A Review of Explainable Artificial Intelligence in Manufacturing

Georgios Sofianidis*, Jože M. Rožanec**, Dunja Mladenić***, and Dimosthenis Kyriazis

1 Department of Digital Systems, University of Piraeus, Piraeus, Greece
{george.sofianidis,dimos}@unipi.gr
2 Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
{joze.rozanec,dunja.mladenic}@ijs.si
3 Jožef Stefan International Postgraduate School, Jamova 39, 1000 Ljubljana, Slovenia

Abstract. The implementation of Artificial Intelligence (AI) systems in the manufacturing domain enable higher production efficiency, outstanding performance, and safer operations, leveraging powerful tools such as deep learning and reinforcement learning techniques. Despite the high accuracy of these models, they are mostly considered black boxes: they are unintelligible to the human. Opaqueness affects trust in the system, a factor that is critical in the context of decision-making. We present an overview of Explainable Artificial Intelligence (XAI) techniques as a means of boosting the transparency of models. We analyze different metrics to evaluate these techniques and describe several application scenarios in the manufacturing domain.

Keywords. Explainable Artificial Intelligence, Interpretability, Transparency, Taxonomy, Manufacturing

1 Introduction

The increasing digitalization of every aspect of life provides vast amounts of data, enabling the implementation of Artificial Intelligence (AI) models. The manufacturing and process industry is not an exception to this trend. AI models play a significant role in many aspects of the manufacturing process. AI models drive better quality by enhancing quality inspection and process monitoring in production lines, ease reconfiguration and customization of automated part handling, fault diagnosis and event prediction, more agile production management, flexible production planning, and enabling safe collaboration between humans and cobots. Especially the latter is a big step towards the transition into Industry 5.0, where the focus is on the synergy between humans and robots and the actors are collaborators instead of competitors.

AI models provide the means to automate many tasks and achieve unprecedented performance levels. However, in most cases, such models are opaque to the user: they work as black-boxes. Their predictions are mostly accurate, but no intuition behind the reasoning process is available to human users. Given the impact of those predictions on the decision-making processes, it is crucial to develop mechanisms and techniques to provide insights to users on such an AI model reasoning process. The development of such techniques and mechanisms and how those insights are presented has given birth to a research field of its own, known as Explainable Artificial Intelligence (XAI). While the field of XAI can be traced back to the 1970’s [14], it has experienced a new flourishing since the rise of modern deep learning [15].

* equal contribution
Though there is no single definition of the scope of this research field, most authors agree it includes intrinsically interpretable models and post-hoc explainability models (the model’s capability of being explained by another interpretable model). Authors identify two sources of model opacity (or opaqueness): (i) the complexity of the formal structure of the model is beyond human comprehension, or alien to human reasoning, or (ii) because the inner workings of the model cannot be shared (e.g., being considered a trade secret). Model opaqueness can be relative to expert knowledge: e.g., it can be opaque to an analyst but not to the machine learning engineer. \cite{14} introduced the term deep opacity to describe models whose opacity cannot be removed even by human experts. When presenting insights on the reasoning process of an AI model, the explanations should resemble a logic explanation, and take into account relevant context. \cite{19} considers context has three elements related to the explainee: (i) Profile (user profile, to whom we present the explanation), (ii) Objective (refer to the goals of the explanation, e.g., are the explanations meant to improve the model, enhance trust in the system, aid on decision-making or foster action based on decisions made), and (iii) focus (if the explanation is either global or local). In local explanations, the specific point of interest must be considered part of the context. When the explanations aim to aid decision-making or take action, they should provide information regarding actionable features.

