

Project Acronym: STAR
Grant Agreement number: 956573 (H2020-ICT-2020-1 – Research and Innovation Action)
Project Full Title: Safe and Trusted Human Centric Artificial Intelligence in Future Manufacturing Lines
Project Coordinator: INTRASOFT International



Funded by the Horizon 2020 Framework Programme of the European Union

DELIVERABLE

D4.1 – Library of XAI algorithms-Initial version

Dissemination level	PU - Public
Type of Document	Report
Contractual date of delivery	28/02/2022
Deliverable Leader	UPRC
Status - version, date	Final – v1.0, 08/03/2022
WP / Task responsible	WP4
Keywords:	Library of Explainable AI Algorithms

This document is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956573. It is the property of the STAR consortium and shall not be distributed or reproduced without the formal approval of the STAR Management Committee. The content of this report reflects only the authors' view. The European Commission is not responsible for any use that may be made of the information it contains.

Executive Summary

In recent years there is a surge of interest in the interpretability and explainability of AI systems, which is largely motivated by the need for ensuring the transparency and accountability of AI operations, as well as by the need to minimize the cost and consequences of poor decisions. In this context, explainable AI (XAI) research aims at providing a set of techniques that produce more explainable models, while maintaining a high level of searching, learning, planning and reasoning performance. XAI is set to enable and facilitate human users to understand, trust, and effectively manage the emerging generation of AI systems. During the last couple of years, many research works have introduced different measures and frameworks for XAI. Most of these frameworks focus on defining model explainability, formulating explainability tasks for understanding model behavior, developing solutions for these tasks, and specifying measures and techniques for evaluating the performance of models in explainability tasks. To that end the proposed methodologies envelop a wide variety of different AI systems (e.g., Machine Learning, Robotics, Deep Learning) which could also be applied in multiple sectors such as finance, healthcare, industry, etc.

STAR validates several explainable AI techniques in manufacturing environments and applications. In particular, the project considers the techniques that are most tailored for manufacturing use cases (including the project's pilots), notably XAI techniques for deep neural networks and human-robot collaboration. Techniques that identify the dominant features used by the AI systems like DeepLIFT [Shrikumar16] and Prediction Difference Analysis are researched and exploited in order to explain and interpret the project's quality management and agile production use cases. Moreover, techniques for explainable robotics will be further explored for the human-robot collaboration use cases of the project. In addition to implementing and validating these techniques, the project **links them to other components of the project** based on the overall architecture, notably the AI security techniques against data poisoning attacks and the reinforcement/active learning techniques that entail interaction/dialogue between robots and humans.

Deliverable Leader:	UPRC
Contributors:	UPRC, QLE, DFKI
Reviewers:	GFT, THA
Approved by:	Charalampos Ipektsidis, John Soldatos (INTRA)

Document History			
Version	Date	Contributor(s)	Description
0.1	08/02/2022	UPRC	Table of Contents.
0.2	15/02/2022	UPRC	Draft, State of the Art, Placement in the overall Architecture.
0.3	25/02/2022	UPRC	XAI Algorithms – 1 st Draft
0.4	03/03/2022	QLE	Application to PCL use-case
0.5	03/03/2022	ALL	2 nd Draft for review
0.6	08/03/2022	GFT, THA	Review – Ready for Submission
1.0	09/03/2022	INTRA	Final QA'ed version

Table of Contents

1. INTRODUCTION	7
1.1 DELIVERABLE SUMMARY	7
1.2 DOCUMENT SCOPE	8
1.3 DOCUMENT STRUCTURE	9
2. REQUIREMENTS AND STATE OF THE ART ANALYSIS.....	10
2.1 REQUIREMENTS	10
2.2 STATE OF THE ART ANALYSIS	12
3. PLACEMENT IN THE OVERALL ARCHITECTURE.....	18
3.1 OVERALL PLACEMENT	19
3.2 INTERNAL ARCHITECTURE	20
3.3 COMPONENT INTERACTIONS	23
3.4 IMPLEMENTATION AND DEPLOYMENT	26
4. XAI ALGORITHMS.....	28
4.1 XAI FOR TIMESERIES.....	29
4.2 XAI FOR TABULAR DATA	31
4.3 XAI FOR TEXT DATA (NLP)	31
4.4 XAI FOR IMAGES.....	32
4.4.1 Repurposing XAI for the PCL use case	33
5. CONCLUSIONS.....	34

Table of Figures

FIGURE 1: THE FOUR INDUSTRIAL EVOLUTIONS.....	13
FIGURE 2: TAXONOMY OF XAI METHODS.....	14
FIGURE 3: HIGH-LEVEL REFERENCE MODEL OF STAR.....	18
FIGURE 4: STAR ARCHITECTURE	19
FIGURE 5: PROVISION OF COUNTERFACTUALS INFORMATION BY THE STAR XAI.....	20
FIGURE 6: FEATURES RANKING PROVISION BY THE STAR XAI.....	21
FIGURE 7 : XAI INTERNAL WORKFLOW.	23
FIGURE 8: INFORMATION FLOW FOR A DEFENDING A POISONING ATTACK.....	24
FIGURE 9: INFORMATION FLOW FOR A DEFENDING AN EVASION ATTACK.	24
FIGURE 10: ACTIVE LEARNING ARCHITECTURE FROM [ROZANEC21]	25
FIGURE 11: N-BEATS.....	30
FIGURE 12: LIME	31
FIGURE 13: IMAGE EXPLANATIONS USING LIME.....	32

List of Tables

TABLE 1: XAI ALGORITHMS PER PILOT DATASET	28
---	----

Definitions, Acronyms and Abbreviations

Acronym/ Abbreviation	Title
AI	Artificial Intelligence
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
GRAD-CAM	Gradient-weighted Class Activation Mapping
IT	Information Technologies
LIME	Local Interpretable Model-Agnostic Explanations
ML	Machine Learning
NN	Neural Network
SHAP	SHapley Additive exPlanations
SVM	Support Vector Machine
WP	Work Package
XAI	eXplainable Artificial Intelligence

1. Introduction

The focus of the STAR project is to research, develop, validate, and demonstrate that Machine Learning (ML) and Artificial Intelligence (AI) technologies can be applied in industrial environments and manufacturing use cases. STAR's audience, in terms of its end users, are mostly domain experts from the fields of Industry and production lines. The necessity for computing systems to quickly analyse large amounts of data in demanding industrial environments has become greater than ever. Human beings have limitations on the amount of information they are able to process fast and efficiently in small time windows. Although a human decision over a specific set of information can be reliable, the main limitation is the timing of that decision. To that end, domain experts trust upon computers (machines) to make different types of real time analysis.

ML and AI are two fields that make use of statistical models to give computers the ability to learn from data and build predictive models. Although, ML and AI models have strong mathematical and statistical foundations, and since they use linear and non-linear transformations for predictive and classification tasks, they can be perceived as "black-boxes" for non-Information Technologies experts. Thus, generating an issue of trust in understanding the "inner-processes" of a ML/AI model. To that end, the need to explain in constructive and understandable methods the ML/AI predictions is of paramount importance.

The field of eXplainable Artificial Intelligence (XAI) aims to generate useful insights to the inner structures of ML algorithms and is a rapidly expanding field of research that also includes visualization techniques to provide clarity to various domain experts. The recent need for explainability is a consequence of the rise of computational strength that has led to the implementation of complex AI models. Although, there is not a unique definition to fully explain the field of XAI, the literature converges to including intrinsically interpretable and post-hoc explainability models.

1.1 Deliverable Summary

Nowadays, as AI systems become prevalent in many areas of the global economy, there is increased interest in the interpretability and explainability of AI systems. This trend is motivated by the need to keep under close control the costs and consequences of false AI decisions, especially human, as well as material and monetary and thus ensuring that AI operations are characterized by transparency and accountability. To this end, eXplainable AI (XAI) research focuses on providing a set of techniques that help develop more explainable models, while at the same time preserving their high-performing learning, planning, search, and reasoning functionalities. The ultimate end of XAI is to increase the human users' understanding and trust in AI systems, so that they can harmoniously coexist in the emerging era of ubiquitous AI.

As described in the state-of-the-art section, many recent research works have developed different frameworks for XAI. These works range from defining model explainability and formulating explainability tasks for understanding model behavior to developing solutions for these tasks as well as specifying measures and techniques for evaluating the performance of models from an explainability perspective. The spectrum of proposed methods is very wide as it includes methods for all the different flavors of AI systems (e.g., ML, Deep Learning,

Robotics, Multi-Agent Systems), different modalities (e.g., text, speech, image, time-series, tabular data) and for a wide array of application sectors such as healthcare, finance, human resources and industry.

The ambition of XAI in STAR and this deliverable in specific is to validate different XAI techniques in real-world manufacturing environments and applications. To that end, the project will investigate and choose XAI methods that best fit manufacturing use cases as defined by the project's pilots. Some examples described in this deliverable are generic, model-agnostic XAI techniques such as SHAP and LIME as well as class activation mapping techniques for deep neural networks and human-robot collaboration. Techniques that identify the most influential features for the decisions of the AI systems such as DeepLIFT [Shrikumar16] and Prediction Difference Analysis, will be researched and exploited in the context of the project's quality management and agile production use cases addressing the problem of interpretability inherent in Deep Neural Networks that operate on complex inputs such as images. Moreover, the envisioned scope could expand to other modalities such as text, timeseries and tabular data and applied to additional contexts such as explainable robotics used for the human-robot collaboration use cases of the project and common industrial use-cases such as demand forecasting. In addition to implementing and validating these techniques, we also envision highlighting synergies with other work packages of the project, most notably the AI defense mechanisms against data poisoning attacks and the reinforcement/active learning techniques that entail interaction/dialogue between robots and humans.

In terms of implementation, we envision the XAI component as a library of algorithms that supports different data modalities and AI algorithms and their technical implementations as they are developed in support of the project's use-cases. The library will be usable and accessible by a variety of different STAR components across multiple work packages and will hold a central role in the project's architecture. Increased focus will be given to taking advantage of model-agnostic methods - wherever possible - to boost reusability and adaptability. In terms of process, the most important aspect is the wide-ranging and thorough research of the state-of-the-art literature, as in this task priority should be given to choosing the right method characteristics for the AI models developed during the project and adjusting to any changes that occur within the project's lifecycle. Closely related and equally important are the implementation and rigorous evaluation of the techniques. Implementation details and the addition of interoperability features is the final step towards completion. Finally, the task also entails a visualization aspect, as it should not be forgotten that the end consumers of the XAI algorithm results will be the human users working in the production lines, and explanations should be presented in a user-friendly manner. Visualization capabilities are therefore considered through the full course of the component development.

1.2 Document Scope

The scope of D4.1 is to research and produce a library of XAI algorithms that will be applicable to the project's pilots and use cases. The focus lies on techniques that identify the features that are most important for the operation of a Deep Learning (DL) classification mechanism. Specifically, DeepLIFT and Prediction Difference Analysis techniques are considered (including their variations) for the predictive quality use cases of the project. The resulting XAI library will be used to boost the transparency of AI systems operations in the

use cases, while driving other functionalities of the STAR platform such as the cyber-defence techniques (WP3) and the active learning techniques of the project. The XAI-based mapping of DL algorithms to their explanations will be integrated/incorporated in the knowledge base to be developed in this WP (see T4.4).

1.3 Document Structure

This document describes the activities carried out in the first 14 months of the project and is structured in 5 sections.

Section 1 provides a summary of the deliverable together with the description and scope of T4.1.

In Section 2 we provide an overview of the recent scientific literature related to XAI continued from the desk research of D2.1 and more specifically tailored to how we plan to apply XAI to the STAR pilot scenarios. This section also includes the component requirements as shaped by the pilot use-cases.

