models are Few-Shot Learners T Brown et al. 2020
[5] Physically-Consistent Generative Adversarial Networks for Coastal Flood Visualization Bjorn Lutjens et al. 2021
[6] IBM Uncertainty Quantification 360 Toolkit https://uq360.mybluemix.net
[7] Questioning causality on sex, gender and COVID-19, and identifying bias in large-scale data-driven analyses: the Bias Priority Recommendations and Bias Catalog for Pandemics. Díaz-Rodríguez et al. 2021 https://arxiv.org/abs/2104.14492
[8] https://catalogofbias.org
[9] A survey on datasets for fairness-aware machine learning Tai Le Quy∗1, Arjun Roy†12, Vasileios Iosifidis‡1, Wenbin Zhang§3, and Eirini Ntoutsi 2022
[10] Datasheets for Datasets. Timnit Gebru et al. 2021
[11] "Searching for a Search Method: Benchmarking Search Algorithms for Generating NLP Adversarial Examples",
[12] EMNLP 2020 Blackbox NLP Workshop track proceedings. https://github.com/QData/TextAttack
[13] TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP, J Morris et al. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020 [In Python]
[14] PLENARY: Explaining black-box models in natural language through fuzzy linguistic summaries K Kaczmarek-Majer, G Casalino, G Castellano... - Information ..., 2022 - Elsevier
[15] Companies Committed to Responsible AI: From Principles towards Implementation and Regulation? Paul B. de Laat Philosophy & Technology volume 34, pages 1135–1193 (2021)[9]
[16] COMPAS Risk Scales: Demonstrating Accuracy Equity and Predictive Parity 2016
[17] Joy Buolamwini and Timnit Gebru. 2018. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In Proceedings of the 1st

[9] Note: SHAP is not from Amazon nor proprietary in Table 3 is Scott Lundberg's creation with a MIT open source license, in Github, and was developed with Microsoft Research teams. AzureML also implements XAI algorithms and Amazon has as ML tool, Amazon SageMaker).

Conference on Fairness, Accountability and Transparency (Proceedings of Machine Learn Research), Sorelle A. Friedler and Christo Wilson (Eds.), Vol. 81. PMLR, New Ing

[18] An Ontology for Fairness Metrics. Franklin et al. https://dl.acm.org/doi/pdf/10.1145/3514094.3534137

[19] Human-centred artificial intelligence https://scilog.fwf.ac.at/en/environment-and-technology/15317/human-centred-artificial-intelligence

[20] A Holzinger et al. Digital Transformation in Smart Farm and Forest Operations Need Human-Centered AI: Challenges and Future Directions

[21] Holzinger, A. 2016. Interactive Machine Learning for Health Informatics: When do we need the human-in-the-loop? *Brain Informatics,* 3, (2), 119-131, doi:10.1007/s40708-016-0042-6.

[22] Holzinger, A., Malle, B., Kieseberg, P., Roth, P.M., Müller, H., Reihs, R. & Zatloukal, K. 2017. Towards the Augmented Pathologist: Challenges of Explainable-AI in Digital Pathology. arXiv:1712.06657.

[23] Holzinger, A., Malle, B., Kieseberg, P., Roth, P. M., Müller, H., Reihs, R. & Zatloukal, K. 2017. Machine Learning and Knowledge Extraction in Digital Pathology needs an integrative approach. Springer Lecture Notes in Artificial Intelligence Volume LNAI 10344. Cham: Springer International, pp. 13-50. doi: 10.1007/978-3-319-69775-8_2

[24] Human-Centered Artificial Intelligence for Designing Accessible Cultural Heritage G Pisoni, N Díaz-Rodríguez, H Gijlers, L Tonolli Applied Sciences 11 (2), 870

[25] Continual Learning for Robotics: Definition, Framework, Learning Strategies, Opportunities and Challenges T Lesort, V Lomonaco, A Stoian, D Maltoni, D Filliat, N Díaz-Rodríguez Information Fusion 220

[26] 2020 A survey on ontologies for human behavior recognition ND Rodríguez, M.P. Cuéllar, J Lilius, M.D. Calvo-Flores ACM Computing Surveys (CSUR) 46(4), 1-33 219 2014 A fuzzy ontology for semantic modelling and recognition of human behaviour ND Rodríguez, MP Cuéllar, J Lilius, MD Calvo-Flores Knowledge-Based Systems 66, 46-60 133 2014