XAI techniques and methods can be classified into three categories, considering the explainability source, the scope of the explanation, and the level of dependency on the forecasting model used (see Fig. 1.1). We distinguish intrinsically explainable models and forecasting models that require post-hoc models to get insights into the forecast’s reasoning process regarding the explainability source. Concerning the explanation’s scope, explanations can be global (describe the behavior of the whole model for the average of forecasts provided) or local (describe the model’s behavior for a particular forecast). Finally, regarding the dependency on the forecasting model’s explanation, we distinguish model-agnostic (can be applied to any AI model) or model-specific techniques (can be applied only to AI models built with a particular algorithm or type of algorithms).

Fig. 1.1. XAI taxonomy

In this chapter, we introduce the field of Explainable Artificial Intelligence, describing methods and techniques used to identify meaningful features driving forecasts, current approaches used to evaluate such models, applications and use cases in the industrial domain, and open challenges. When doing so, we do not consider intrinsically explainable models.

2 Methods and techniques

Different methods and techniques have been introduced to boost the transparency and acceptance of AI models and different taxonomies have been proposed in literature based on the explanation generating mechanism, the type of explanation, the scope of explanation, the type of model...
it can explain, or a combination of these features. [1] classified those methods into intrinsic interpretable models and post-hoc explanations and divided the latter to text explanations, visual explanations, local explanations, explanations by example, explanations by simplification, and feature relevance explanations techniques. [2] introduced a categorization of explanation methods based on the type of explanation returned and divided them based on the most common data types such as tabular, image, and text. For tabular data, feature importance is one of the most popular types of explanation returned by local explanation methods. The explainer assigns to each feature an importance value which represents how much that particular feature was important for the prediction under analysis. The sign and magnitude of each importance value are also considered to understand the contribution of each feature. Similar to the above but in the field of image classification, saliency maps can be used as explanations. Those are modeled as matrices with the same dimensions as that of the image we want to explain, and each element of the matrix represents the saliency of each pixel to the forecast. Another type of explanation that can be implemented on tabular data is the rule-based explanation. Human readable decision rules can give the end-user an explanation about the reasons that lead to the final prediction. A decision or factual or logic rule is a set of premises that lead to a specific forecast. Counterfactual rules are a set of rules that lead to the opposite of a specific forecast. [3] classified XAI techniques according to the type of explanation and the scope of explanation. The three types he distinguished are model-based, attribution-based, and example-based explanations. In this chapter, we present some of the well-known explainability methods based on the taxonomy introduced by [3].

The class of model-based explanations include methods that are either explainable by nature (intrinsic explainability) or methods that use a different interpretable model to explain the task model (post-hoc explainability). The first subclass can be divided into sparse linear classifiers (e.g., linear or logistic regression, generalized additive models (GAMs)), discretization methods (e.g., rule-based learners, decision trees), and example-based models (e.g., K-nearest neighbors). The second subclass includes interpretable surrogate models that can approximate the task model and can be used as post-hoc explanations.

The class of attribution-based explanations use the explanatory power of input features to explain the task model. These approaches are also known as feature (a.k.a variable) importance, relevance, or influence methods. Most post-hoc explanations fall under this category which can further be divided into perturbation-based and backpropagation-based methods.

Among the perturbation-based methods, we can find the Prediction Difference Analysis (PDA) [4], which is based on the idea that the relevance of an input feature concerning the class can be estimated by measuring how the predictions change if this particular feature is removed. This method cannot deal with saturated classifiers (models whose output does not change after removing part of the features). A similar approach for images was developed by [5] with the Deconvolutional Networks, which attempts to reconstruct the feature map into the layer input or the original image. The proposed networks used convolution, max-pooling layers, and the ReLU activation function. Sliding a gray-color square over the image, they measure changes in feature activations and the classification scores. A variation of this method was developed by [6], who, instead of using a gray-square, replaces regions of an image with constant values, noise, or performs some blurring on the image. This method was evolved by [7], who chose upsampled, random binary masks to perform the occlusions and analyzed their impact on the target class classification score. Another variation of [6] was introduced by [8], who 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 [9] through the Contextual Prediction Difference Analysis, which also solved the problem of saturated classifiers by producing a model-aware saliency map.