Section 3 places the XAI component and highlights its role within the STAR architecture and presents interactions and synergies with other STAR components. It also contains some technical and deployment considerations.

In Section 4 we determine the XAI methods best corresponding to the different STAR use-cases based on the datasets provided in D2.4 and describe them in further detail as they will constitute the building blocks of the XAI library.

Finally, Section 5 includes the conclusions as well as steps towards further implementation and research.

2. Requirements and State of the Art Analysis

2.1 Requirements

In deliverable D2.1 the template of requirements for each individual component was presented. The template introduced the structure of the requirements emerging both from the pilots of STAR and from the technical components of the project. We have further refined the previously presented requirements as the project evolves to fit the STAR pilots and pilot datasets.

Section	Description
Id	REQ-XAI-1
Type	FUNC
Short name	XAI for Time-Series Data.
Description & quantification	XAI algorithms aiming to support time-dependent pilot datasets such as the sensor data from the Agile Manufacturing Pilot #2 and IMU data from Pilot #3 in order to explain predictions for subsequent time steps and future decisions based on the time-series.
Additional information	N/A
Priority	MAN

Section	Description
Id	REQ-XAI-2
Type	FUNC
Short name	XAI for Tabular Data.
Description & quantification	XAI methods suitable for datasets such as process and asset data or operator feedback from Pilot #1 and odometry sensor data from Pilot #3, which are organized in a tabular format. Extracted explanations to be visualized as bar-plots.
Additional information	N/A
Priority	MAN

Section	Description
Id	REQ-XAI-3
Type	FUNC
Short name	XAI for Text Data.
Description & quantification	Explanation provision for raw text data and NLP algorithms. Can be applied to interpret the potentially automated analysis of operator feedback (Pilot #1). Extracted explanations to be visualized as bar-plots.
Additional information	N/A
Priority	MAN

Section	Description
Id	REQ-XAI-4
Type	FUNC
Short name	XAI for Image Data.
Description & quantification	Aim is to provide interpretability for Convolutional Neural Networks and other ML algorithms which operate on images (color and grayscale) and potentially images from video sequences. Most notable application is the automated quality inspection of Pilot #1, but also potentially the grayscale video camera footprints and factory layout images from Pilot #3. The explanations are to be visualized as heatmaps overlaid over the inputs.
Additional information	N/A
Priority	MAN

Section	Description
Id	REQ-XAI-5

Type	FUNC
Short name	XAI for Cyber Security
Description & quantification	This group of algorithms will utilize XAI methods to identify possible poisoning and evasion attacks and data drifts to the training distribution. It will be interacting directly with the WP3 cyber security components.
Additional information	N/A
Priority	MAN

2.2 State of the Art Analysis

The current day and age, also known as the Digital or Information Age, is characterized by complex computing systems which generate enormous amounts of data on a daily basis. This trend towards digitization highlighted in Figure 1 has brought with it a continuous increase of computational resources and capabilities, which in conjunction with the reduction of their monetary cost, has led to a surge of computational models attempting to solve complex mathematical problems. To that end, the fields of ML and AI, which are based on strong statistical models, gained a lot of attention during the last decade serving as key enabling technologies. The need for large, dense and complex DL models has generated an issue of trust against the ML/AI predictions, especially when used by non-IT researchers. [Montavon18] The notion of explaining and expressing a ML model is called interpretability or explainability. [Choo18] This need for interpretability mainly exists in deep Neural Network models since in real world applications they need to operate as high-performance models which contain a huge amount (up to thousands) of hyper parameters which indicates extreme internal complexity by using non-linear transformations. The internal complexity of Neural Networks (NN) is mainly referred to as [Weller17] [Zahavy16] “black box”, “peering through a black box, “interpretable neural networks”. A lot of research addresses the problematic of ML “black-boxes” as not having clarity or insight also known as XAI [Gunning16].

Evidence that the field of XAI has become a rapidly increasing domain are prevalent in both industry and research. In 2017, DARPA funded the “Explainable AI program” aimed to increase the explainability of AI decisions [Turek17]; the same year “The Development Plan for New Generation of Artificial Intelligence” was published by the Chinese government highlighting the strong need for trustworthy and explainable AI models [ChineseCoun17]; in 2018 the “General Data Protection Regulation” (GDPR) published by the European Union, allows every individual a right of “explanation” in case personal data have been affected by algorithmic manipulations [Goodman17]. Industrial applications integrate a variety of AI solutions to enable systems, machines and devices to “learn” from their own data and enhance human capabilities [Ahmed22]. To that end, the field of XAI occupies a very sensitive, but nevertheless, key role in industrial applications since it serves as the bridge between complex DL models and non-Information Technology (non-IT) experts. To that end, the explanations provided by XAI methods need to be precise and understandable by

experts of various domains in order to increase the notion of “trust” in a real time industrial environment.

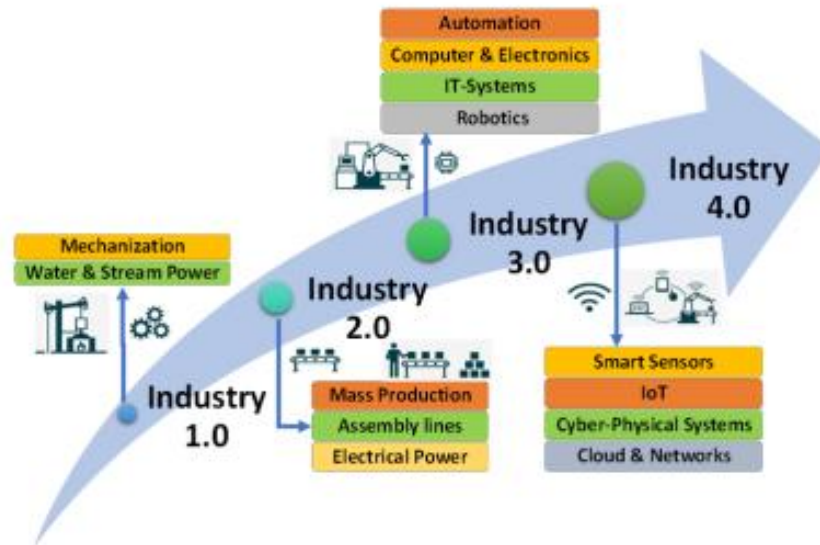


Figure 1: The four industrial evolutions.

Source: I. Ahmed, G. Jeon and F. Piccialli, "From Artificial Intelligence to eXplainable Artificial Intelligence in Industry 4.0: A survey on What, How, and Where," in IEEE Transactions on Industrial Informatics, doi: 10.1109/TII.2022.3146552.

Bhatt et al. [Bhatt20] performed extensive research to identify ways that industry practitioners make use of and deploy XAI models. Model debugging, monitoring, transparency and audit were the most common needs among industrial domain experts. Regarding model debugging, which mainly focuses on data scientists, experts require high levels of explainability in order to identify bottlenecks in poor model performance, especially when they exist in “static” feature spaces. To that end, experts need to be able to receive quickly and efficiently guidance on how to engineer new features, drop redundant ones and gather more data to improve model performance. Model monitoring, mostly, refers to evolving feature spaces in which [Zenisek19] concept drifts are noticed, and experts need to have an intrinsic understanding of the ML/AI models used to adapt to the new statistical distributions that might arise. Concerning model transparency, organizations deploy models to make decisions that directly affect user-sought explanations for model predictions. Finally, for model audit, industrial domain experts conduct various kinds of tests defined by a series of regulations. The findings of the survey indicated that there is a gap between explainability practices in real world applications and the goal for transparency since explanations primarily serve internal stakeholders.

In the literature of the XAI field, many times, there is confusion or misuse of the terms interpretability and explainability. One needs to define the differences between the two terms as to avoid confusion. The notion of interpretability is a reference to a passive characteristic of a ML/AI model which describes the extension that a model is understandable (makes sense) to a domain expert. On the contrary, the term explainability is used to describe an active characteristic of a given ML/AI model, signifying any type of action taken by a model to produce higher levels of clarity of its internal functions and structure. Although during the last couple of years several XAI methodologies, strategies

and frameworks have been presented, for the purposes of this research which focuses on industrial applications we will classify XAI methods according to their simplicity, the scope of interpretability and the percentage of dependencies from the analyzed/used AI models (Figure 2).

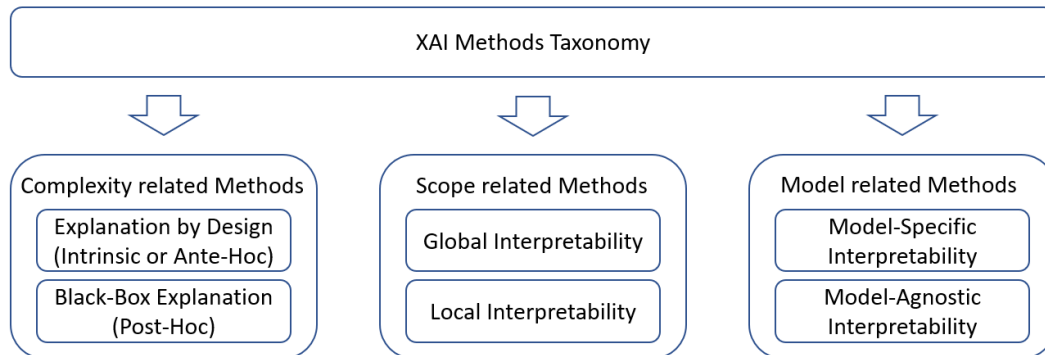


Figure 2: Taxonomy of XAI Methods

The notion of complexity is directly linked to the interpretability of a given ML/AI model. Interpretation and explainability of given ML/AI solutions, becomes a challenging task as these models increase in complexity and become more dense in order to deal with large feature spaces. To that end, Adadi et al. [Adadi18] highlighted that the most prominent way to create precise and understandable explanations is to develop ML/AI algorithms which are intrinsically interpretable. Furthermore, complexity related methods can be further distinguished to i) intrinsically explainable (Ante-Hoc) models, which are also referred as transparent or glass box approaches and ii) forecasting black-box (Post-hoc) models that require insights into the prediction’s reasoning process regarding the explainability source.

Intrinsic explainable models are specific by definition and include logistic regression, decision trees, k-nearest neighbors, rule-based learners and Bayesian models [Holzinger19]. [Ustun15] Ustun et al. proposed sparse linear models that used data-driven scoring mechanisms, which focus on providing domain experts a qualitative understanding of the inner models due to a high level of sparsity. Although these methods tend to be transparent by nature as highlighted in [Holzinger19], when dealing with [Molnar20] high-dimensional feature spaces in which one needs more sophisticated models (e.x. deep decision trees), the level of transparency and explainability drops significantly. As [Sarkar16] highlighted there is a fine line between accuracy and explainability. This challenge can be seen as a trade-off between models with high levels of complexity, and therefore the accuracy, and the effort required to explain them to domain experts. Back in 2001, Breiman noted that [Breiman01] *"accuracy generally requires more complex prediction methods . . .[and] simple and interpretable functions do not make the most accurate predictors"*.