[27] Explainability in Deep Reinforcement Learning A Heuillet, F Couthouis, N Díaz-Rodríguez Knowledge-Based Systems 214, 106685 119 2021

[28] Don't forget, there is more than forgetting: new metrics for Continual Learning N Díaz-Rodríguez, V Lomonaco, D Filliat, D Maltoni NeurIPS workshop on Continual Learning 2018

[29] Information fusion as an integrative cross-cutting enabler to achieve robust, explainable, and trustworthy medical artificial intelligence A Holzinger, M Dehmer, F Emmert-Streib, R Cucchiara, I Augenstein, ... Information Fusion 79, 263-278 2022

[30] Personas for Artificial Intelligence (AI) An Open Source Toolbox A Holzinger, M Kargl, B Kipperer, P Regitnig, M Plass, H Müller IEEE Access 10, 23732-23747 2022

[31] Measuring the quality of explanations: the system causability scale (SCS) A Holzinger, A Carrington, H Müller KI-Künstliche Intelligenz 34 (2), 193-198

[32] Interdisciplinary Research in Artificial Intelligence: Challenges and Opportunities R Kusters, D Misevic, H Berry, A Cully, Y Le Cunff, L Dandoy, ... Frontiers in Big Data 3, 45 2020

[33] S. Stanton, P. Izmailov, P. Kirichenko, A. A. Alemi, A. G. Wilson, Does knowledge distillation really work? (2021). doi:10.48550/ARXIV.2106.05945. URL https://arxiv.org/abs/2106.05945
[34] A Neural-Symbolic learning framework to produce interpretable predictions for image classification, PhD Thesis 2022.
[35] Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. AB Arrieta, N Díaz-Rodríguez, J Del Ser, A Bennetot, S Tabik, A Barbado, ... Information Fusion 58, 82-115
[36] Wilcoxon, Frank. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*. 1945, 1 (6), 1095 pages 80-83.
[37] Dietterich et al. Approximate Statistical Tests for Comparing Supervised Classification 1092 Learning Algorithms. *Neural Computation, Volume 10, Issue 7*. 1998, 10 (7), pages 1895-1923. 1093 https://doi.org/10.1162/089976698300017197
[38] Akenine-Möller, Tomas, and Johnsson, Björn. Performance per what? Journal of *Computer Graphics* 1073 *Techniques*. 2012, **1**, pages 37-41. http://jcgt.org/published/0001/01/03/paper.pdf
[39] Blouw, Peter and Xuan Choo and Hunsberger, Eric and Eliasmith, Chris. Benchmarking keyword 1075 spotting efficiency on neuromorphic hardware. *Proceedings of the 7th Annual Neuro-inspired* 1076 *Computational Elements Workshop*. 2019, pages 1-8. https://arxiv.org/pdf/1812.01739.pdf
[40] Suffering-focused AI safety: In favor of "fail-safe" measures Lukas Gloor Center on Long-Term Risk Report
[41] Superintelligence as a Cause or Cure for Risks of Astronomical Suffering
[42] Kaj Sotala and Lukas Gloor Foundational Research Institute, Berlin, Germany Superintelligence as a Cause or Cure ... Informatica 41 (2017) 389–400
[43] Safe Deep RL in 3D environments using human feedback. 2022.
[44] Safeguard By Design Lessons Learned from DOE Experience Integrating Safety in Design
[45] Sustainability at scale: towards bridging the intention-behavior gap with sustainable recommendations S Tomkins, S Isley, B London, L Getoor - Proceedings of the 12th ACM conference on ..., 2018
[46] Green AI. Schwartz et al.
[47] Distilling the Knowledge in a Neural Network
[48] EVALUATION METRICS FOR LANGUAGE MODELS Stanley Chen, Douglas Beeferman, Ronald Rosenfeld.
[49] Chip Huyen, "Evaluation Metrics for Language Modeling," The Gradient, 2019. https://thegradient.pub/understanding-evaluation-metrics-for-language-models/
[50] Model cards for model reporting. M Mitchell, S Wu, A Zaldivar, P Barnes, L Vasserman... - Proceedings of the ..., 2019
[51] Frank McSherry Materialize: a platform for building scalable event based systems
[52] Frank McSherry, Kunal Talwar Mechanism design via Differential Privacy, 2008
[53] Nobel Prize Report, Mechanism Design. 2007 Scientific background on
[54] the Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel 2007 Mechanism Design Theory, based on:
[55] L. Hurwicz & S. Reiter (2006) Designing Economic Mechanisms, p. 30