Another family of explainability methods computes feature attributions from a forward or backward pass through the network. They require architectural or backpropagation rule modifications or access to intermediate layers. However, most of these methods have lower computa-
tional costs than the ones mentioned above, leading to faster results. One of the first approaches of this kind was introduced by [47], who computed feature attributions by taking the partial derivative of the output class with respect to the input. The resulting absolute values allow identifying which input features can be perturbed the least for the output to change the most. A drawback of this method is that it is noisy, and the absolute value of the gradients prevents the detection of positive and negative evidence in the input. This approach was improved by the Gradient * Input method [46], which increases the sharpness of attribution maps by taking the signed partial derivatives of the output with respect to the input and multiplying feature-wise by the input itself. The multiplication with the input indicates the interest in the salience rather than sensitivity. [46] introduced the Deep Learning Important Features (DeepLIFT) method, which uses a derivative-based method to propagate activation differences instead of gradients through the network. The intuition behind the method is that though the partial derivatives do not explain a single decision, they indicate what change in the image could make a change in the prediction. In the same line, [53] developed the Integrated Gradients approach, which relies on the idea of computing attributions by multiplying the input variable element-wise with the average partial derivative, as the input varies from a baseline to its final value. Smooth-Grad [29] takes a different approach, and focuses on local sensitivity, and calculates averaging maps with a smoothing effect made from several small perturbations of an input image. The effect is enhanced by further training with these noisy images. Finally, it sharpens the sensitivity maps, to increase their quality. [60] was evolved by [53], who proposed the All Convolutional Net, as an alternative that replaces the max-pooling layer for convolutional layers with an increased stride. A slightly different approach was proposed by [61], who introduced the Class Activation Mapping (CAM). This method relies on the observation that some convolutional layers behave as unsupervised object detectors, and it uses global average pooling to create heat maps of a pre-softmax layer. The heat maps point out the regions of an image that are responsible for a prediction. Gradient-weighted Class Activation Mapping (GradCAM) [74] uses the gradient information to understand how strongly does each neuron activate in the last convolutional layer of the neural network. The localizations are combined with existing high-resolution visualizations to obtain high-resolution class-discriminative guided visualizations as saliency masks. The CAM and GradCAM approaches inspired the GradCAM++ method [6], which combines the positive partial derivatives of feature maps of a rear convolutional layer with a weighted special class score to explain the occurrence of multiple object instances in an image. Layer Wise Relevance Propagation (LRP) [3] is a gradient method suffering from vanishing gradient problems. The main idea behind this is the decomposition of the prediction function as a sum of layer-wise relevance values. The prediction is redistributed backward using local redistribution rules until assigning a relevance score to each input feature. There are different variations of the LRP algorithm based on the backward redistribution rule.

