EC3.21

Methodological Guideline for Multi-Scale Online Monitoring Framework

L3.5.2.3

contact@confiance-ai.fr | www.confiance.ai

Document reference: 321JA

Contributors

	Name	Organisation	Role
Responsible for the deliverable	Paul-Marie RAFFI	IRT SystemX	Research Engineer
Scientific responsible	Fateh Kaakai	Thales	Research Director
Co-authors	Fateh Kaakai	Thales	Research Director
	Paul-Marie RAFFI	IRT SystemX	Research Engineer

Document Control

Revision	Date	Commentary	Author
v1.0	10/01/2024	Initialization from Batch 2	Paul-Marie RAFFI
v1.1	12/01/2024	Delivery	Paul-Marie RAFFI

Contents

A Introduction and abstract **3**
A.1 General introduction to trustworthy AI challenges . . . 3
A.2 Introduction to Online Monitoring Framework . . . 3
A.2.1 Scientific Challenges . . . 3
A.2.2 Trustworthiness Attributes . . . 3
A.2.3 Target audience and disclaimer . . . 3
A.2.4 Taxonomy . . . 4
A.2.4.1 Operational Design Domain (ODD) Definition . . . 4
A.2.4.1.1 ODD definition in EC2 Taxonomy . . . 4
A.2.4.1.2 Concept of AI Model ODD . . . 4
A.2.4.1.3 Principles of the AI Model ODD . . . 4
A.2.4.2 Model Robustness and Stability Definitions . . . 5
A.2.4.3 Monitoring Time Scales . . . 6
A.2.4.4 AI system interactions . . . 6
A.2.5 Overall vision, context, motivation and objectives . . . 6
A.2.5.1 General context . . . 6
A.2.5.2 Motivation and scope . . . 7
A.2.5.3 Objectives . . . 8

B Description of the method **9**
B.1 What is Multi-timescale Online Monitoring? . . . 9
B.2 Some Methodological Considerations to Design a Monitor . . . 12
B.3 Key Design Principles . . . 14
B.4 Monitoring Methodologies . . . 14

C Conclusion **16**

Bibliography **17**

A. Introduction and abstract

A.1. General introduction to trustworthy AI challenges

Trustworthiness in AI within critical systems (systems that can directly or indirectly affect human life and moral entities) is essential for its widespread adoption (by the industry, the decision makers, the general public, etc.) and poses the following significant challenges.

- First, how to design AI models, so that, by construction, they satisfy trustworthy properties (accuracy, robustness. . .).
- Secondly, how to characterize these AI models, for example to understand and explain their behavior and their adequacy to the operational domain.
- Then, how to implement and embed those AI models on hardware, by making them fit for the target without losing their trustworthy properties.
- Another question is, what methods of data engineering to apply in order to, among other topics, manage important volumes of data and adapt to the evolution of the operational domain.
- At system level, what verification and certification processes to consider specifically for AI-based systems.
- Finally, a federation of all these matters is necessary to build an end-to-end methodological approach, supported by a consistent engineering environment compatible with industrial practices.

These are the challenges, among others, that the Confiance.ai program addresses.

A.2. Introduction to Online Monitoring Framework

A.2.1 Scientific Challenges

This document addresses the scientific challenges:

- Components with integrated self-monitoring of the detection of the exit from the operating zone.
- Calculation of the confidence score by various approaches.

A.2.2 Trustworthiness Attributes

Maintainability, Operability, Safety, Robustness, Stability

A.2.3 Target audience and disclaimer

This document has been written in a focus to be easy to comprehend for any reader having a few notions in Mathematics, Digital Imagery and Time Series. It is intended to give rule-based methods for:

- Data Scientists and Data Managers who want to improve the quality of their training datasets by detecting and annotating anomalies.

- AI Algo Engineers and Safety Engineers who want to control the inputs and outputs of AI models in operation.

A.2.4 Taxonomy

This chapter presents useful definitions of key concepts that will be frequently used in the rest of this report. It is recommended to have a good understanding of these definitions before deep diving into the next chapters.

A.2.4.1 Operational Design Domain (ODD) Definition

According to the taxonomy document developed by EC2 [L1.1.1 - 1st version of Confiance.ai Taxonomy], the Operational Design Domain (ODD) is defined as follows:

A.2.4.1.1 ODD definition in EC2 Taxonomy

Operating conditions under which a given driving automation system or feature thereof is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristic SAE J3016 (2018).

This definition in EC2 taxonomy document is coming from the automotive domain (Road Motor Vehicle Automated Driving Systems) has been considered by project EC6 to develop a top-down approach to specify ODD from system considerations.