[56] https://divedeep.ai/2022/03/17/data-drift-vs-concept-drift/
[57] University of Oxford researchers have created a tool called capAI, a procedure for conducting conformity assessment of AI systems in line with the EU Artificial Intelligence Act. CapAI provides organizations with practical guidance on how to translate high-level ethics principles into verifiable criteria that help shape the design, development, deployment and use of ethical AI. This tool can be used to demonstrate that the development and operation of an AI system are trustworthy. The tool is being validated with firms at the moment and the most up-to-date version can be found here
[58] A survey on concept drift adaptation ACM computing surveys (CSUR), 46(4):1–37, 2014, Gama et al.
[59] Analysis of representations for domain adaptation Neurips 2007, Ben-David et al
[60] Dataset Shift in Machine Learning Quiñonero-Candela et al 2022
[61] Generalized out-of-distribution detection: A survey. Yang et al 2022
[62] Understanding Continual Learning Settings with Data Distribution Drift Analysis" Lesort et al. 2022 video: https://www.youtube.com/watch?v=WFhozvAgnsU
[63] Sagemaker Clarify: Amazon AI Fairness and Explainability Whitepaper https://pages.awscloud.com/rs/112-TZM-766/images/Amazon.AI.Fairness.and.Explainability.Whitepaper.pdf
[64] https://aws.amazon.com/blogs/machine-learning/learn-how-amazon-sagemaker-clarify-helps-detect-bias/
[65] Data Privacy and Trustworthy Machine Learning. Strobel et al. 2022
[66] Gradual (In)Compatibility of Fairness Criteria Corinna Hertweck, Tim Räz 2022 https://arxiv.org/abs/2109.04399
[67] Decoupling feature extraction from policy learning: assessing benefits of state representation learning in goal-based robotics A Raffin, A Hill, R Traoré, T Lesort, N Díaz-Rodríguez, D Filliat ICLR 2019 Workshop on Structure & Priors in Reinforcement Learning (SPIRL) 2019
[68] Continual reinforcement learning deployed in real-life using policy distillation and sim2real transfer RT Kalifou, H Caselles-Dupré, T Lesort, T Sun, N Diaz-Rodriguez, D Filliat ICML Workshop on Multi-Task and Lifelong Learning 2019
[69] S-RL Toolbox: Environments, Datasets and Evaluation Metrics for State Representation Learning
[70] A Raffin, A Hill, R Traoré, T Lesort, N Díaz-Rodríguez, D Filliat NeurIPS workshop on Deep Reinforcement Learning 2018
[71] Deep Unsupervised state representation learning with robotic priors: a robustness analysis
[72] T Lesort, M Seurin, X Li, N Díaz-Rodríguez, D Filliat. 2019 International Joint Conference on Neural Networks (IJCNN) 2017
[73] Information fusion as an integrative cross-cutting enabler to achieve robust, explainable, and trustworthy medical artificial intelligence A Holzinger, M Dehmer, F Emmert-Streib, R Cucchiara, I Augenstein, et al. Information Fusion.
[74] Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study Zech et al 2018 (extended from Confounding variables can degrade generalization performance of radiological deep learning models. Zech et al. 2018.)

[75] ISO/IEC 25000, Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE and ISO/IEC 25059:2023, Software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Quality model for AI systems.
[76] STRIDE-AI: An Approach to Identifying Vulnerabilities of Machine Learning Assets Lara Mauri et al. IEEE CSR 2021
[77] Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks Papernot 2016
[78] Steps Toward Robust Artificial Intelligence Thomas G. Dietterich 2017
[79] Improving the Robustness of Deep Neural Networks via Stability Training
[80] SECURING MACHINE LEARNING ALGORITHMS. December 2021 ANNEX D: REFERENCES by input data type and lifecycle stages.
[81] Towards Resilient Artificial Intelligence: Survey and Research Issues
[82] The Robustness of Counterfactual Explanations Over Time, A Ferrario et al.
[83] Research priorities for robust and beneficial artificial intelligence.