Many explainability methods were built, relying on surrogate models to provide explanations regarding the reference model. One of such methods is TREPAN [7] which provides heuristics to issue queries against neural networks and create a decision tree that approximates forecasts from the given network, while providing an interpretable set of rules that explain the forecast. A more general approach was presented in the Local Interpretable Model-agnostic Explanations (LIME) [38], which can explain the predictions of any AI model through a post-hoc, local, linear, and interpretable model. The model attempts to learn a particular forecast, by matching the given feature vector and perturbed inputs, to the results obtained from the reference model. Since the creation of LIME, multiple variants were developed. k-LIME [66] uses local generalized linear model surrogates to explain the predictions, while local regions are defined by k clusters instead of perturbed samples. The criteria to define the value of k is to K that predictions from the local generalized linear models maximize R². In addition to this, a global surrogate linear generalized model is trained to provide information about overall feature average trends. DLIME [68] proposes a deterministic version of LIME, where instead of random perturbations, they apply agglomerative hierarchical clustering to group the training data. The hierarchical clustering does not require prior knowledge regarding
clusters. A dendrogram is cut where the gap is the largest between two successive groups to determine the number of clusters. A k-Nearest Neighbour classifier is trained to classify new instances into those clusters based on the clusters obtained. All data points belonging to a given cluster are used to train a linear model, which provides deterministic and consistent local explanations. LIME-tree ([10]) follows a similar approach to LIME, building a regression tree as surrogate model. The regression tree enables capturing non-linear relationships between the interpretable features and the target variable. At the same time, it does not require independence between interpretable features. The authors consider the model’s biggest advantage is providing personalized counterfactual explanations through an interactive interface that enables imposing certain conditions on the sample of interest. Inspired in LIME, [9] developed STREAK, an interpretability method for neural networks conceived as a set function maximization, achieving similar accuracy than LIME, while having a faster runtime execution. A slightly different approach is presented in Anchors [19], where a set of rules replaces the surrogate model. Since the local behavior of a model can be highly non-linear, the authors propose using a set of if-then rules, which are intuitive and easy to understand. To explore the model’s behavior in the perturbation space, the authors apply multi-armed bandits to incrementally construct the rules, generate candidate predicates, and choose the one with the highest precision until a given precision threshold is reached with a high probability. LoRE - Local Rule-Based Explanations [14] proposes a parameter-free, two step method that also provides rule-based explanations. First, it creates a balanced set of neighbor instances using a genetic algorithm to explore the decision boundary of the data point of interest. Then it builds a decision tree classifier, which enables to derive decision rules and counterfactuals. Local Foil Trees [5] specifically deal with generating counterfactual explanations. To that end, they consider two possible outputs: the model forecast (fact), and the desired label (foil). A decision tree is then built based on the local dataset. The rules are computed from the difference between paths regarding the “fact leaf”, and “foil leaf”.

While most explainability methods based on surrogate models provide specific techniques, [17] developed a framework that enabled comparing surrogate models on three dimensions: data sampling, explanation generation, and interaction. [15] considered a slightly different approach and developed an algorithmic framework (bLIMEy - build LIME yourself) that enables building custom local surrogate explainers for model predictions, considering three dimensions: data sampling, explanation generation, and interpretable representation.

Another local-agnostic explanation method is SHAP [28] which stands for SHapley Additive exPlanations and can be used to produce several explanation models. These models compute SHAP values: a unified measure of feature importance based on the Shapley values, a concept from cooperative game theory. The different explanation models proposed by SHAP differ on how they approximate the computation of the SHAP values. The explanation models provided by SHAP are called additive feature attribution methods. The construction of the 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 image classification field, two explainators can be implemented for deep networks: DEEP-SHAP and GRAD-SHAP. DEEP-SHAP is a high-speed approximation algorithm for shap values in deep learning models that connect with the DeepLift algorithm. The implementation is different from the original DeepLift by using 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. GRAD-SHAP, instead, is based on IntGrad and SmoothGrad algorithms. IntGrad values are a bit different from SHAP values, and require a single reference value to integrate from. As an adaptation to approximate SHAP values, GRAD-SHAP reformulates the integral as an expectation and combines that expectation with sampling reference values from the background dataset as done in SmoothGrad.

Another family of explainability techniques is that of example-based explanations. Methods in this class explain the task model by selecting particular instances from the dataset that describe the model or by creating new instances. Instances that are well predicted by the forecasting model
Explainability is considered a subjective concept. [30] considers that an AI system is explainable if either the model is intrinsically interpretable or if the non-interpretable model can be complemented with an interpretable and faithful explanation. While the XAI techniques provide different kinds of information, the perceived quality of the explanations depends on the users, the domain, the information of interest, and the explanation itself. To evaluate the explanations, it is necessary to define different criteria of goodness for an explanation. Given an interpretable approximation for a reference, model [35] lists four aspects to be considered on evaluation: fidelity (ability to capture the reference model behavior correctly), unambiguity (ability to provide a single and deterministic rationale to explain each data instance), interpretability (the approximation should be human-understandable), and interactivity. The aspect of fidelity is further elaborated by [32], who considers two properties: soundness (the extent to which each explanation component is truthful to the reference model) and completeness (the extent to which the explanation describes the reference model). [36] enumerate another three criteria: sensitivity, the degree of integration, and cognitive