Most recent work performed in Complexity related methods evolve around Black-Box explanations, also known as Post-Hoc explanations. This type of explanations is the exact opposite of Explanations by Design, since one starts from high, complex and difficult to interpret models without knowing the inner structures such as Support Vector Machines (SVM) with Non-Linear kernels, Neural Networks (NN) and random forests. Post-hoc

methods are usually model agnostic, and although they operate as “black-boxes” unable to provide a holistic explanation for the whole model, [Arrieta20] they may provide local explanations over particular decisions/classifications. Furthermore, Post-hoc explanations can further be analysed in perturbation-based and backpropagation-based methods. Perturbation-based methods, such as [Robnik08] Prediction Difference Analysis (PDA), are based on the importance of specific features given a high-dimensional feature spaces but cannot handle saturated classifiers. To tackle the problem of saturated classifiers in image processing [Ruth17] Ruth et al. proposed a variation called Meaningful Perturbations, which replaced regions of an image with constant values, noise or blurring on an image to measure changes in feature activations and classification scores. Extending the PDA [Zintgraf17] Zintgraf et al. removed several features at once by using prior knowledge about images and choosing patches of connected pixels as feature sets to analyze the effects of different window sizes on top scoring classes. The huge computational cost of this method was later minimized by [Gu19] through the Contextual Prediction Difference Analysis, which also solved the problem of saturated classifiers by producing a model-aware saliency map.

A saliency map [Adebayo18] is an image in which the brightness of a pixel represents how salient the pixel is i.e., the brightness of a pixel is directly proportional to its saliency. It is generally a grayscale image. Saliency maps are also called heat maps where hotness refers to those regions of the image which have a big impact on predicting the class to which the object belongs. The purpose of the saliency map is to find the regions which are prominent or noticeable at every location in the visual field and to guide the selection of attended locations, based on the spatial distribution of saliency. The difference between a saliency map and a heatmap lies in the fact that a saliency map shows properties of the picture/image (i.e. likelihood of attracting attention based on bottom-up features) while a heat map is a representation of gaze behavior data. So, they are based on different data: the image = saliency or the gaze behavior = heat map

Observing the saliency maps:

- Can you use the saliency maps to spot patterns that can be used to re-write an example to an adversarial one, which fools the model to predict the wrong label?
- Can you use the saliency maps on wrong predictions to find what the model fails to capture?

As there are a lot of explainability approaches and they can produce quite different saliency maps, it is important to be aware of their properties and be able to perform sanity checks as illustrated in Adebayo et al. [Adebayo18]. The most prominent properties of saliency maps are [Li20] Faithfulness and Stability. Since explanation techniques are employed to explain model predictions for a single instance, an essential property is that they are faithful to the model’s inner workings and not based on arbitrary choices. A well-established way of measuring this property is by replacing a number of the most-salient words with a mask token and observing the drop in the model’s performance. The notion of stability refers to testing whether instances with similar rationales also receive similar explanations. For simplicity, one could consider two instances to have similar rationales if the input is similar and the produced output is the same. A more consistent approach would be also to measure the similarity between the activation maps in the separate layers, which was not considered for computational reasons.

Yang et al. [Yang18] proposed the Global model Interpretation via Recursive Partitioning (GIRP), which creates a wider interpretation tree for a variety of ML/AI models based on local explanations. GIRP identifies whether ML/AI models tend to overfit particular patterns and alerts domain experts in cases of unreasonable behaviour. [Shrikumar16] The Deep Learning Important FeaTures (DeepLIFT) method, proposed by Shrikumar et al. uses derivative-based methodologies to propagate activation differences instead of gradients through the model; although the partial derivatives do not explain an isolated decision, they indicate which modifications of the feature space would propagate to the models' outcome differences. [Sundararajan17] Sundararajan et al. proposed an Integrated Gradients approach based on calculating attributions by multiplying input features with the average partial derivative, as the input feature space might be sparse. Class Activation Mapping (CAM), which was presented by Zhou et al. [Zhou16], relied on the observation that some convolutional layers behave as unsupervised object detectors, and uses global average pooling to create heatmaps of a pre-softmax layer. The generated heatmaps highlight the regions of a feature space that are most responsible for a classification task. [Ramprasaath17] Gradient-weighted Class Activation Mapping (GradCAM), which is an extension of CAM, uses the gradient information to rank Neuron activation in the last layer of a Convolutional Neural Network (CNN).

Using explanation methods to interpret the reasons that a particular decision was made (e.g., classification task of an outlier detection system), indicates that the explanation task occurs locally. Model-agnostic solutions are not defined by a particular type of ML/AI model since they separate the classification outputs from the explanations. [Adadi18] A variety of techniques using visualizations, example-based explanations and knowledge extraction methods are being enveloped. [Ribeiro16] Local Interpretable Model-Agnostic Explanation (LIME), proposed by Ribeiro et al. approximates a black-box model locally in the "area" of any prediction that a domain expert would like to focus on. The model learns a particular forecast related to the local region, by matching the given feature vector and perturbed inputs, to the results obtained from the reference model. The creators of LIME proposed an extension [Ribeiro18] of the original model, using decision rules. Since the local behaviour of the model can be non-linear, the authors propose using a set of if-then rules, which are intuitive and easy to understand. To explore the model's behaviour in the perturbation space, the authors apply multi-armed bandits to incrementally construct a set of rules, which generates candidate predicates and choose the one with the highest precision until a given precision threshold is reached. Similar to the extension of LIME presented by Ribeiro et al., Local Rule-Based Explanations (LoRE) [Guidotti18] proposes a parameter-free, two-step methodology that also provides rule-based explanations. It creates a balanced set of neighbour instances using a generic algorithm to explore the decision boundary of the data point(s) of interest and builds a decision tree classifier, which allows acquiring decision rules and counterfactuals.

[Lundberg17] The SHapley Additive exPlanations (SHAP) method introduced by Lundberg and Lee, uses a unified measure of feature importance based on the Shapley values, a concept from cooperative game theory. Multiple explanation models proposed by SHAP differ on how they approximate the computation of the SHAP values. The explanation models are called additive feature attribution methods and the construction of SHAP values allows to employ them both locally, in which each observation gets its own set of SHAP values, and globally, by exploiting collective SHAP values. In the field of image classification, two main explanators can be used for DL networks; [Lundberg17] DEEPSHAP and

[Lundberg18] Gradient-SHAP. [Lundberg17] DEEPSHAP is a high-speed approximation algorithm dedicated to DL models that are also connected to [Shrikumar16] DeepLift. In this scenario, the difference from the original [Shrikumar16] DeepLift model lies in the use of a baseline distribution of background samples instead of a single value and using Shapley equations to linearise non-linear components of the black-box such as max, softmax, products, divisions. Gradient-SHAP is based on [Schuchert10] IntGrad and [Milli19] SmoothGrad algorithms; IntGrad values require a single reference value to integrate from. As an adaptation to approximate SHAP values, Gradient-SHAP reformulates the integral as an expectation and combines that expectation with sampling reference values from the background dataset as done in SmoothGrad.

Several explainability techniques and methods have been applied in the field of manufacturing and more specifically on the predictive quality management domain (Quality 4.0) to boost transparency of deployed AI models. Goldman et al. used [Goldman21] XAI techniques such as CAM and Contrastive gradient-based saliency maps to explain black-box classifiers in quality welds in ultrasonically welded battery tabs. Lee et al. [Lee21] implemented several XAI methods to provide explanations for domain experts in defect classification of thin-film-transistor liquid-crystal display panels. Techniques such as CAM, LRP, integrated gradients, guided backpropagation, and SmoothGrad were implemented and visualized on a VGG-16 classification model. Based on the visualized results, LRP and guided backpropagation were selected for the creation of distributed heatmaps. Fitting the model into a decision tree and converting the prediction results into human interpretable text, the authors achieved an increased level of explainability as confirmed by a series of domain experts who performed multiple evaluations. In the area of manufacturing cost estimation, [Yoo20] described a method based on visualization of the machining features of a 3D computer aided design model that are influencing the increase in manufacturing costs. For the proposed purpose, a 3D gradient-weighted class activation mapping as the XAI method was applied.

3. Placement in the Overall Architecture

As detailed in previous sections XAI is crucial in building trust between non-expert stakeholders (e.g., workers, operators, production managers) and the manufacturing floor AI systems. It connects closely with the internals of AI algorithms and directly operates on their predictions and trained parameters, but also plays a critical role for their presentation and is intimately connected to the visualization and UI layers of the STAR architecture. As presented in the next sections XAI algorithms can be useful for many STAR scenarios, apart from the obvious interpretation of shop-floor AI decisions. Good examples are the poisoning and evasion attack detection capabilities it can bring to STAR’s AI - cyber security layer. For this reason, the XAI component was chosen to be designed as a library of XAI algorithms and their accompanying utilities. It will constitute a collection of state-of-the-art XAI algorithms and will be accessible to all work-package modules, holding a central place in the overall STAR architecture. Our initial and primary focus will be on the Human-Robot Collaboration domain in which the need for fostering trust between humans and AI systems is more central. The techniques created for this domain will be repurposed and expanded or further techniques will be developed to support the Attack to AI Detection modules of the Cyber Security domain. Finally, we will also take note and try to support relevant use cases arising from the Safety Domain in scenarios such as Fatigue Monitoring and Safety Zones detection.

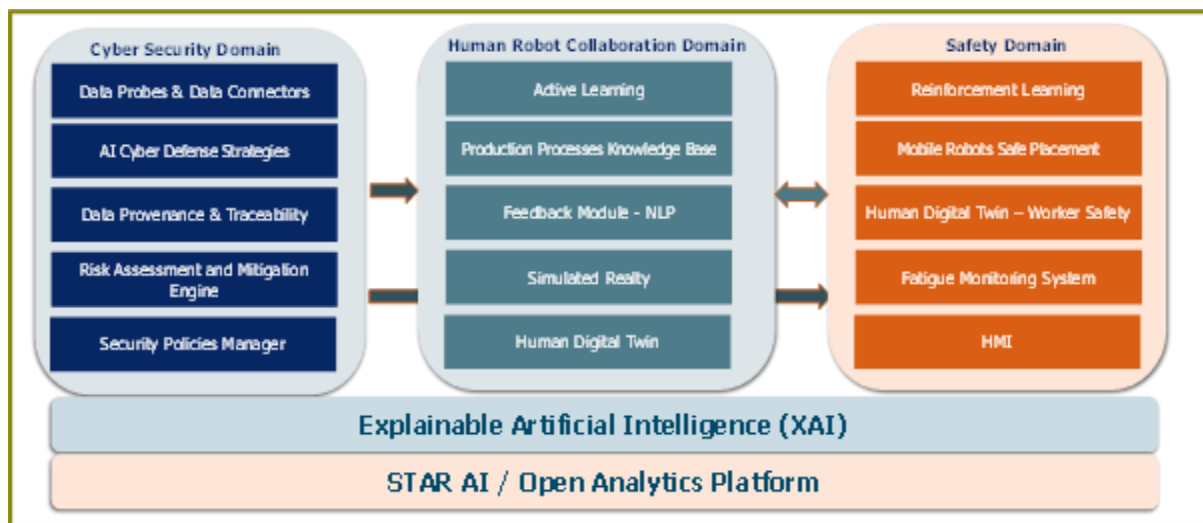


Figure 3: High-level Reference Model of STAR

This role is visible in the above high-level reference model of the STAR functionalities introduced in D2.6, showing XAI as a layer parallel to the STAR AI/ Open Analytics Platform and the same in its central role. Different modules, whether they are placed in the cyber-security, human-robot collaboration or safety domains, dependent upon it. Thus, XAI is of paramount importance in supporting defense strategies, active learning strategies employed during human-robot collaboration, as well as data generation for simulated reality and data augmentation modules and the development of human digital twins both of which are vital for AI safety.

3.1 Overall Placement

In Figure 4, from D2.6 we can see more specifically and in detail how our library of XAI algorithm fits in the architecture of the STAR project as well as its interaction with other components from WP3 and WP4.