A.2.4.1.2 Concept of AI Model ODD

Now, if we assume that this "ODD as specified" exists and has driven the development of an AI model as depicted in in Figure A.1. Let also assume that the AI Model has been verified against its allocated requirements and is ready for deployment in the field. Therefore, at the end of the development process, we obtain what we call an "ODD of the AI Model as designed, implemented and verified)" or simply and shortly "AI Model ODD". To characterize this AI Model ODD, we need to measure the performance of the AI Model in operational conditions as close as possible to real one (i.e., nominal and abnormal conditions). By using inference results of the AI Model to input data containing nominal values and anomalies, we characterize the AI Model ODD through the measured ranges of parameters where the AI Model provides a correct output within an acceptable margin error. This AI Model ODD will be the baseline for the online monitoring capability since it corresponds really to the domain where the AI Model guarantee the expected behavior and the expected level of performance (since it has been verified against its allocated requirements during the development process). At this point, many questions arise like: *does the AI Model ODD perfectly fit with the specified ODD? Is the AI Model ODD a superset or a subset of the specified ODD?* All these questions are legitimate and should be managed by relevant process interface between the system engineering processes and AI Model development processes (EC2, EC4 and EC6 scopes). In this report, we develop more deeply this concept of AI Model ODD since it is a key concept for the online monitoring capability.

A.2.4.1.3 Principles of the AI Model ODD

There are various potential classes of anomalies depending on the data type (images, time series, natural language, ...) and on the type of problem the industrial is trying to address (classification, object detection, semantic segmentation, regression task, ...). For example, the Renault

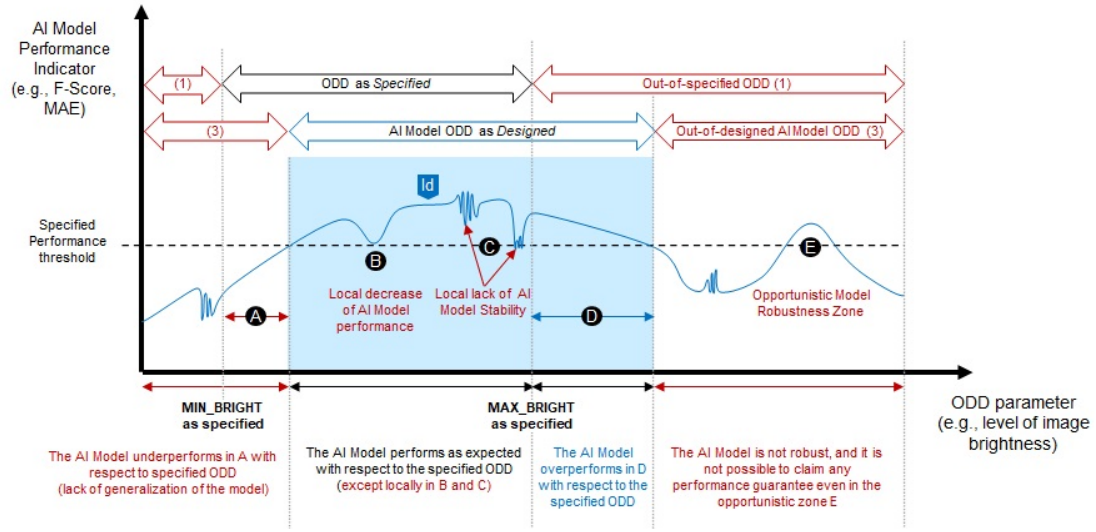

Figure A.1: Specified ODD at system level and AI Model ODD as designed

welding Use Case is a problem of image classification into 3 classes: "Compliant welding", "non-compliant welding", or "Unknown" (see the use case description in Part III-Appendixes). Based on the system ODD as specified with the use case owner, we identified potential anomalies of interest for this type of problem (images with too much color, blur, translation, rotation, brightness, darkness, ...) and defined ranges and increment steps (sampling) for these anomalies. Some anomalies such as Rotation can be selected in their whole range, from 0 degree to 360 degrees, and only the increment step is chosen. Other anomalies such as Blur or Brightness cannot be estimated easily until tested against the AI Model during a calibration phase (few cycles of tests before finding a range of interest). There are also hardware limitations that can reduce the number of steps. A trade-off has been found during the Batch 2 for each class of anomalies between storage, time of computation and performance of the Model ODD.

A.2.4.2 Model Robustness and Stability Definitions

According to the taxonomy document developed by EC2 [L1.1.1 - 1st version of Confiance.ai Taxonomy], the definition of model robustness and model stability are the following F. et al. (2021):

- The Global robustness of a model or a system is the ability of the model/system to perform the intended function in the presence of abnormal or unknown inputs.
- The local robustness is the extent to which the system provides equivalent responses for similar inputs.

The above definitions are very general and are reworded in the EUROCAE/SAE AS8963 (draft4) SAE (2022) in more accurate terms using the ODD concept and where the global robustness is called robustness and the local robustness is called stability.

A.2.4.2.0.1 Robustness of an AI-based model/system The capacity of an AI-based model/system to preserve its expected / intended performance under well-characterized abnormalities or deviations to its inputs and operating conditions outside its operational design domain (ODD).

A.2.4.2.0.2 Stability of an AI-based model/system The capacity of an AI-based model/system to preserve its expected / intended output(s) under well-characterized and bounded perturbations to its inputs and operating conditions within its operational design domain (ODD)
In the rest of this report, we will use the definitions of SAE (2022) which are consistent with EC2 taxonomy since they provide an important reference to the ODD and a clear distinction between robustness to invalid inputs outside the ODD and stability within the ODD.

A.2.4.3 Monitoring Time Scales

Monitoring methods on a system can happen at different temporalities. We propose to regroup them into 3 time scale families: Present Time, Near Past and Near Future.