[84] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. 2016.
[85] Evaluating Robustness of Counterfactual Explanations Artelt et al 2021
[86] Exploring the Trade-off between Plausibility, Change Intensity and Adversarial Power in Counterfactual Explanations using Multi-objective Optimization. Javier Del Ser et al. 2020
[87] Towards Human-Compatible XAI: Explaining Data Di erentials with Concept Induction over Background Knowledge. Widmer et al 2022
[88] A survey on bias in visual datasets 2022. Fabrizzi et al.
[89] Omitted variable bias: A threat to estimating causal relationships. Wilms et al.
[90] Google. Machine Learning Glossary: Fairness. 2021 [cited 29 November, 2021]; available
[91] from: https://developers.google.com/machine-learning/glossary/fairness.
[92] David Lopez-Paz, Krikamol Muandet, Bernhard Scholkopf, and Ilya O. Tolstikhin. Towards a learning theory of cause-e ect inference. In Francis R. Bach and David M. Blei, editors, Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, volume 37 of JMLR Workshop and Conference Proceedings, pages 1452{1461. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/ lopez-paz15.html.
[93] David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Scholkopf, and Leon Bottou. Discovering causal signals in images. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 58{ 66, 2017. doi: 10.1109/CVPR.2017.14.
[94] ICO, Guidance on the AI auditing framework: draft guidance for consultation. Information Commissioner's Office, 2020.
[95] PwC, PwC Ethical AI Framework. 2020.
[96] Deloitte, Deloitte introduces trustworthy AI framework to guide organizations in ethical application of technology. August 26, 2020. New York.
[97] Orcaa, It's the age of the algorithm and we have arrived unprepared. 2020.

[98] [Epstein18] Epstein, Z., et al., Turingbox: an experimental platform for the evaluation of AI systems. IJCAI International Joint Conference on Artificial Intelligence, 2018. 2018-July: p. 5826-5828. #Discontinued.
[99] Shi et al. Robustness Verification for Transformers. International Conference on Learning Representations. 2020. arXiv:2002.06622
[100] Incremental Bounded Model Checking of Artificial Neural Networks in CUDA Luiz H. Sena et al.
[101] Decoupling feature extraction from policy learning: assessing benefits of state representation learning in goal based robotics A Raffin, A Hill, R Traoré, T Lesort, N Díaz-Rodríguez, D Filliat ICLR 2019 Workshop on Structure & Priors in Reinforcement Learning (SPIRL) 2019
[102] Continual reinforcement learning deployed in real-life using policy distillation and sim2real transfer RT Kalifou, H Caselles-Dupré, T Lesort, T Sun, N Diaz-Rodriguez, D Filliat ICML Workshop on Multi-Task and Lifelong Learning 2019
[103] A Raffin, A Hill, R Traoré, T Lesort, N Díaz-Rodríguez, D Filliat
[104] NeurIPS workshop on Deep Reinforcement Learning 2018
[105] Stable-Baselines3 Reliable Reinforcement Learning Implementations https://stable-baselines3.readthedocs.io/en/master/
[106] T Lesort, M Seurin, X Li, N Díaz-Rodríguez, D Filliat 2019 International Joint Conference on Neural Networks (IJCNN) 2017
[107] Error Analysis tool, part of the Responsible AI Dashboard in Azure: https://learn.microsoft.com/en-us/azure/machine-learning/concept-responsible-ai-dashboard
[108] Evolved from "Towards Accountable AI: Hybrid Human-Machine Analyses for Characterizing System Failure besmira Nushi Ece Kamar Eric Horvitz HCOM 2018.
[109] Deep Reinforcement Learning that Matters - P Henderson · 2017
[110] L. Hurwicz & S. Reiter (2006) Designing Economic Mechanisms,
[111] Facial Recognition: Analyzing Gender and Intersectionality in Machine Learning. Report. http://genderedinnovations.stanford.edu/case-studies/facial.html#tabs-2
[112] Piggyback: Adapting a Single Network to Multiple Tasks by Learning to Mask Weights. 2018
[113] AS Ross, MC Hughes, F Doshi-Velez. Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations. IJCAI'17.
[114] K Burns, LA Hendricks, K Saenko, T Darrell, A Rohrbach. Women also Snowboard: Overcoming Bias in Captioning Models. ECCV'18, 771-787.
[115] A Rohrbach, L.A. Hendricks, K Burns, T Darrell, K Saenko. Object Hallucination in Image Captioning. EMNLP'18.
[116] References Glossary
[117] For the glossary, the following references have been used, which can be consulted to expand on the aspects described therein.
[118] https://medium.com/analytics-vidhya/abstract-interpretation-for-ai-cc52bc0ddec2
[119] https://arxiv.org/abs/2209.08754
[120] https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc
[121] https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