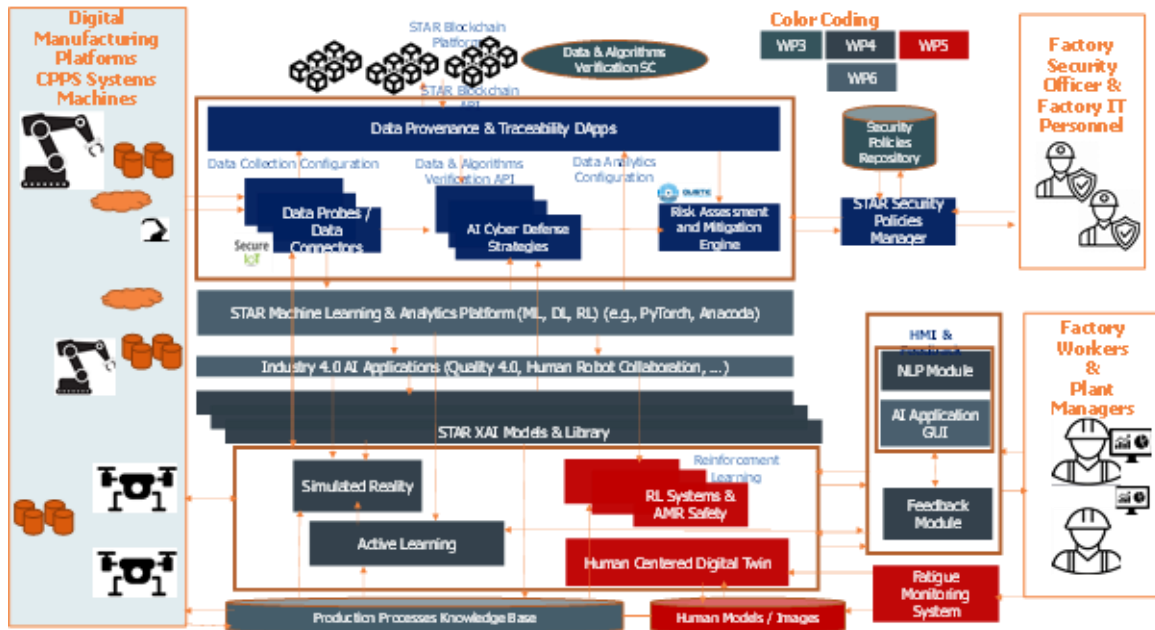


Figure 4: STAR Architecture

In a platform aiming to offer AI safety, reliability and trust, XAI should be associated with any component that contains complex AI logic, which is not immediately understandable by non-experts. In STAR there are many such modules that include complex ML algorithms, DL and reinforcement learning. XAI aims to shed light on the inner workings of these components and can be seen connecting with all of them, namely STAR’s Industry 4.0 applications such as Automated Quality Inspection (Quality 4.0) and the Human-Robot Collaboration Module, but also to the AI Cyber Defence Strategies (ACDS) module, which is responsible for producing data driven responses to attacks on AI-based STAR modules. There is also an implicit connection between the STAR XAI library and the User Interface Layer to facilitate STAR’s important goal of providing useful access to users to the AI models’ metadata and outcomes. The visualization capabilities of the chosen XAI algorithms for different STAR use-cases will play a key part in the component’s design. Finally, as a functional AI module of the STAR architecture, the XAI library will interact directly with the ML and analytics platform facilitating the actual execution of AI models. In case raw data is also needed in the future evolution of the XAI library, it will also make use of the data connectors developed for the STAR Data Storage Infrastructure as part of D2.4.

The XAI module itself will be responsible for the execution of XAI models and algorithms. Similar to other STAR modules (e.g., ACDS), it will support different types of XAI algorithms such as algorithms for explaining deep neural networks like Gradient-weighted Class

Activation Mapping (Grad-CAM) [Ramprasaath17] or general-purpose black-box algorithms like Local Interpretable Model-agnostic Explanations (LIME) [Ribeiro16] and Shapley Additive Explanations (SHAP) [Lundberg17] that interpret the outcomes of any type of AI model. Structured as a library of algorithms, XAI will provide method calls for the different algorithms it implements making them available to the several STAR modules that require explainable algorithms for their operation e.g., the AI Cyber Defence Strategies module and the Simulated Reality (SR) module.

3.2 Internal Architecture

The internal architecture of the XAI component is depicted in the following UML diagrams (Figure 5 and Figure 6). The diagrams illustrate both the execution of counterfactual logic and feature ranking based on the relevant importance of CNN features in the predictions of the AI model. These are the two of the main “internal” operations of the XAI module.

To produce counterfactual examples the XAI Library will be called while fitting the model to the full input dataset, in order to compute the feature statistics necessary for interpretability, depending on the underlying XAI algorithm. This will bring the XAI to a state where it can produce counterfactual examples, usually for all or part of the test data. The chosen algorithm will create perturbed instances based on the number of samples (n_samples) specified by the caller. For the perturbed instances, the algorithm will check what changes they bring to model predictions and return the collected counterfactuals back to the caller. Of course, the above steps will differ depending on the underlying implementation but the general interface for counterfactuals is expected to conform to the above blueprint.

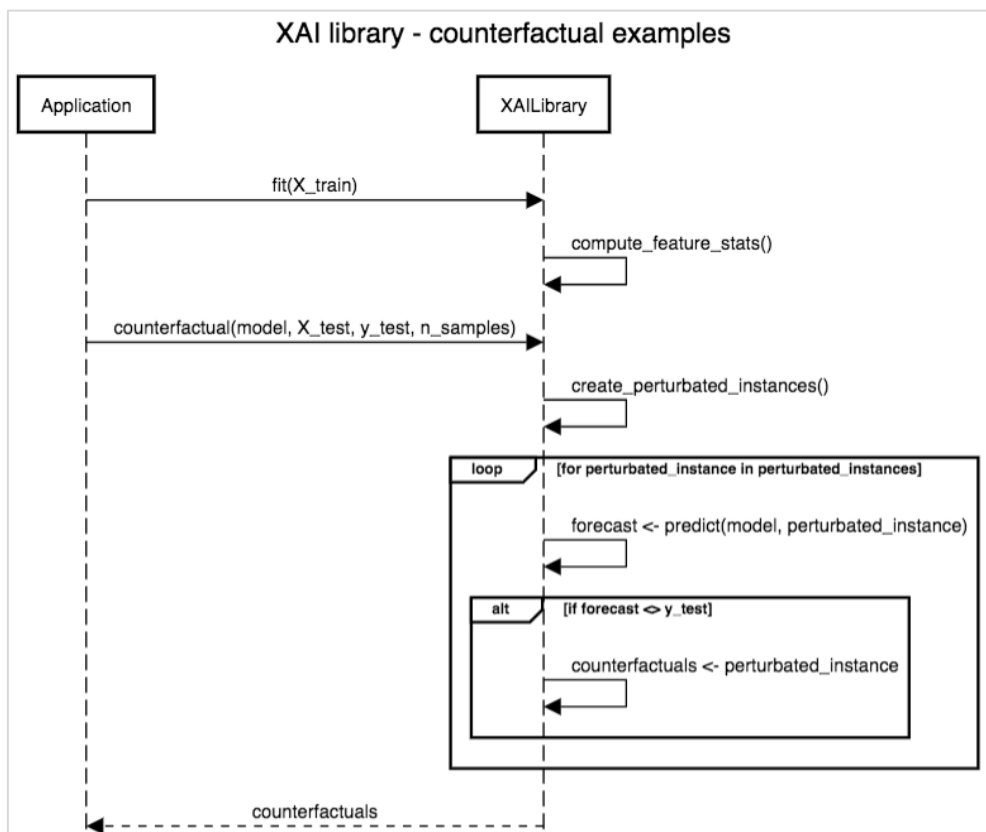


Figure 5: Provision of Counterfactuals Information by the STAR XAI.

Features ranking is displayed in the second figure of this section. The process starts again during the model.fit being applied to the training set, which will help the XAI algorithm prepare by collecting statistical information. Thereafter, that information will be available for use during dynamic calls with test samples, which will be answered by feature importance scores for the forecasts of the supplied instances. As in the previous example, the XAI algorithm will require access to the trained model and its .predict() method to perform interpretability computations. The feature importance calculation process will be, as before, based on the generation of perturbed instances for each input sample. The form of the result might vary according to the underlying method, but usually, for these types of methods, it will be a mapping of the feature to its importance score based on some difference score using the feature-forecast difference.

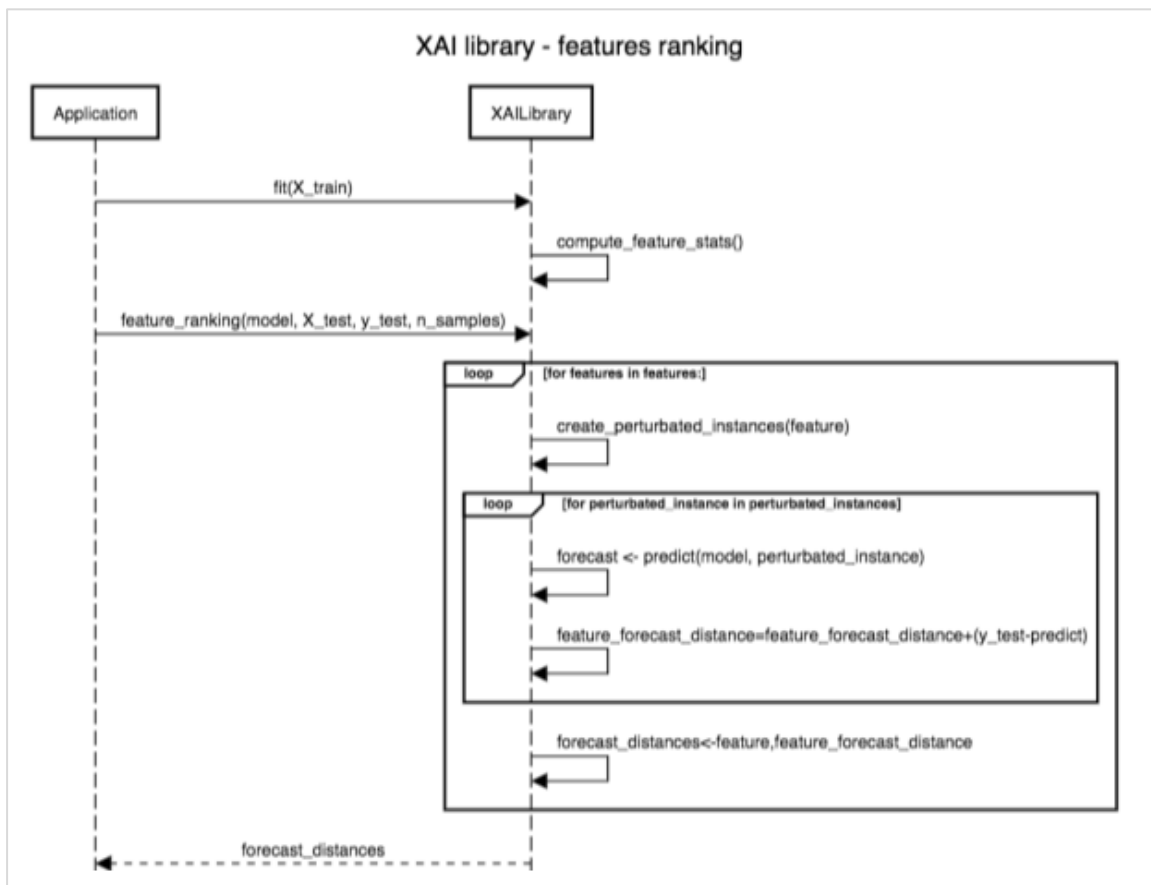


Figure 6: Features Ranking Provision by the STAR XAI.