A.2.4.3.0.1 Present Time The Present Time Monitoring (PTM) method applies on a system in operation and gives immediate metrics based on information coming in real time.

A.2.4.3.0.2 Near Past The Near Past Monitoring (NPM) method applies on a system in operation and gives analysis metrics based on recent information.

A.2.4.3.0.3 Near Future The Near Future Monitoring (NFM) method applies on a system in operation and gives forecast metrics based on recent information.

A.2.4.4 AI system interactions

We also propose to regroup monitoring functions into 3 types depending on the interactions they have with the AI system.

A.2.4.4.0.1 Black box The monitored AI system can't be modified and there is no access to the internal states. Only the input and the output of the system can be observed.

A.2.4.4.0.2 Grey box The monitored AI system can't be modified but there is access to the internal states. The AI system architecture is known. The input, output and internal states of the system can be observed. The input and output can be modified.

A.2.4.4.0.3 White box The parameters and code to train the AI system are known. The AI system architecture is known. The input, output and internal states of the system can be observed and modified. The AI system can be modified to enable monitoring capabilities by design.

A.2.5 Overall vision, context, motivation and objectives

A.2.5.1 General context

Artificial Intelligence (AI) technologies are increasingly used in the development of mission-critical and safety-critical products in many application fields such as automotive, aviation, defense, energy, naval, railway, space, etc. AI technologies are diverse, including Machine Learning (ML), Symbolic AI and Hybrid AI . The models designed with these AI technologies are not easily testable and certifiable due to several difficulties well known to specialists in the field. For example, in the case of ML models, the certification challenges are the availability and the quality of the data used for the learning, the stability of the model against small perturbations on

input data within the Operational Design Domain (ODD), the robustness of the model against invalid inputs outside the ODD, the ability of the model to provide a correct prediction for data not seen during the design (generalisation capability), the explainability of the model composed of a very large number of internal parameters, the resilience of the model against concept drift, etc. In the industry, it is commonly established that the main objective of online monitoring of AI models (also called Run Time Assurance in [21] or safety control structure in [25]) is to detect (i) any deviation of the AI component deployed in production from the expected behavior (i.e., intent specified at the system level and allocated to the AI model), and (ii) precursors of the occurrence of failure conditions (i.e., feared events at the system boundaries) based on a predefined set of safety properties. Deploying a monitoring component running in parallel with the AI model is a practical way to manage the risk induced by a model for which it is not possible or feasible to formally demonstrate the achievement of the performance and the safety objectives resulting from the system analyses. Online monitoring is an architectural pattern well-known to safety engineers, but it had to be adapted to AI technologies.

A.2.5.2 Motivation and scope

In an ideal world, the AI model can perform its prediction over its entire Operational Design Domain (ODD) with the expected level of performance (e.g., 99.9 % correct predictions, and this accuracy is maintained over time in operation). However, in practice, if we consider for example machine learning models in many recent papers, most of the time it is very difficult to achieve more than 99 % accuracy, which is an average of one wrong prediction out of 100 inferences in production. But should we hastily conclude that 1 % of bad prediction systematically triggers unexpected behavior leading to a system failure condition? In practice, from the extensive industrial feedback collected in Confiance.ai program and for a wide range of industrial applications, a single error does not directly lead to hazardous or catastrophic events, because the system design has eliminated Single Points Of Failure (SPOF) (e.g., application of the following guidelines ARP4754 Aerospace (2010) and ARP4761 Aerospace (1996) in the aviation, IEC 61508 IEC (2010) in industry, ISO 26261 ISO (2011) in the automotive, etc.). Therefore, based on this assumption (no SPOF in the system), it implies that a single failure of an AI component (i.e., an incorrect prediction at a given time) cannot directly lead to hazardous or catastrophic events. However, what about the case where the AI component has persistent failures (i.e., the model fails to infer the correct prediction at a given point in time, and continues to fail during a subsequent time interval)? This could increase the residual risk due to a higher probability of having (during that time interval where the AI component continues to fail) a combination of multiple internal failures in the system leading to a system failure condition. To detect persistent failures, considering the dynamics of the system and thus the "time" variable is a major issue. The technical problem to solve is related to the detection performance of the on-line monitor (detection rate of anomalies during operations) for AI models whose detailed design is not easily traceable to an upstream functional or non-functional specification. Indeed, technical products developed using the traditional method are generally subject to a functional and non-functional requirements specification phase (safety, security, performance, human factors, etc.), a product architecture definition phase, a design phase, an implementation phase, an integration-verification-validation phase, and then a production deployment phase. These developments are based on end-to-end knowledge-driven engineering. In contrast, products containing AI may contain software or hardware components whose design is the result of data-driven engineering and learning algorithms, rather than knowledge-driven engineering (the availability of a large volume of data associated with powerful learning algorithms replaces human knowledge in this

case). For example, a neural network model can be represented mathematically by operations of matrices, vectors and activation functions with a very large number of internal parameters (potentially several tens or hundreds of thousands, or even millions), called "weights". The numerical values of these weights are obtained after learning, and do not derive directly from "business" rules based on knowledge of the domain and the application under consideration. Traditionally, online monitors have been specified, designed and qualified on the basis of these business rules. With AI models, the problem arises of correctly defining the properties to be monitored with a partial or superficial understanding of the detailed design of the model (as the specific values of the weights obtained through machine learning cannot easily be justified). And even if one assumes that engineers can define some of the properties to be monitored, there is the technical challenge of demonstrating the completeness of these properties to ensure the appropriate level of safety.