[122] http://dmip.webs.upv.es/papers/BFHRHandbook2010.pdf
[123] https://arxiv.org/pdf/1706.08840.pdf
[124] https://support.zivid.com/en/latest/reference-articles/blooming-bright-spots-in-the-point-cloud.html
[125] https://www.geeksforgeeks.org/branch-and-bound-algorithm/
[126] http://dmip.webs.upv.es/papers/BFHRHandbook2010.pdf
[127] http://dmip.webs.upv.es/papers/BFHRHandbook2010.pdf
[128] https://www.statisticshowto.com/cohens-kappa-statistic/#:~:text=What%20is%20Cohen%27s%20Kappa%20Statistic,to%20the%20same%20data%20item.
[129] https://towardsdatascience.com/predict-your-models-performance-without-waiting-for-the-control-group-3f5c9363a7da
[130] https://jmlr.csail.mit.edu/papers/volume2/crammer01a/crammer01a.pdf
[131] https://arxiv.org/abs/1811.00866
[132] https://en.wikipedia.org/wiki/Data_differencing
[133] https://gsarantitis.wordpress.com/2020/04/16/data-shift-in-machine-learning-what-is-it-and-how-to-detect-it/
[134] https://gsarantitis.wordpress.com/2020/04/16/data-shift-in-machine-learning-what-is-it-and-how-to-detect-it/
[135] https://neptune.ai/blog/knowledge-distillation
[136] https://towardsdatascience.com/you-cant-predict-the-errors-of-your-model-or-can-you-1a2e4a1f38a0
[137] https://arxiv.org/abs/2105.04093
[138] https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/
[139] https://paperswithcode.com/method/experience-replay
[140] https://towardsdatascience.com/the-f1-score-bec2bbc38aa6
[141] http://theory.stanford.edu/~barrett/pubs/LAL+21.pdf
[142] http://theory.stanford.edu/~barrett/pubs/LAL+21.pdf
[143] https://arxiv.org/abs/1809.08098
[144] https://medium.com/geekculture/cross-validation-techniques-33d389897878
[145] https://machinelearningmastery.com/k-fold-cross-validation/
[146] https://medium.com/dataseries/how-modern-game-theory-is-influencing-multi-agent-reinforcement-learning-systems-2a64a3ba0c2c
[147] https://medium.com/geekculture/cross-validation-techniques-33d389897878
[148] https://medium.com/geekculture/cross-validation-techniques-33d389897878
[149] https://www.analyticsvidhya.com/blog/2020/11/binary-cross-entropy-aka-log-loss-the-cost-function-used-in-logistic-regression/#:~:text=Log%20loss%2C%20also%20known%20as,predicted%20probabilities%20and%20actual%20values.
[150] https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjdgb2Rv5__AhUvUqQEHTxkCnIQFnoECBwQAQ&url=https%3A%2F%2Ftowardsdatascience.com%2Fmicro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f&usg=AOvVaw2K03HLxoh-9m77323sHXGE
[151] https://towardsdatascience.com/the-best-classification-metric-youve-never-heard-of-the-matthews-correlation-coefficient-3bf50a2f3e9a