Below we can also see what a typical workflow for the operation of feature scoring XAI systems looks like, as consisting of the following steps:

- **Data acquisition and input:** this initial step is concerned with gathering the necessary data for model training from various data sources and in various modalities as specified by the STAR component in different WPs. It might also include (depending on the XAI algorithm chosen) statistics useful for interpretation that can be extracted from the raw input data.

- **Machine Learning models:** ML models will utilize the training data provided from the previous layer for fitting the data according to the task the models need to perform. It is important that there is a correspondence between the choice of models, the data modalities and the XAI algorithms applied since not all XAI algorithms are suitable for all use-cases and not all models lend themselves to interpretation. The learned model will be a key part in the interpretability process, therefore the XAI library will need access to the prediction capabilities of the learned models (in the case of a black box method), but there are also methods that will need additional access to the model description, specific layer information for CNNs etc.
- **XAI methods utilization:** use-case specific XAI methods will be utilized initially to gather useful explainability information and statistics during model training. This step brings XAI algorithms to a better position to be applied to novel (test) samples and produce human understandable explanations during real-world operation.
- **Rules Extraction (optional):** running in parallel with the feature importance branch, this part of the workflow aims to extract human understandable rules. It is optional since it can be applied to a restricted number of modalities (e.g., tabular data), while it is incompatible with others (e.g., image data which better fit with heatmaps). An example of such a rule might be explaining a spike in demand due to the price of a good crossing a certain threshold, similar to what is produced by rules-based AI engines. To achieve this many methods try to learn a surrogate classifier (e.g., a decision tree) in parallel to a more complicated one (e.g., a neural network) and then directly extract to the rules from the surrogate classifier (tracing a path in the decision tree), which might work well and lead to clear-cut rules for certain areas of the state spaces.
- **Calculation of feature importance scores:** results of the previous step will be leveraged to perform on-the-fly calculations related to feature attribution through feature importance scores and provision of counterfactual information. The results of this step can serve as input to other modules such poisoning/evasion attack detection (WP3) or simulated reality (WP4).
- **Visualization of results:** To be human understandable, especially by non-expert stakeholders, XAI results should be properly visualized. This could be as simple as a table for vector inputs, to heatmaps overlayed over the original image, to more sophisticated graphs and plots highlighting feature importance.
- **Presentation of results to end users and domain experts:** this step can combine the results of different XAI and visualization methods to showcase the algorithms way of operating and rationale as a coherent whole, aiming to increase the trust of non-expert stakeholders.

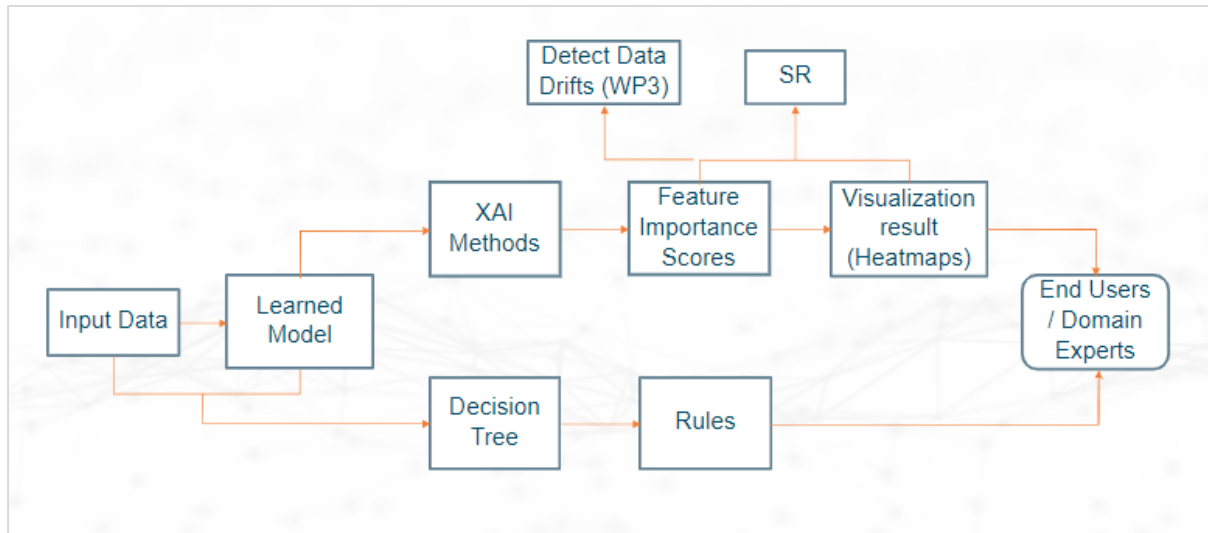


Figure 7 : XAI Internal Workflow.

The above diagrams are the blueprints for a library aiming to provide interpretability to different components of STAR in different WPs, ranging from the learning models used in the production line such as the automated quality inspection in PCL UC2 to the attack detection modules of WP3 and even potentially WP5 components such as reinforcement learning and safety zones detection. Component interactions are explained in detail in the following sub-section.

3.3 Component Interactions

As already mentioned, the XAI library plays a central role serving many different components across WPs 3, 4 and 5. The following are examples of the nature of these connections and the role XAI plays in these components.

- **AI Cyber Defence Strategy Module (ACDS):** Here XAI is used in two scenarios, namely defending against poisoning attacks and evasion attacks, which try to use malformed instances to corrupt AI models.
 - **Defending a Poisoning Attack:** The AI system’s expected functionality is compared with the explanation provided by the XAI module to determine any abnormalities. More specifically, after the model is trained with a given set of data, feature importance scores are obtained from the XAI library which are processed by the attack detection module and play an important role in the final verdict provided by the Risk Assessment module (see process diagram below).

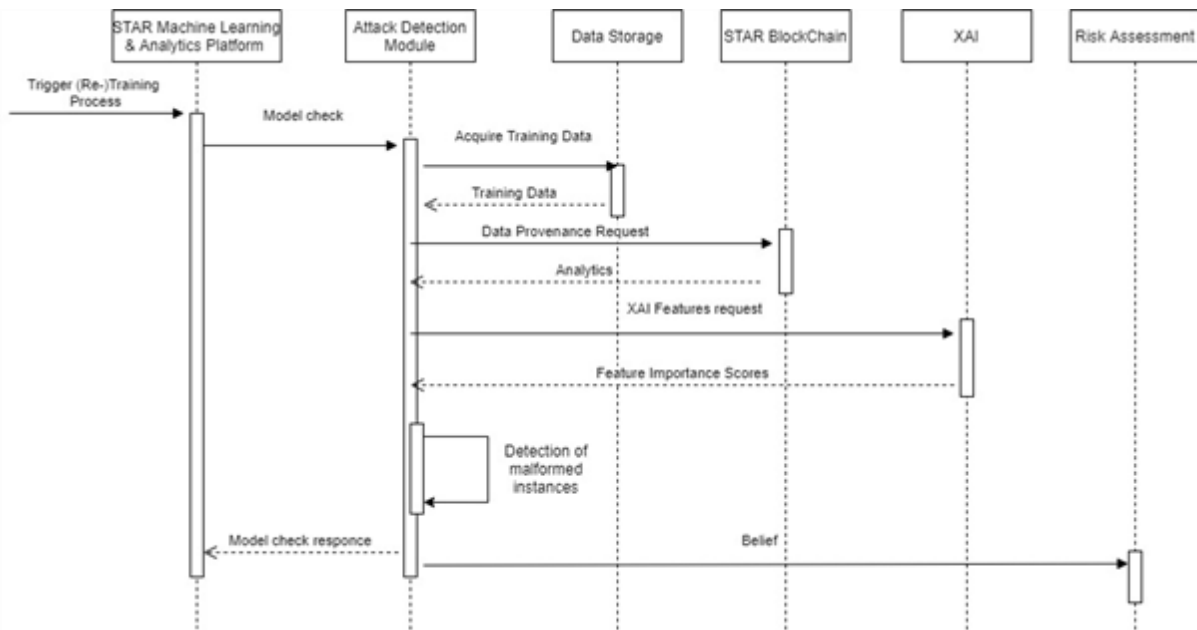


Figure 8: Information Flow for a Defending a Poisoning Attack.

- **Defending an Evasion Attack:** In order to defend against an evasion attack, the feature importance scores provided by means of the XAI library are also important. This time manufacturing floor data is fed directly to XAI during model operation and the requested scores are returned to the Attack Detection Module to detect malformed instances and continue with risk assessment and adversarial training.

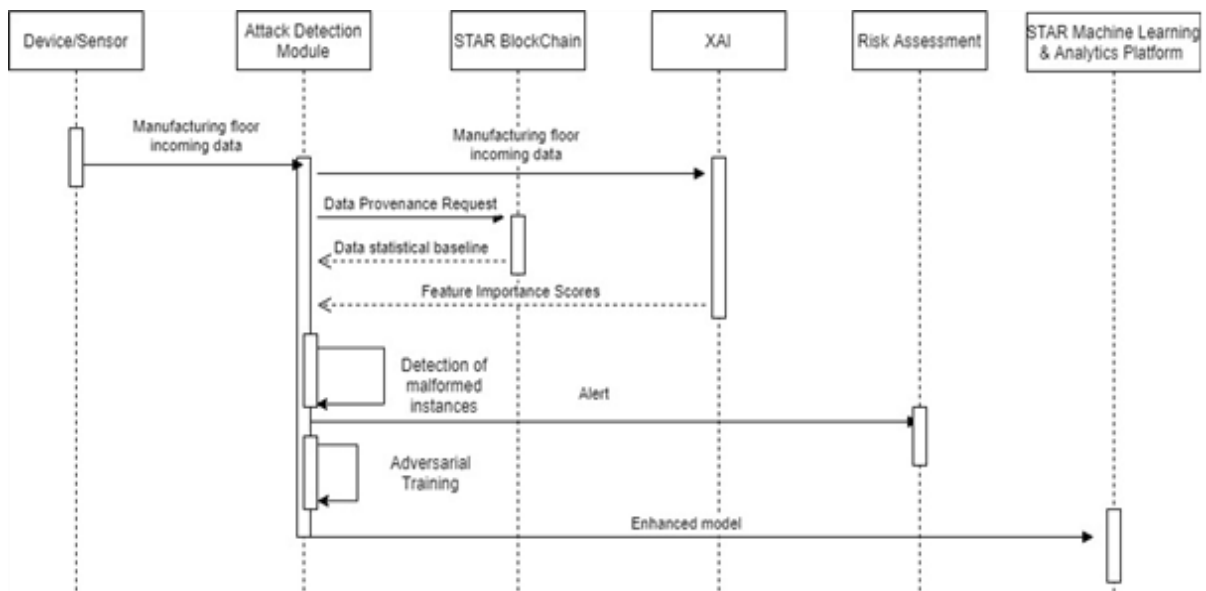


Figure 9: Information Flow for a Defending an Evasion Attack.

- Active Learning:** In Figure 10 one can see an Active Learning for Human-Robot Collaboration architecture diagram, detailed in a recent publication by STAR partners [Rozanec21] where XAI comes to play a key role. Here the XAI module processes forecasts originating from the Forecasting Module and leverages feature importance scores to determine the most important features for shaping the final forecast. The resulting explanations are then processed and tailored to the user that will see them according to their profile (e.g., depending on their role in their organization, their security permissions etc.). A novel idea is also introduced in that the XAI component incorporates human feedback to assess the quality and efficacy of the explanation and adjust the way explanations are computed, but also detect potential biases of the forecasting model.