A.2.5.3 Objectives

In this report, an innovative multi-time scale online monitoring architecture is presented. The main idea is to combine several monitoring timescales - Present-Time Monitoring (PTM), Near-Past Monitoring (NPM), and Near-Future Monitoring (NFM) - and various monitoring technologies on different monitoring assets (inputs, internal states, and outputs of the AI model) to ensure a high anomaly detection rate by design of the online monitor while minimizing the rate of false alarms.

B. Description of the method

B.1. What is Multi-timescale Online Monitoring?

The context of the online monitoring device is described in the figure B.1. The product incorporating one or more AI models is shown in black. This product can be a component of a system or a system as a whole. The generic term "product" is used in the following. Above the product, in a white oval, the online monitoring device receives both the external inputs and outputs of the product, as well as some information about the internal state of the product using pre-designed probes placed in the product's software code or hardware. A device for collecting all relevant operational data is usually required to calculate off-line metrics and to fine-tune some of the on-line monitor parameters. Finally, a controller is responsible for synthesising the output produced by the product and the verdict of the monitor to calculate the final output, which is so-called the "safe output". Consider now the deliberately simplified example in Figure B.1, which represents a product incorporating an AI model that approximates a linear physical phenomenon in the form of an linear function $y = a * x + b$ (solid green segment in Figure B.1).

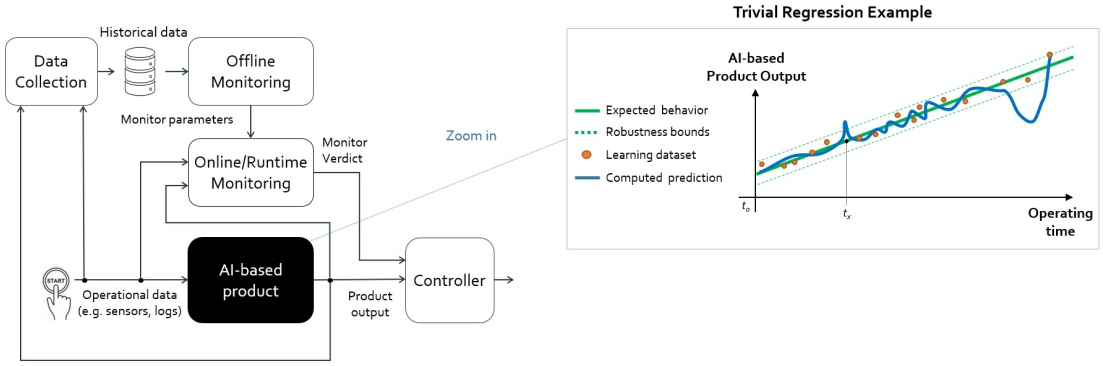

Figure B.1: Context of the AI Model and its online monitoring function

The principle of multi-scale monitoring is described in Figure B.2. It consists in combining several monitoring time scales: monitoring of the product or system behaviour at the present time, monitoring over a configurable time window of the near past, monitoring over a configurable time window of the near future.
In system engineering analyses, robustness bounds were defined (green dashed segments) that define the validity domain of the product output y. As sufficient data was available to characterise the physical phenomenon, it was decided to use machine learning technology to design the product (e.g. using deep neural network technology). The model obtained after learning is described by the blue curve in Figure B.1 where one can see several operating points of the output y that fall outside the validity domain. The aim of the monitoring device is to detect all operating points that violate the monitoring properties previously defined during the design of the monitoring device. The operation of the product in production is clocked by a system clock tk, which also clocks the monitoring device (at each time tk the device acquires data to produce a verdict). To illustrate the combination of these 3 different time-scale monitoring methods, let us continue the discussion of the trivial example of Figure B.3, the learning datasets (orange points)