| Explanation technique | Reference | Model based | Attribution based | Example based | Local (L) | Global (G) | Specific (S) | Data Type |
|-----------------------|-----------|-------------|-------------------|--------------|----------|-----------|------------|----------|
| Shapley Values Model | X         | X           |                  | L            |          | A         | ANY        | IMAGE    |
| Class Activation Mapping (CAM) | X         |            |                  |              |          |           | ANY        | IMAGE    |
| Contouring Prediction Difference Analysis | X         |            |                  | L            |          |            | ANY        | IMAGE    |
| Decomposed Network | X         |            |                  | L            |          |            | ANY        | IMAGE    |
| Deep Learning Important Features (DeepLIFT) | X         |            |                  |              |          |           | ANY        | IMAGE    |
| DICE | X         |            |                  |              |          |           | ANY        | ANY      |
| DirectCAM | X         |            |                  |              |          |           | ANY        | ANY      |
| GradCAM ++ | X         |            |                  |              |          |           | ANY        | ANY      |
| GradCAM | X         |            |                  |              |          |           | ANY        | ANY      |
| GradCAM* | X         |            |                  |              |          |           | ANY        | ANY      |
| GradCAM-Input | X         |            |                  |              |          |           | ANY        | ANY      |
| Gradient Weighted Class Activation Mapping (GradCAM) | X         |            |                  |              |          |           | ANY        | ANY      |
| Integrated Gradients | X         |            |                  |              |          |           | ANY        | ANY      |
| k-LIME | X         |            |                  |              |          |           | ANY        | ANY      |
| Localized Weighted Reference Propagation (LWRP) | X         |            |                  |              |          |           | ANY        | ANY      |
| LIME | X         |            |                  |              |          |           | ANY        | ANY      |
| LIME Tree | X         |            |                  |              |          |           | ANY        | ANY      |
| Localized Forward | X         |            |                  |              |          |           | ANY        | ANY      |
| LIME | X         |            |                  |              |          |           | ANY        | ANY      |
| LIME Tree | X         |            |                  |              |          |           | ANY        | ANY      |
| MAPLE | X         |            |                  |              |          |           | ANY        | ANY      |
| Marabini et al. | X         |            |                  |              |          |           | ANY        | ANY      |
| MMD-CRITIC | X         |            |                  |              |          |           | ANY        | ANY      |
| Prediction Difference Analysis (PDA) | X         |            |                  |              |          |           | ANY        | ANY      |
| RISE | X         |            |                  |              |          |           | ANY        | ANY      |
| Shapley Tree | X         |            |                  |              |          |           | ANY        | ANY      |
| SmoothGrad | X         |            |                  |              |          |           | ANY        | ANY      |
| TREPAN | X         |            |                  |              |          |           | ANY        | ANY      |
| XGBoost | X         |            |                  |              |          |           | ANY        | ANY      |

Table 1: Classification of XAI techniques.
salience. Sensitivity is defined as the strength of the relationship of explanatory variables with background conditions: the weaker the relationship, the more convincing the explanation. The degree of integration refers to the connectedness of the explanation to a larger theoretical framework. Finally, cognitive salience is defined as the ease with which the rationale behind the explanation can be followed.

The aforementioned criteria require different evaluation approaches. [8] identified three categories of them:

– **Application-grounded evaluation**: grounded in a real-world application, collects domain expert’s feedback regarding the explanations provided to them.

– **Human-grounded evaluation**: refers to feedback obtained from experiments performed with lay users, when no real-world application exists in place.

– **Functionality-grounded evaluation**: the evaluation is performed considering some formal definition or criteria, that measures the explanation quality.