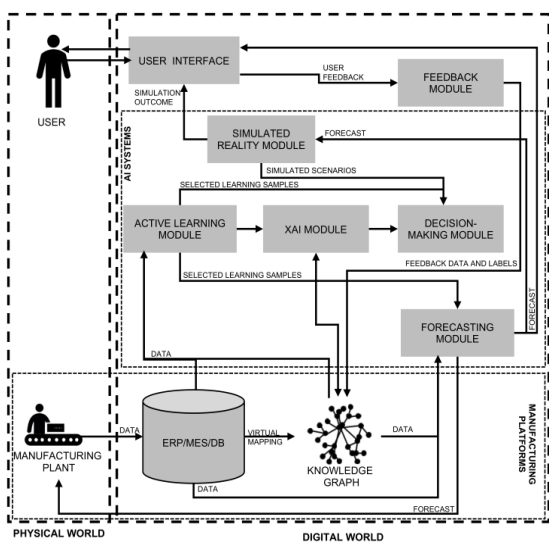


Fig. 1A

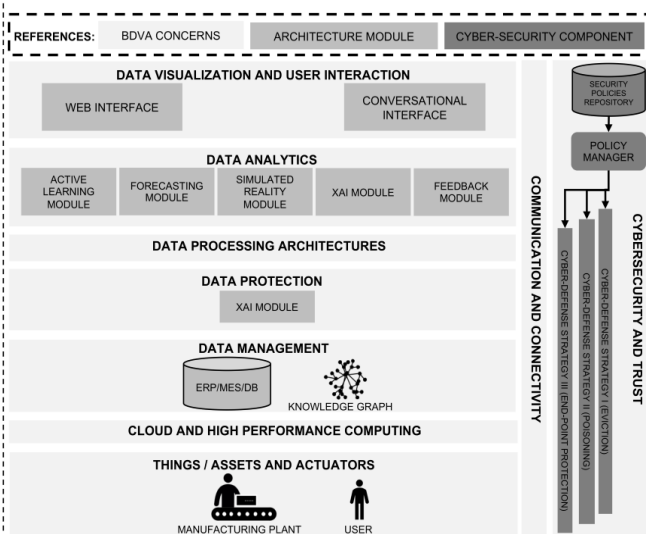


Fig. 1B

Figure 10: Active Learning Architecture from [Rozanec21]

- Simulated Reality:** Another component that can benefit from model explanation is the Simulated Reality component, which is responsible for generating data to help with the more efficient training but also with the assessment and robustification of the industrial AI models. In this context, XAI explanations can be leveraged in two ways. Firstly, to create novel instances e.g., in automated quality inspection transfer defect features from one dataset to another to check if algorithms are robust to unseen defects. Secondly, to interpret the data generation algorithms themselves and use those explanations to guide generation and produce synthetic inputs of better quality. Contrary to the previous example this interaction might require more complex explanations and patterns of interaction that go beyond general model agnostic algorithms by introducing the need to make use of model metadata. It could also spread to other novel research areas such as interpretability for advanced DL models like Generative Adversarial Networks (GANs) and Variational Autoencoders.

3.4 Implementation and Deployment

Since XAI is a library of algorithms its implementation will follow a different course from traditional software components. The most important factor that comes into play is the appropriate choice and evaluation of methods, which will, after rigorous testing and verification of their suitability for the various STAR use-cases, end up in the final library implementation.

The various implementations are at the moment using Python 3.x and make heavy use of DL libraries such as Tensorflow and PyTorch as well as scikit-learn. Especially in the DL model case, the availability of at least one GPU tailored for ML would be beneficial for performance although maybe not absolutely necessary depending on the methods implemented and on delay tolerances.

Depending on how the library evolves together with its interacting components and to ensure compatibility with all of them, two kinds of deployment are envisioned. The first is the use as a python package (e.g., a wheel package) downloaded via pip from the project's artifact repository and the second is the use of a docker container and the provision of the XAI library as a service accessible through a suitable API. One of the two or both options could be implemented in the future according to the project's needs.

Finally, there are a variety of off-the-shelf XAI libraries that have or can be leveraged for the implementation of this component or can alternatively provide baselines for comparisons.

Some notable ones are the following:

- **lime [Lime]:** Accessible through pip, the library implements the Local Interpretable Model-agnostic Explanations family of methods for multiple modalities such as vectors, images and text. Receives the model.predict method as input and is not dependent on underlying implementations. No GPU is needed unless the model is really complex and many instances need to be interpreted.
- **shap [Shap]:** Implements Shapley additive explanations and is from a usage viewpoint similar to lime without being tied to a specific model and ML/DL framework. Also extends to images by using random masking.
- **EthicalML [EthicalML]:** Available through the xai pip package and developed by IBM, provides many additional utilities for data loading, operations and results visualizations aiming to be an end-to-end solution. Only dependence is on scipy and scikit, currently, maybe not mature enough for complex DL models.
- **Dalex [Dalex]:** It is a very good all-around library in the dalex pip package providing wrappers for different ML frameworks as well as XAI methods such as SHAP and LIME and can be easily integrated with both scikit and keras (on top of tensorflow) as well as xgboost, mlr and mlr3. Also includes impressive visualization tools and dashboards.
- **InterpretML [InterpretML]:** Implemented by Microsoft and supporting SHAP, Lime, Partial Dependence Plots (PDP) and Sensitivity Analysis among others. While a good end-to-end toolbox, currently has limited support for image and text data.
- **Grad-CAM [GradCAM]:** A family of methods specific to CNNs. Many variants have been implemented in the pip grad-cam package, currently implemented in PyTorch and also supports GPU usage. Note that these methods most often require some access to the actual model (e.g., weights, architecture) to perform gradient calculations and do not treat the model as a black box.

- **SeFa [Shen21][SeFa]:** An interpretability method for generative adversarial networks. Very fast as it is based on matrix calculations on top of existing weights and does not use any learning but currently only supports StyleGAN and PG-GAN. Written in and supporting PyTorch. A GPU, while not necessary for SeFa, it is needed to load the GANs which are very heavy. Currently not under a pip package, so the code (open-source) needs to be packaged individually and loaded as a dependency.

Our intention is not only to take full advantage of these methods and apply the best one for each specific STAR use-case, but also to develop our own extensions and improvements as especially some of our challenging use-cases (e.g., automated defect recognition) might stretch these more general-purpose methods. For example, feature importance algorithms successful in explaining DL models that classify across widely differing classes (e.g., recognizing desk supply objects) might have difficulties in a defect detection use-case where images from different classes could be very similar, as they depict the same object but only differing in a small scratch.

4. XAI Algorithms

This section documents the XAI models, techniques, and corresponding meta-XAI models that have been under study over the first 14 months of the STAR project. The main purpose of the XAI component is to deploy an interface that will integrate different XAI algorithms applicable to the project’s pilots and use cases. In that direction, the proposed framework will integrate well-established XAI solutions and models making the appropriate modifications, when required, adding value to STAR-specific pilots. Furthermore, part of the library will be used by the cyber-defence component (or other(s) if requested).

During the first months of the project within the T4.1, the aim was to collect the state-of-the-art XAI models and identify the most suitable to STAR use-cases. It should be mentioned that the suitability of an XAI algorithm is dependent on the type of provided data from the manufacturing partners and the related AI-enabled components of STAR leveraged by each pilot. Each XAI algorithm can be used under specific datasets (i.e. tabular, text data, images, or time-series) and coupled with different ML/DL models for more accurate performance.

To this end, a summary of the provided datasets should be introduced, derived mainly by D2.1 and D2.4.

Table 1: XAI Algorithms Per Pilot Dataset

Data	AI Models and Techniques	Outcomes	Proposed XAI Techniques	Proposed Visualisations/ API/Interfaces
<i>Demonstrator # 1 Human-CoBot Collaboration for Robust Quality Inspections</i>				
Colour Images	Reinforcement learning system & simulation	How to handle a certain part.	CAMgrad	Heatmap, Saliency Map
Greyscale Images		Extended categorized dataset	CAMgrad	Heatmap, Saliency Map
Operator feedback (text or tabular)			DT builtin Feature Importance, SHAP, LIME, DeepLIFT	Barplots
Process data and asset data (tabular)		Operator data linked to process data	DT builtin Feature Importance, SHAP, LIME, DeepLIFT	Barplots
<i>Demonstrator # 2 Human Centred AI for Agile Manufacturing 4.0</i>				
sensors data (timeseries)	TBD	TBD	NBEATS, timeLIME, timeSHAP	
<i>Demonstrator # 3 Human Behaviour Prediction and Safe Zone Detection</i>				

time-series data of IMU and capacitive sensors (text)		Classification (discrete outputs)	DT builtin Feature Importance, SHAP, LIME, DeepLIFT	
3D view of the factory layout (text or tabular or image)		XML-based obstacle coordinates	DT builtin Feature Importance, SHAP, LIME, CAMgrad	
Sensor Data from Odometry Module (tabular)		Position, Direction and Orientation (multidimensional scalar output)	DT builtin Feature Importance, SHAP, LIME, CAMgrad	
Video footprints from Cameras (greyscale images)		Heatmaps highlighting	CAMgrad	

The following sections analyse the project’s XAI solutions which are categorized with respect to the input data types. Thus, we study and develop specialized XAI algorithms for time series, tabular, text, and image data.

4.1 XAI for Timeseries

All of the methods listed below apply to time series. First, we report post-hoc methods that approach the behaviour of a model by exporting relationships between feature values and predictions, in this case, feature lags of time series also play a role. While the ante-hoc methods incorporate the explanation in the structure of the model, which is therefore already explainable at the end of the training phase.

The first category includes variants of the widely used [Lime] LIME, [Hall21] k-LIME, [Zafar19] DLIME, [Sokol20] LIMETree and [Shap] SHAP (e.g [Bento20] TimeSHAP) as well as [Ribeiro18] Anchors, Local Foil Trees, or [Guidotti18] LoRE. Surrogate models are built for each prediction sample in most of these techniques, learning the behaviour of the reference model in the particular instance of interest by adding perturbations (or masking) to the feature vector variables. The numerous feature disturbance techniques for assessing the contribution of features to the projected value when they are deleted or covered are nearly a distinct topic of research in this context. These XAI models may be used for DNN since they are unaffected by the underlying ML model. The CAM, [Kashiparekh19] ConvTimeNet, which belongs to this group but intervenes in the model structure, is particularly interesting. The FIT framework, which evaluates the significance of observations for a multivariate time series, is also worth highlighting.

Beyond that, in DL models, things get more complicated when it comes to ante-hoc approaches. It is worth mentioning the Gradient-Input method, which calculates the activations of neurons and filters for a specific case by multiplying the input by the partial derivative of a layer relative to the input. Similar approaches followed, such as [Shrikumar16] DeepLIFT, and Smooth-Grad. The introduction of the attention layer is also

considered as a source of explanation, as it provides information about the time points related to the forecast.

Apart from these, it is worth mentioning approaches such as [Choi16] RETAIN (Reverse Time Attention) with application to Electronic Health Records (EHR) data. RETAIN achieves high accuracy while remaining clinically interpretable and is based on a two-level neural attention model that detects previous visits. Finally, [Oreshkin19] NBEATS focuses on an architecture based on residual back-and-forth connections and a very deep stack of fully connected layers. The architecture has a number of desirable properties, as it is interpretable, applicable without modification to a wide range of target areas.

Within the STAR project, we envision to integrate and partially modify the NBEATS model, as we believe that it will be a nice fit for the time series data of the project. NBEATS is an interesting step in applying DL to time series because it crafts an architecture dedicated to time-series. The previous approach consists in translating sequences (sequences to sequence). Timepoints are given to the network one after the other, and the network updates some internal memory in order to update the internal representation of state of the system. Then the output is computed using internal state representation and current output.