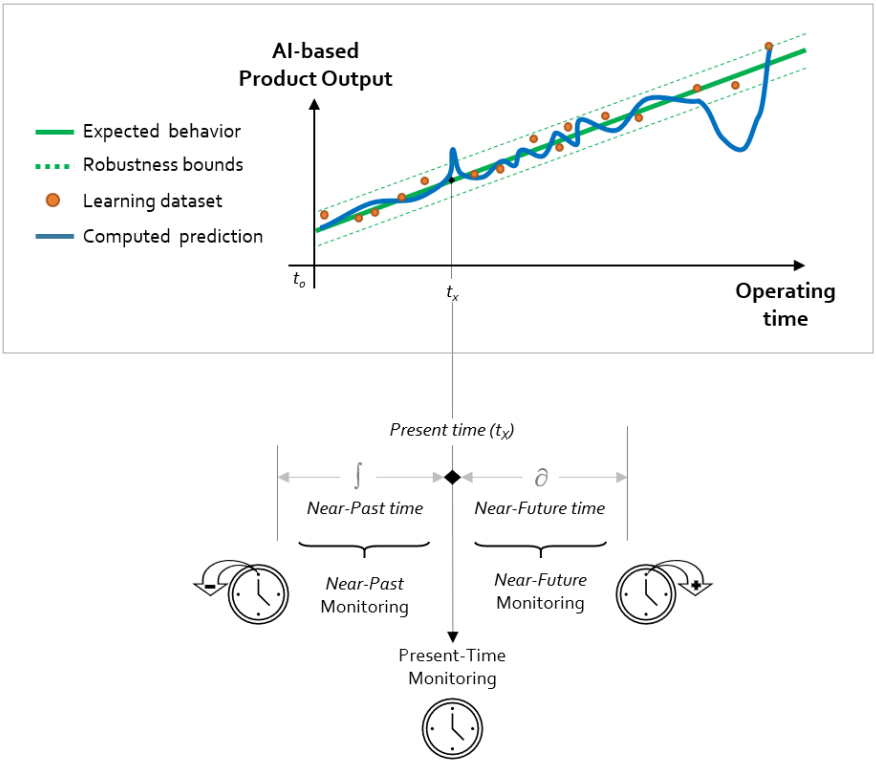

Figure B.2: Principle of the multi-timescale online monitoring

have been used to train and verify the regression model (e.g. feed-forward neural network). After several training iterations, the final model (M) is obtained as represented by the blue solid line. It is possible to see in this figure that M has several issues.
At t_x, M output overpasses the robustness boundaries, and it is expected that the Present Time Monitoring (PTM) will be able to detect such abnormal behavior. Between t_x and t_y, it is possible to observe that M output starts to unexpectedly oscillate. It is clearly an unintended behavior that could be a precursor of a failure of the model. Since this oscillation phenomenon should be observed and confirmed on several clock cycle, it is expected that the Near Past Monitoring (NPM) will be the appropriate monitoring timescale to detect such anomalies. At t_z, the output of M has a abrupt trend that will make it overpassing the robustness boundaries at the next clock cycle. Here, the Near-Future Monitoring (NFM) is the most appropriate method to detect such abnormal behavior since its is based on trend analysis. Through this didactic example, one can observe that an efficient combination of these 3 different monitoring timescale (NPM, PTM, NFM) allows to detect several classes of anomalies and to achieve by design a high rate of detection of failures that could occur in operational conditions when the AI model is in production and the minimisation of false alarms. The next chapters will present how this multi-timescale monitoring is implemented in practice through different industrial use cases.