To assess the explainability methods, [13] propose three tests for functionality-grounded evaluations: **Feature Augmentation Test**, **Synthetic Test**, and **Feature Deduction Test**. The **Feature Augmentation Test** considers that if the values of the explainable features from a specific instance are replaced by the values of those features from an instance with a different label (e.g., “new-label”), the classification outcome should be “new-label”. The **Synthetic Test** is based on the assumption that if the explainability features are accurately selected, new synthetic instances can be created by preserving the explainability feature values and assigning random values to the rest of the features without affecting the forecast outcome. Finally, the **Feature Deduction Test** considers that if the selected explainability features are correctly selected, removing one of them from the input should lead to a different forecast. Even though this approach is frequently adopted in the literature [60,16,35,320], pointed out that samples, where a subset of features are removed have a different data distribution than the samples the model was trained on, violating a key machine learning assumption. They instead propose the RemOve And Retrain (ROAR) approach, which for each feature deemed important, they replace it by a non-informative value in the train and test sets, retrain the model and measure the performance change. In addition to this technique, they propose using a random assignment of feature importance as a benchmark to measure the quality of explainability feature extraction techniques.

There is currently little research regarding application and human-grounded evaluations [8,62]. A popular and domain-specific method is to evaluate to create a heatmap regarding model sensitivity to region-based perturbations. According to the heatmap, the main idea behind this is that the perturbation of relevant input variables would lead to a decline in prediction score than the perturbation of input features with less importance. [22] used questionnaires with short responses and Likert scales. In contrast, [23] used three quantitative metrics: accuracy, response time, and subjective satisfaction. The authors measured accuracy and response time regarding the subject response to different tasks proposed in their research. Subjective satisfaction was measured on a Likert scale for each explanation. [24] proposed the Human Interpretability Score (HIS - see Eq. 1), which constitutes an alternative metric regarding the user’s response time. On the other side, there is a wider set of metrics reported for functionality-grounded evaluations.
\[ HIS(x, R) = \begin{cases} 0, & \text{if } RT_{\text{mean}}(x, R) > RT_{\text{max}} \\ RT_{\text{max}} - RT_{\text{mean}}(x, M), & \text{if } RT_{\text{mean}}(x, R) \leq RT_{\text{max}} \end{cases} \]  

Equation 1: Human Interpretability Score. Measures how long it takes the user to predict the label assigned to certain data point, assigning a cap to the response time. \( x \) and \( R \) correspond to the instance and model considered.

Among the metrics proposed by [55] we find Mutual Information, Diversity, Monotonicity, Non-sensitivity, and Effective complexity. Mutual Information is considered when creating an interpretable data representation. [55] proposes measuring Mutual Information on two cases: (i) between the features of the original model and the subset of explainable features, and (ii) against the target values. Ideally, the number of explainable features should be reduced to maximize simplicity and broadness, while aiming towards keeping a high fidelity regarding the target label (see Eq. 2).

\[ I(x, y) = D_{KL}(P_{(x,y)} \parallel P_x \otimes P_y) \]  

Equation 2: Mutual Information. Measures the mutual dependence between two random variables \( x \) and \( y \).

\textit{Diversity} attempts to measure the degree to which a set of rules integrates to the explanation (see Eq. 3). \textit{Monotonicity} considers that feature attributions should be monotonic. [55] proposes measuring it as the Spearman’s correlation between two vectors: (i) the absolute values of attributions, and (ii) the corresponding expectations. The intuition behind the \textit{Non-sensitivity} metric (see Eq. 4) is to assess that the explainability method does not assign any relevance score to the features the model is not functionally dependent on. The authors compute it as the cardinality of the symmetric difference between features assigned zero attribution and the features the model does not functionally depend on. \textit{Effective complexity} measures if some explanation features can be ignored without significantly affecting the prediction (see Eq. 5).