Recurrent neural networks do only this, while LSTM uses several different mechanisms to explicitly compute what parts to forget and what parts to update given current input.

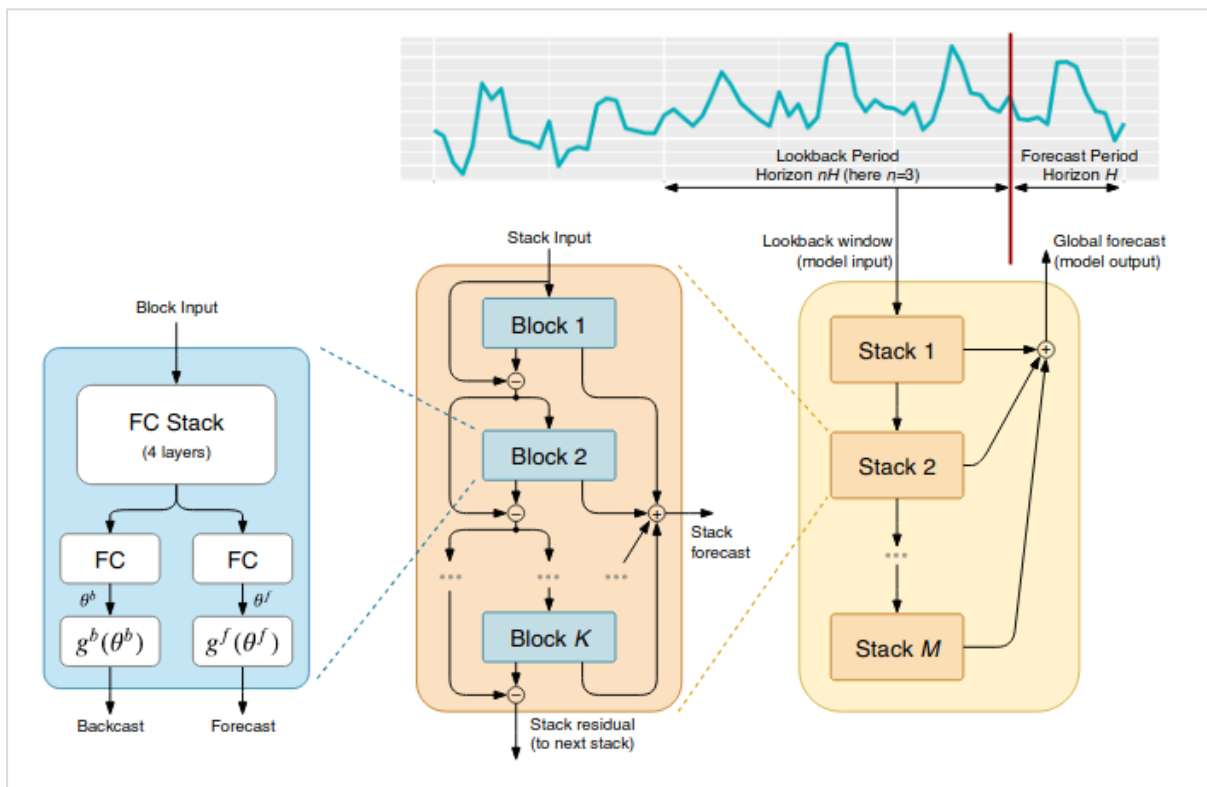


Figure 11: N-BEATS

Source: 1905.10437.pdf (arxiv.org), "N-BEATS", Oreshkin et al (2020)

NBEATS uses a completely different approach: it takes an entire window of past values and computes many forecast timepoints values in a single pass. For doing so, it uses extensively fully connected layers. It is made up of several blocks that are connected in a residual way: the first tries to model the past window (backcast) and future (forecast) as accurately as

possible, the second tries to model only the residual error of the previous block's past reconstruction (and updates the forecast based on this error), and so on. The forecast is the sum of predictions from several blocks, where the first block catches the main trends, the second specializes in smaller errors, and so on. This residual architecture also has the advantages of boosting/ensembling technique (used in classical ML as the forecast is the sum of predictions from several blocks. Some specialized trend and frequential blocks also can be used, where the block learns parameters of given functions, i.e., polynomial trends and sine/cosine with several frequencies.

4.2 XAI for Tabular Data

Working with an interpretable representation of the input that is understood by humans is a key requirement for LIME. A BoW vector for NLP or an image for computer vision are examples of interpretable representations. Dense embeddings, on the other hand, are difficult to comprehend, and using LIME is unlikely to help.

LIME generates a list of explanations that indicate the contribution of each characteristic to the data sample prediction. This enables for local interpretation as well as determining which feature adjustments will have the most impact on the forecast.

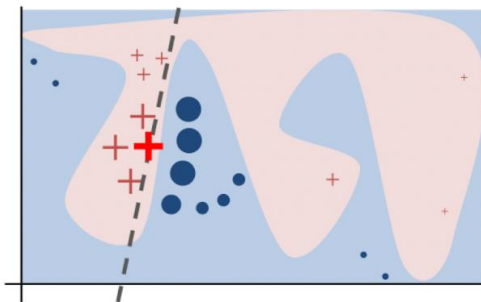


Figure 12: LIME

Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f , Number of samples N

Require: Instance x , and its interpretable version x'

Require: Similarity kernel π_x , Length of explanation K

$\mathcal{Z} \leftarrow \{\}$

for $i \in \{1, 2, 3, \dots, N\}$ **do**

$z'_i \leftarrow \text{sample_around}(x')$

$\mathcal{Z} \leftarrow \mathcal{Z} \cup (z'_i, f(z_i), \pi_x(z_i))$

end for

$w \leftarrow \text{K-Lasso}(\mathcal{Z}, K) \triangleright$ with z'_i as features, $f(z)$ as target

return w

Source: Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pages 1135–1144, 2016

4.3 XAI for Text Data (NLP)

In addition to the main objective mentioned above, it is of interest that NLP modules can be incorporated in other parts of the architecture, so that designers of AI solutions for the industry can apply it to their use cases. Natural Language Processing is the topic of how computers interact with human (natural) languages, notably how to train computers to process and analyze huge volumes of natural language data. The main research topics include question answering, conversational agents, natural language interpretation, and generation. To that aim, T4.1 will explain the objective of employing text data in STAR use-cases as well as the provided text dataset.

However, recognizing the value of features or entities is crucial in the XAI of NLP models, since it seeks to identify which part of speech is driving the most significant information. Explaining the rationale for question sequencing in dialogue, troubleshooting a plan-based

dialogue system, or explaining the produced utterances, to name a few examples. There are numerous approaches to determining the most representative entities in a text categorization task. Some works extract plan-based models in order to comprehend the goal and rationale of the discourse. On the one hand, ML-based techniques that focus on important items in text are limited to statistics-based explanations.

4.4 XAI for Images

The main concept of this component is to provide explanations in terms of attribution scores. Given a trained model, such as a neural network, the component will output the importance of each input feature for a particular prediction. When dealing with image data, feature importance can be translated into the importance of each pixel to the output forecast. The latter can also be visualized into a heatmap where the importance of each feature (pixel) can be displayed with different colours and can work as an excellent explanation for the human operator.

The integrated explainable AI (XAI) algorithms are applied to the model after it has completed its training phase (post-hoc). They also use various algorithms and strategies to generate local and global explanations. As far as image type data are concerned, there are two types of explainability approaches: gradient-based and perturbation-based. Gradient-based methods require several backward runs through our network before calculating the significance scores, while perturbation-based methods just require forward passes once the input has been changed. DeepLIFT and [Montavon19] LRP (LayerWise Relevance Propagation) are state-of-the-art gradient-based explainability algorithms, whereas LIME (Local Interpretable Model Agnostic Explanations) and PDA (Prediction Difference Analysis) are perturbation-based explainability methods. The SHAP algorithm is another explainability method that uses game theory to provide explanations (Shapley Additive exPlanations).

The abovementioned algorithms that are integrated into the component are mainly developed on Python Libraries and can easily be applied to the STAR project’s use cases. Once again LIME is one of the handiest techniques (with interpretable visualizations) for image type of input data.

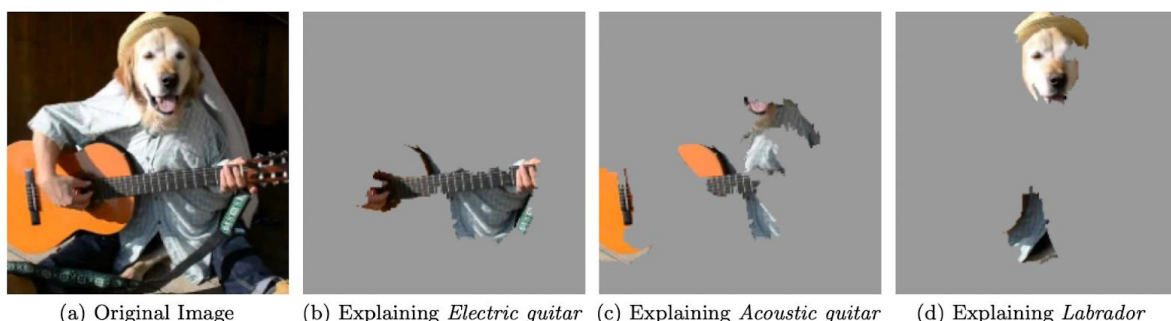


Figure 13: Image Explanations Using LIME

Source: Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “why should i trust you?” explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pages 1135–1144, 2016

Within the context of T4.1, special focus will be paid on modification of the LIME framework, making it either faster in terms of performance. To this end, LIME may be coupled with a custom Image pre-processing approach making the images “lighter” to be consumed.

4.4.1 Repurposing XAI for the PCL use case

As our first step we are tackling the Automated Quality Inspection use case of Pilot #1. The aim of using AI in the use-case is to reduce the cost and time of manual inspection and shift workers to more meaningful and less repetitive tasks. XAI will play an important role here as an intermediary between the AI and the human operator explaining the predictions produced by CNNs operating on image data and trying to detect defects in manufactured parts. The aim is not only to increase the trust in AI decisions but also to assist workers in performing their work, especially in difficult inspection cases where human input is necessary.

In the context of the above use case we envisioned the use of XAI algorithms to hint the user where the model believes a defect can be located, and thus speed up the defect detection in a manual revision context. We set up multiple experiments, to measure how different settings affect users’ labelling accuracy, and understand whether the hints provided to them were good enough, or not. The hints we provided to the users were created with XAI algorithms, such as GradCAM, but also included other approaches, such as images resulting from an unsupervised detection (e.g., DRAEM [Zavrtanik21]), and the nearest labelled image. Further experiments will be conducted in the future, to refine our understanding on the usefulness and perception of the information provided by the XAI algorithms.

5. Conclusions

This document aimed to provide an overview of the methodologies and algorithms that have been selected and/or will be extended for the purposes of STAR project. Although many XAI approaches are under evaluation, an interesting perspective for extension or modification of the underlying models is the User Interfaces or visualizations which will serve as the XAI component's output. Each specific type of XAI solution may have a different representation of the results regarding the stakeholders or the usage of it. The output of a XAI model may as well be injected in another STAR-component (i.e., input to other modules such poisoning/evasion attack detection (WP3) or simulated reality (WP4)) and should differ from the one that will be represented to domain experts. Thus, one particular goal of the next phase of the requirement-gathering process should be to identify which types of visualizations were needed most by the pilots and for what purpose. When that information becomes available, we will exploit the most common visualizations to explore XAI results and to communicate them.