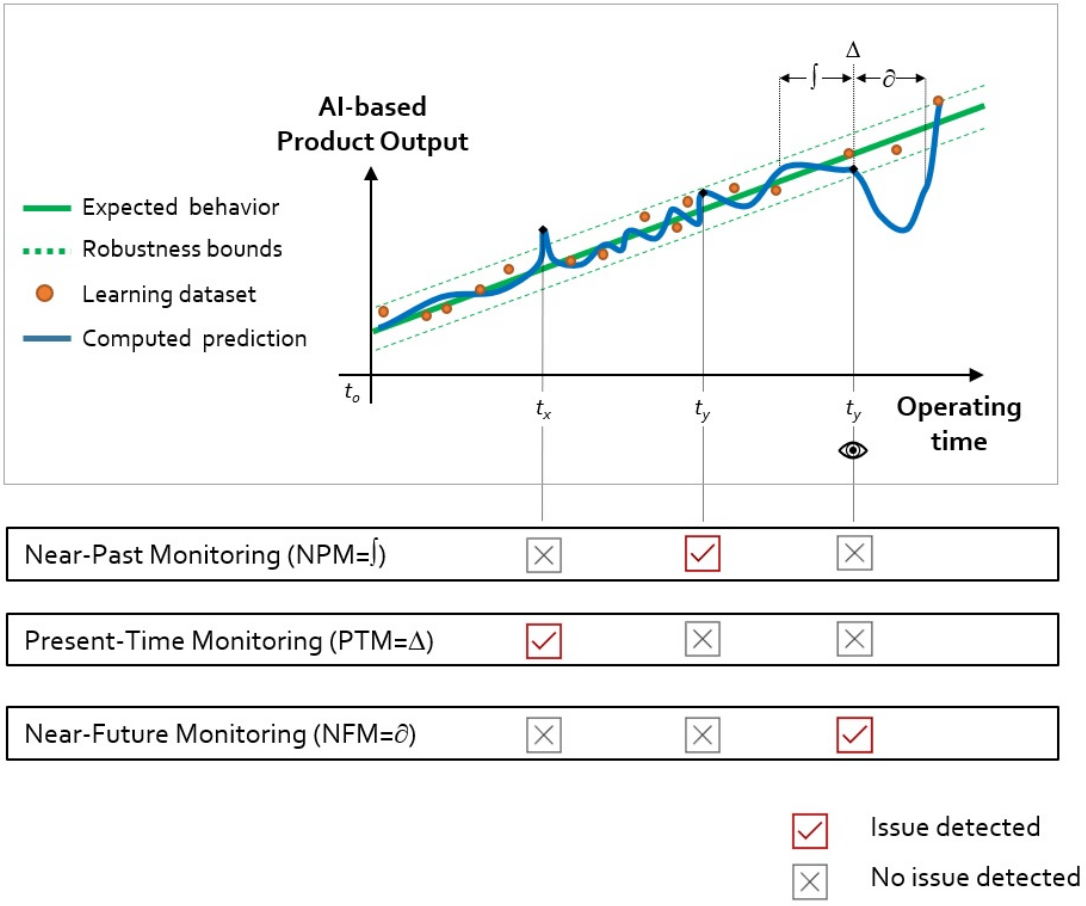

Figure B.3: Trivial example of multi-timescale online monitoring

B.2. Some Methodological Considerations to Design a Monitor

The development of monitors should follow two complementary design methods, one which is top-down and the other bottom-up as depicted in Figure B.4. The top-down approach is based on a traditional system engineering where the intended behavior of the AI-based product is specified in system requirements from CONOPS, Human Factor assessments, and System Safety Assessments. These system requirements are analyzed to characterize the ODD and the high level properties of the AI-based product. From this knowledge of the ODD and high level properties, it is possible to identify the classes of anomalies to be detected by the online monitor. In addition to this top-down approach, a complementary bottom-up design method is necessary to capture the classes of anomalies that are specific to the selected AI technology used to implement the AI model. For example, some AI technologies may be more sensitive to noise or geometric transformation compared to others and it is important to capture these intrinsic technology weaknesses in order to monitor the corresponding anomalies (e.g., monitoring the level of noise, or monitoring the level of translation/rotation of an image).

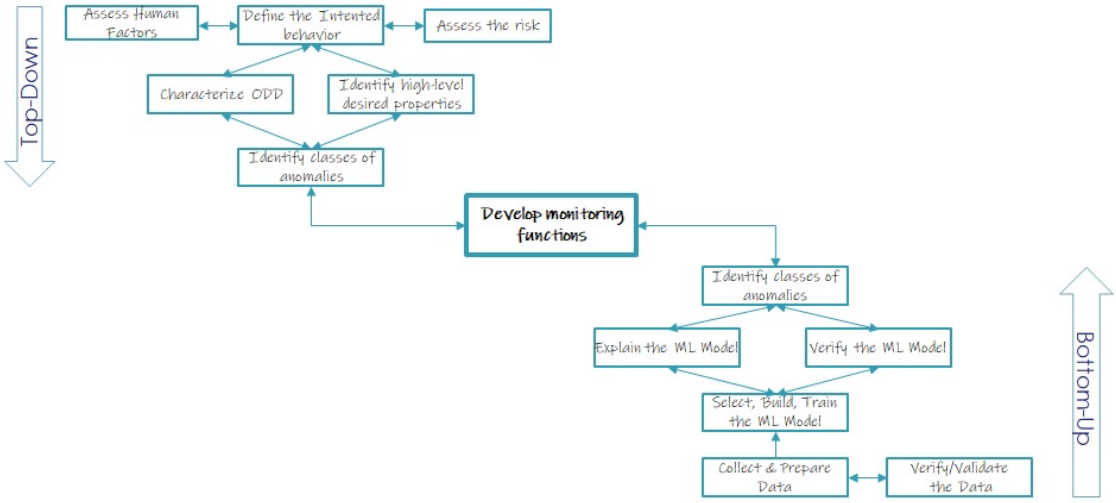

Figure B.4: Top-Down and Bottom-Up Design Methods

A necessary condition to reach a higher anomaly detection completeness of the monitor is to carefully select the monitoring assets: inputs of the AI model, output(s) of the AI model, internal states and variables of the AI model and also the assumptions considered during the design of the AI model, as illustrated in Figure B.5. For example, a current assumption that is taken for machine learning models is the *identically and independently distributed* (i.i.d.) property of the input data. Some verification proofs, like the underlying statistical theory to compute generalization guarantee, will no more hold if this assumption is not verified on the input data received from the sensors at inference time. Therefore, it is important to monitor online that this kind of assumption is still valid or not, and trigger an appropriate recovery mechanism at system level in case of invalid assumption.

The development of PTM, NPM and NFM monitoring functions can be done using different AI technologies: rule-based (or knowledge-based) technology, data-driven technology (Machine Learning (ML)), or hybrid technologies combining rule-based and data-driven. Data-driven technologies will generally give better performance and precision but can lack explainability and tracability in the raised alarms. On the contrary rule-based technologies can have weaker