[152] https://www.mydatamodels.com/blog/regression-metrics/
[153] https://moodle.unob.cz/pluginfile.php/42899/mod_resource/content/1/Semiquantitative%20risk%20assessment%2C%20the%20risk%20position%20of%20an%20entity.pdf
[154] https://towardsdatascience.com/a-hitchhikers-guide-to-mixture-density-networks-76b435826cca
[155] https://towardsdatascience.com/cross-validation-k-fold-vs-monte-carlo-e54df2fc179b
[156] https://www.britannica.com/science/mean-squared-error
[157] https://onnx.ai/
[158] https://ieeexplore.ieee.org/document/8514437
[159] https://medium.com/analytics-vidhya/out-of-distribution-detection-in-deep-neural-networks-450da9ed7044
[160] https://www.nature.com/articles/s43586-022-00125-7
[161] https://towardsdatascience.com/mixed-integer-linear-programming-1-bc0ef201ee87
[162] https://towardsdatascience.com/forecast-kpi-rmse-mae-mape-bias-cdc5703d242d
[163] ethics-by-design-and-ethics-of-use-approaches-for-artificial-intelligence_he_en.pdf (europa.eu)
[164] https://homepage.cs.uiowa.edu/~tinelli/papers/BarTin-HBMC-18.pdf
[165] https://www.solver.com/optimization-tutorial
[166] https://towardsdatascience.com/deep-domain-adaptation-in-computer-vision-8da398d3167f
[167] https://christophm.github.io/interpretable-ml-book/counterfactual.html
[168] http://proceedings.mlr.press/v70/oh17a/oh17a.pdf

## 8.2 Standards and Norms

The present section compiles the main recommendations from a set of international standards related to the robustness of artificial intelligence systems, within the framework of **Article 15 of Regulation (EU) 2024/1689**, which establishes the obligation to ensure that AI systems remain robust, safe, and accurate throughout their entire life cycle.

These standards provide **technical guidelines** for the assessment of robustness, reliability, and quality of AI systems, helping providers and developers to demonstrate compliance with robustness requirements through appropriate **design, testing, and validation practices.**

The set of standards referenced herein has been developed primarily within the framework of the **ISO/IEC JTC 1/SC 42 Technical Committee**, specialised in Artificial Intelligence, as well as by other international standardisation bodies (**IEEE, ETSI, ITU, DIN, and CEN/CENELEC**). Some of these standards have already been published, while others are currently under development or review.

The **normative standards** that encompass the contents of this document are as follow:

[1] ISO/IEC TR 24029-1:2021, Artificial Intelligence (AI) – Assessment of the robustness of neural networks – Part 1: Overview.

[2] ISO/IEC 24029-2:2023, Artificial intelligence (AI) – Assessment of the robustness of neural networks – Part 2: Methodology for the use of formal methods.
[3] ISO/IEC/IEEE 29119-3:2021, Software and systems engineering – Software testing – Part 3: Test documentation.
[4] ISO/IEC 25000:2014, Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE.
[5] ISO/IEC 25030:2019, Systems and software engineering – Systems and software quality requirements and evaluation (SQuaRE) – Quality requirements framework.
[6] ISO/IEC 25059:2023, Software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Quality model for AI systems.
[7] prEN 18229-2 (en desarrollo), AI Trustworthiness Framework – Part 2: Accuracy and Robustness.

Additionally, the following related standards and technical documents can also be consulted, as they address complementary aspects of **robustness, reliability, and evaluation metrics in AI**:

[8] DIN SPEC 92001-2:2020-12, Artificial Intelligence – Life Cycle Processes and Quality Requirements – Part 2: Robustness.
[9] ETSI TR 103 821 (INT-008 / AFI), Autonomic network engineering for the self-managing Future Internet (AFI); Artificial Intelligence (AI) in Test Systems and Testing AI Models (Technical Report).
[10] ITU-T F.748.12 (2021), Deep learning software framework evaluation methodology.
[11] ITU-T F.748.11 (2020), Metrics and evaluation methods for a deep neural network processor benchmark.

Financiado por
la Unión Europea
NextGenerationEU

GOBIERNO DE ESPAÑA
MINISTERIO PARA LA TRANSFORMACIÓN DIGITAL Y DE LA FUNCIÓN PÚBLICA
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL

Plan de
Recuperación,
Transformación
y Resiliencia

España | digital 2026