As a future roadmap, we will be starting from Human-Robot Collaboration use-cases where we are already implementing XAI methods for defect hinting during the AI-assisted quality inspection. After that, we aim to expand and develop additional techniques for poisoning and evasion attack detection assisting the Cyber Security components of STAR. Finally, we will attempt to repurpose our methods and support relevant use-cases in other WPs of the project such as the Fatigue Monitoring and Safety Zones Detection systems.

References

Reference	Name of document
[Adadi18]	A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," in IEEE Access, vol. 6, pp. 52138-52160, 2018, doi: 10.1109/ACCESS.2018.2870052.
[Adebayo18]	Adebayo, Julius and Gilmer, Justin and Muelly, Michael and Goodfellow, Ian and Hardt, Moritz and Kim, Been. Sanity Checks for Saliency Maps. Advances in Neural Information Processing Systems 31 (NeurIPS 2018), vol. 31, 2018. https://proceedings.neurips.cc/paper/2018/file/294a8ed24b1ad22ec2e7efea049b8737-Paper.pdf
[Ahmed22]	I. Ahmed, G. Jeon and F. Piccialli, "From Artificial Intelligence to eXplainable Artificial Intelligence in Industry 4.0: A survey on What, How, and Where," in IEEE Transactions on Industrial Informatics, doi: 10.1109/TII.2022.3146552.
[Arrieta20]	Arrieta, A.B.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; García, S.; Gil-López, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Inf. Fusion 2020, 58, 82–115
[Bento20]	Bento, J.; Saleiro, P.; Cruz, A.F.; Figueiredo, M.A.; Bizarro, P. TimeSHAP: Explaining recurrent models through sequence perturbations. arXiv 2020, arXiv:2012.00073.
[Bhatt20]	Bhatt, U., Xiang, A., Sharma, S., Weller, A., Taly, A., Jia, Y., ... & Eckersley, P. (2020, January). Explainable machine learning in deployment. In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (pp. 648-657).
[Breiman01]	L. Breiman, "Statistical modeling: The two cultures (with comments and a rejoinder by the author)," Stat. Sci., vol. 16, no. 3, pp. 199–231, 2001.
[ChineseCoun17]	State Council Chinese Government: Development Plan for New Generation Artificial Intelligence (2017). http://www.gov.cn/zhengce/content/2017-07/20/content_5211996.htm
[Choi16]	Choi, E., Bahadori, M. T., Sun, J., Kulas, J., Schuetz, A., & Stewart, W. (2016). Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. Advances in neural information processing systems, 29.
[Choo18]	J. Choo and S. Liu, "Visual Analytics for Explainable Deep Learning," in IEEE Computer Graphics and Applications, vol. 38, no. 4, pp. 84-92, Jul./Aug. 2018, doi: 10.1109/MCG.2018.042731661.
[Dalex]	https://dalex.drwhy.ai/
[EthicalML]	https://github.com/EthicalML/xai
[Goldman21]	Claudia V Goldman, Michael Baltaxe, Debejyo Chakraborty, and Jorge Arinez. Explaining learning models in manufacturing processes. Procedia Computer Science, 180:259–268, 2021.
[Goodman17]	Goodman, B., Flaxman, S.: European union regulations on algorithmic decision-making and a "right to explanation". AI Mag. 38(3), 50–57 (2017)
[GradCAM]	https://github.com/jacobgil/pytorch-grad-cam
[Gu19]	Jindong Gu and Volker Tresp. Contextual prediction difference analysis for

	explaining individual image classifications. arXiv preprint arXiv:1910.09086, 2019.
[Guidotti18]	Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. Local rule-based explanations of black box decision systems. arXiv preprint arXiv:1805.10820, 2018.
[Gunning16]	D. Gunning. Explainable Artificial Intelligence (XAI) DARPA-BAA16-53. Technical report, Defense Advanced Research Projects Agency (DARPA), 2016.
[Hall21]	Hall, P.; Gill, N.; Kurka, M.; Phan, W. Machine Learning Interpretability with H2O Driverless AI. H2O. AI. 2017. Available online: http://docs.h2o.ai/driverless-ai/latest-stable/docs/booklets/MLIBooklet.pdf (accessed on 5 August 2021).
[Holzinger19]	Holzinger, A.; Langs, G.; Denk, H.; Zatloukal, K.; Müller, H. Causability and explainability of artificial intelligence in medicine. <i>Wiley Interdiscip. Rev. Data Min. Knowl. Discov.</i> 2019, 9, e1312.
[InterpretML]	https://github.com/interpretml/interpret
[Kashiparekh19]	Kashiparekh, K., Narwariya, J., Malhotra, P., Vig, L., & Shroff, G. (2019, July). ConvTImenet: A pre-trained deep convolutional neural network for time series classification. In 2019 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.
[Lee21]	Minyoung Lee, Joohyoung Jeon, and Hongchul Lee. Explainable ai for domain experts: a posthoc analysis of deep learning for defect classification of tft–lcd panels. <i>Journal of Intelligent Manufacturing</i> , pages 1–13, 2021.
[Letham15]	B. Letham, C. Rudin, T. H. McCormick, and D. Madigan, “Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model,” <i>Ann. Appl. Statist.</i> , vol. 9, no. 3, pp. 1350–1371, 2015.
[Li20]	Xiao-Hui Li and Yuhan Shi and Haoyang Li and Wei Bai and Yuanwei Song and Caleb Chen Cao and Lei Chen (2020). Quantitative Evaluations on Saliency Methods: An Experimental Study. arXiv:2012.15616
[Lime]	https://github.com/marcotcr/lime
[Lundberg17]	Lundberg, Scott M and Lee, Su-In A Unified Approach to Interpreting Model Predictions, <i>Advances in Neural Information Processing Systems</i> , volume 30, 2017.
[Lundberg18]	Scott M Lundberg, Bala Nair, Monica S Vavilala, Mayumi Horibe, Michael J Eisses, Trevor Adams, David E Liston, Daniel King-Wai Low, Shu-Fang Newman, Jerry Kim, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. <i>Nature biomedical engineering</i> , 2018.
[Milli19]	Smitha Milli, Ludwig Schmidt, Anca D. Dragan, and Moritz Hardt. 2019. Model Reconstruction from Model Explanations. In <i>Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT* '19)</i> . Association for Computing Machinery, New York, NY, USA, 1–9. DOI: https://doi.org/10.1145/3287560.3287562
[Molnar20]	Molnar, C.; Casalicchio, G.; Bischl, B. Interpretable Machine Learning—A Brief History, State-of-the-Art and Challenges. arXiv 2020, arXiv:2010.09337
[Montavon18]	G. Montavon, W. Samek, and K.-R. Muller, “Methods for interpreting and understanding deep neural networks,” <i>Digital Signal Process.</i> , vol. 73, pp. 1–

	15, 2018.
[Montavon19]	Montavon, G., Binder, A., Lapuschkin, S., Samek, W., & Müller, K. R. (2019). Layer-wise relevance propagation: an overview. <i>Explainable AI: interpreting, explaining and visualizing deep learning</i> , 193-209.
[Oreshkin19]	Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2019). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. arXiv preprint arXiv:1905.10437.
[Ramprasaath17]	Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In <i>Proceedings of the IEEE international conference on computer vision</i> , pages618–626, 2017.
[Ribeiro16]	Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. In <i>Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining</i> , pages 1135–1144, 2016
[Ribeiro18]	Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In <i>Proceedings of the AAAI Conference on Artificial Intelligence</i> , volume 32, 2018.
[Robnik08]	Marko Robnik-Šikonja and Igor Kononenko. Explaining classifications for individual instances. <i>IEEE Transactions on Knowledge and Data Engineering</i> , 20(5):589–600, 2008.
[Rozanec21]	Joze M. Rozanec, Patrik Zajec, Klemen Kenda, Inna Novalija, Blaz Fortuna, Dunja Mladenic, Entso Veliou, Dimitrios Papamartzivanos, Thanassis Giannetsos, Sofia-Anna Menesidou, Rubén Alonso, Nino Cauli, Diego Reforgiato Recupero, Dimosthenis Kyriazis, Georgios Sofianidis, Spyros Theodoropoulos, John Soldatos: "STARdom: an architecture for trusted and secure human-centered manufacturing systems", In the Proc. Of the Advances in Production Management Systems (APMS) Conference, Sep 5-9, 2021.
[Ruth17]	Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In <i>Proceedings of the IEEE International Conference on Computer Vision</i> , pages3429–3437, 2017
[Sarkar16]	S. Sarkar, "Accuracy and interpretability trade-offs in machine learning applied to safer gambling," in Proc. CEUR Workshop, 2016, pp. 79–87.
[Schuchert10]	T. Schuchert, T. Aach and H. Scharr, "Range Flow in Varying Illumination: Algorithms and Comparisons," in <i>IEEE Transactions on Pattern Analysis and Machine Intelligence</i> , vol. 32, no. 9, pp. 1646-1658, Sept. 2010, doi: 10.1109/TPAMI.2009.162.
[SeFa]	https://github.com/genforce/sefa
[Shap]	https://shap.readthedocs.io/en/latest/index.html
[Shen21]	Shen, Y., & Zhou, B. (2021). Closed-Form Factorization of Latent Semantics in GANs. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 1532-1540.
[Shrikumar16]	Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. arXiv preprint arXiv:1605.01713, 2016.

[Shrikumar17]	Shrikumar, A., Greenside, P., & Kundaje, A. (2017, July). Learning important features through propagating activation differences. In International conference on machine learning (pp. 3145-3153). PMLR.
[Sokol20]	Sokol, K.; Flach, P. LIMETree: Interactively Customisable Explanations Based on Local Surrogate Multi-output Regression Trees. arXiv 2020, arXiv:2005.01427.
[Sundararajan17]	Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In International Conference on Machine Learning, pages 3319–3328. PMLR, 2017.
[Turek17]	Turek, M.: DARPA - Explainable Artificial Intelligence (XAI) Program (2017). https://www.darpa.mil/program/explainable-artificial-intelligence
[Ustun15]	B. Ustun and C. Rudin, "Supersparse linear integer models for optimized medical scoring systems," Mach. Learn., vol. 102, no. 3, pp. 349–391, 2015
[Weller17]	A. Weller, "Challenges for transparency," in Proc. Int. Conf. Mach. Learn. Workshop Human Interpretability ML, 2017.
[Xu15]	K. Xu et al., "Show, attend and tell: Neural image caption generation with visual attention," in Proc. Int. Conf. Mach. Learn. (ICML), 2015, pp. 1–10
[Yang18]	C. Yang, A. Rangarajan, and S. Ranka. (2018). "Global model interpretation via recursive partitioning." [Online]. Available: https://arxiv.org/abs/1802.04253
[Yoo20]	Soyoung Yoo and Namwoo Kang. Explainable artificial intelligence for manufacturing cost estimation and machining feature visualization. arXiv preprint arXiv:2010.14824, 2020.
[Zafar19]	Zafar, M.R.; Khan, N.M. DLIME: A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems. arXiv 2019, arXiv:1906.10263.
[Zahavy16]	T. Zahavy, N. Ben-Zrihem, and S. Mannor, "Graying the blackbox: Understanding DQNs," in Proc. Int. Conf. Mach. Learn., 2016, pp. 1899–1908.
[Zavrtanik21]	Zavrtanik, Vitjan, Matej Kristan, and Danijel Skočaj. "DRAEM-A discriminatively trained reconstruction embedding for surface anomaly detection." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.
[Zenisek19]	Jan Zenisek, Florian Holzinger, Michael Affenzeller, Machine learning based concept drift detection for predictive maintenance, Computers & Industrial Engineering, Volume 137, 2019, 106031, ISSN 0360-8352, https://doi.org/10.1016/j.cie.2019.106031 .
[Zhou16]	Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2921–2929, 2016.
[Zintgraf17]	Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. arXiv preprint arXiv:1702.04595, 2017.