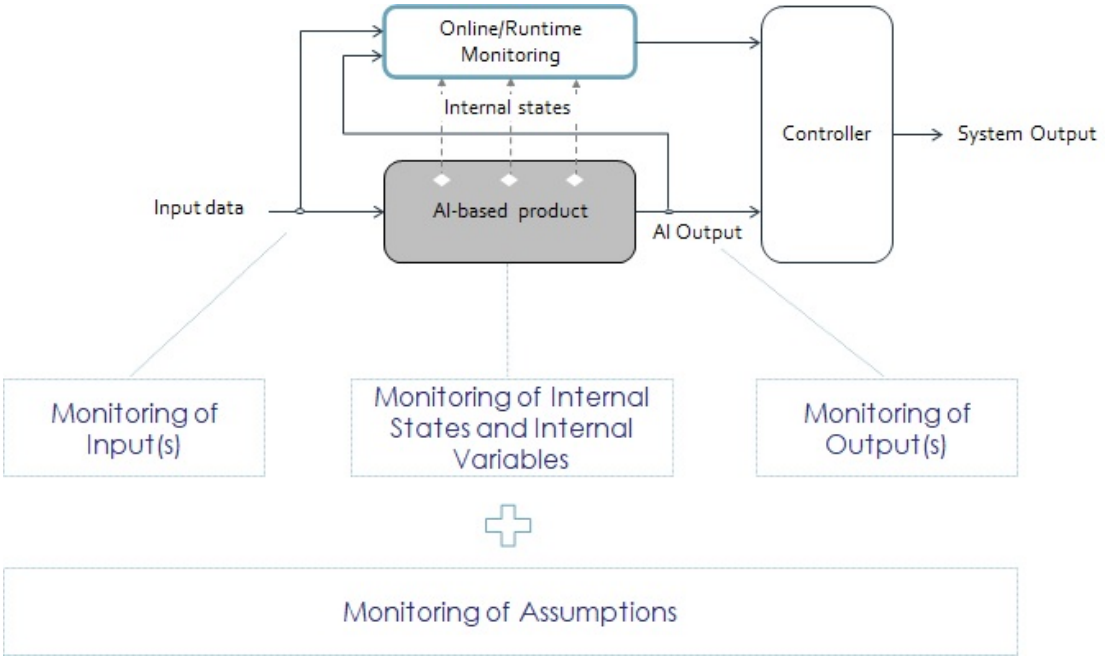

Figure B.5: Multiple Monitoring Assets

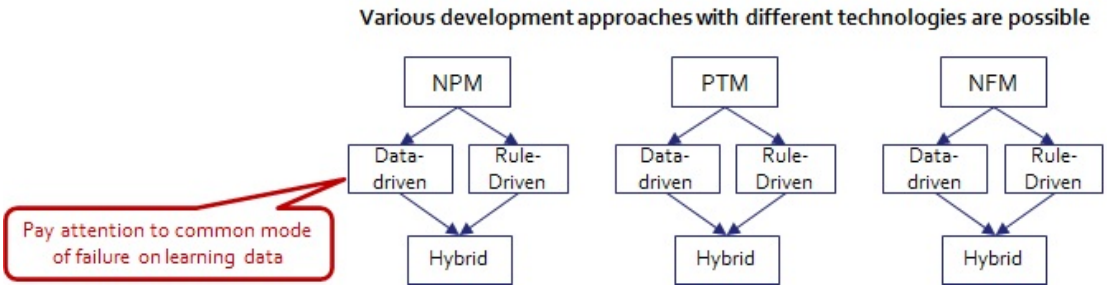

Figure B.6: Various development technologies are possible

performance and precision but will give reports understandable by humans with a tracability in the process that raised the alarms. Rule-based technologies might be required by some regulators to obtain quality or safety certifications. A special attention should be paid in the case where the AI model to be monitored and the monitoring function are both developed using ML and if they share the same learning datasets. In that specific case, it would be difficult to take credit from the ML-based monitoring function since it will have the same weaknesses as the ML model to be monitored (data quality attributes, robustness and stability issues, lack of generalization guarantee, etc.), and if they both use the same learning datasets, there is a high risk to have common modes of failure coming from the shared data. In general, the monitored function should be independent of the AI model in terms of specification, design and implementation as summarized in the following key design principles.

B.3. Key Design Principles

To conclude this part on the general presentation of Online Monitoring Framework, we provide a capitalization of five (5) principles to design monitors for AI systems:

- **Principle 1** The monitoring function should by-design ensure completeness of anomaly detection while minimising false alarms.
- **Principle 2** The sophistication of the monitoring function should be proportionate to the criticality level of the AI function.
- **Principle 3** The monitoring function should be smart to manage complexity and performance issues Principe 4.
- **Principle 4** The monitoring function should not have any safety adversarial common mode of failure with the monitored AI function.
- **Principle 5** The monitoring function itself should not have unacceptable impact (innocuity).

B.4. Monitoring Methodologies

Some monitoring methodologies of various technologies have been explored during this program. They are categorized in B.1 to help the reader choose among them which ones are the best suited to its use case requirements. The next step would be to implement the monitoring functions from the chosen methodologies (or to reuse the code of the components delivered in Confiance.ai implementing those methodologies), evaluate them on the reader's use case and assemble them into a multi-scale online monitor. The following step would be to aggregate the outputs of the monitoring functions by implementing a controller. Offline monitoring to analyze alarms trending can be implemented before or during deployment of the Online monitor. No methodologies have been explored in this program regarding the controller and the Offline monitor. In this table, the columns *Data type* and *Problem Type* contain only the types that have been tested on a Confiance.ai use case. It doesn't mean that it is impossible to handle other types of data or problem with the given method. Some adaptation and exploration might be needed to achieve it.

Reference - Methodological Guidelines	Monitoring Function	Data type	Problem type	Monitoring Asset	Time scale	AI System interactions	Monitoring Explicability
321DC - OODEEL	OOD Detection	Images	Classification	Output, Internal State	PTM	Grey-Box	No
321BA - Rule-based Monitoring	OOD Detection, Robustness, Stability	Images, Time Series	Classification, Segmentation, Regression	Input, Output	PTM, NFM	Black-Box, Grey-Box	Yes
321EA - Data stream	Data Drift Detection, Robustness	Images, Time Series	Detection, Regression	Output	PTM, NPM	Black-Box	No
321FA - Anomaly Detection	OOD Detection	Images	Classification	Output	PTM	Grey-Box, White-Box	No
321GA - OOD Detection	OOD Detection	Images	Classification, Detection, Segmentation	Input, Output	PTM	Grey-Box, White-Box	No
321HA - Uncertainty Quantification	OOD Detection	Time Series	Classification, Regression	Input, Output	PTM	White-Box	Yes
321IA - Data Drift Detection	Data Drift Detection	Images	Classification	Input, Output	PTM, NPM	Back-Box	No

Table B.1: Monitoring Methodologies

C. Conclusion

This report presents the innovative multi-time scale online monitoring framework. This framework is based on the acknowledgment that the dynamics of the system and thus the "time" variable is a major factor in the detection of anomalies. The combination of several monitoring timescales - Present-Time Monitoring (PTM), Near-Past Monitoring (NPM), and Near-Future Monitoring (NFM) - on different monitoring assets (inputs, internal states, and outputs of the AI model) is a solution to ensure "by design" a high anomaly detection rate of the online monitor. The development of a monitor following this framework should be structured using the principle of proportionality to risk with a sophistication level of the online monitor that is commensurate to the criticality of the AI-based product. As depicted in C.1, for low critical application the AI Model will be monitored as a black box (only external inputs and outputs of the AI Model are considered by the monitor). For medium critical application, a grey-box approach consists in adding to the former black-box approach the monitoring of internal states of the model obtained through an appropriate instrumentation of the AI Model (use of probes). For high critical application, which need the the highest level of sophistication, in addition to grey-box approach, a full access to the detailed design of the AI Model is granted with the possibility to observe the evolution of several variables/internal states of the AI Model at the same time. The AI Model can also be retrained or modified to enable monitoring capabilities by design.

Automotive (ISO 26262)	ASIL
Aviation: airborne (ED-12/DO-178/DO-254)	DAL
Aviation: ground (ED-109/DO-278)	AL/SWAL
Industries (IEC 61508)	SIL
Railway (CENELEC 50126/128/129)	SIL
Space (ECSS-Q-ST-80)	CAT

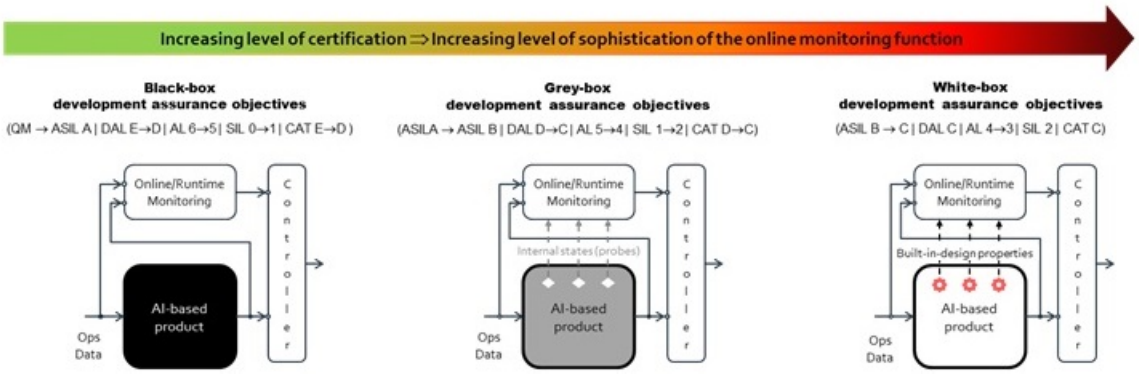

Figure C.1: Black-Grey-White box Monitoring Strategy

Bibliography

Aerospace, S. (1996). Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment. Technical report.

Aerospace, S. (2010). Guidelines for development of civil aircraft and systems. Technical report.

F., M., E., J., G., F., H., D., C., G., A., G., B., B., L., P., L., A., H., B., B., B., D., D., J.-B., G., A., H., S., P., K., D., C., P., J.-M., G., C., C., S., P., M., D., C., C., L., G., D., G. F., B., L., S., G., and A., A. (2021). White paper machine learning in certified systems. Technical report.

IEC (2010). Functional safety of electrical/electronic/programmable electronic safety-related systems (e/e/pe, or e/e/pes). Technical report.

ISO (2011). Véhicules routiers - sécurité fonctionnelle. Technical report.

SAE, E. . (December 2022). Process standard for development and certification/approval of aeronautical safety-related products implementing ai (draft 4b). Technical report.

SAE J3016 (2018). Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems.

Confiance ai

Title: Methodological Guideline for Multi-Scale Online Monitoring Framework

Keywords: monitor

Guidelines to combine several monitoring timescales and various monitoring technologies on different monitoring assets to improve anomaly detection rate and decrease the rate of false alarms

Our partners:

Air Liquide, AIRBUS, Atos, cea, Inria, NAVAL GROUP, GROUPE RENAULT, SAINT EXUPÉRY, SAFRAN, sopra steria, SystemX, THALES Building a future we can all trust, Valeo, Investir l'avenir