\[ Diversity = \sum_{x_i, x_j \in E, x_i \neq x_j} \frac{d(x_i, x_j)}{2N_E} \]  

Equation 3: Diversity metric. \( E \) is the set of examples considered, \( d \) is a distance metric for the space \( X \), while \( N_E \) corresponds to the number of examples.

\[ |A_0 \triangle X_0| \]  

Equation 4: Non-sensitivity. \( A_0 \) represents features with zero attribution, \( X_0 \) refers to features on which the model is not functionally dependent on. \(| \cdot |\) denotes the set cardinality, and \( \triangle \) the symmetric set difference.
\[ k^* = \arg\min_{k \in \{1, \ldots, N\}} |M_k| \text{ where } E(l(y^*, f - M_k)|x^*_{M_k}) < \varepsilon \] (5)

Equation 5: Effective Complexity. \( M_k \) denotes the set of top \( k \) features, \( x \) denotes features, \( \varepsilon > 0 \) corresponds to some arbitrary tolerance, \( f - M_k \) is the restriction of the model \( R \) to non-important features, given \( M_k \).

The **Local Approximation Accuracy** was proposed by [13] to compare the decision boundary of the surrogate model against the original one. The authors do so by computing the Root Mean Squared Error between the original and surrogate model predictions on the test samples. A similar intuition is present in the **Disagreement** metric proposed by [13]. For a classification setting, they attempt to measure the surrogate model fidelity by computing the disagreement between labels of the surrogate model and the original one (see Eq. 6).

\[
\text{Disagreement}(R) = \sum_{i=1}^{N} |x: x \in D, x satisfies q_i \land s_i, B(x) \neq c_i|
\] (6)

Equation 6: Disagreement metric. Quantifies the disagreement between a surrogate model \( R \) and the reference forecasting model \( B \), given a dataset \( D \). The triplet \((q, s, c)\) stands for (feature, operator, class).

[13] propose another six metrics to evaluate forecast explanations: rule overlap, cover, the rule set size (see Eq. 7), the rule set maximum width, the number of descriptor sets, and feature overlap. The **Rule overlap** computes the overlap between pairs of rules defined in the surrogate model. It is expected that the lower the overlap, the lower the surrogate model ambiguity (see Eq. 8). **Cover** is defined as the number of instances that match a given rule from the surrogate model (see Eq. 9). The **Maximum Width** refers to the maximum width obtained from computing the width over all the elements from the surrogate model. The authors define an element as either rule conditions or neighborhood descriptors (see Eq. 10). The authors define the **Number of Unique Descriptor Sets** as the number of unique neighborhood descriptors provided in the surrogate model (see Eq. 11). Finally, the **Feature overlap** measures the features overlap between every pair of unique neighborhood descriptor and rule (see Eq. 12).

\[
\text{RuleSetSize}(R) = \text{NumberOfRules}(q, s, c)
\] (7)

Equation 7: Rule set size. \( R \) denotes the decision set. The triplet \((q, s, c)\) stands for (feature, operator, class). The triplets are contained in the decision set.
Equation 8: Rule overlap. $R$ denotes the decision set. The triplet $(q, s, c)$ stands for (feature, operator, value).

$$RuleOverlap(R) = \sum_{i=1}^{N} \sum_{j=i, j \neq i}^{N} overlap(q_i \land s_i, q_j \land s_j)$$

Equation 9: Cover. $R$ denotes the decision set. The triplet $(q, s, c)$ stands for (feature, operator, value). $D$ represents a dataset, and $x$ and instance in such dataset.

$$cover(R) = \left| \{ x \in D, x \text{satisfies } q_i \land s_i, \text{ where } i \in 1...N \} \right|$$

Equation 10: Maximum Width. $R$ denotes the decision set. $e$ represents elements, which can be either rule conditions or neighborhood descriptors.

$$MaximumWidth(R) = \max(\text{width}(e), e \in \bigcup_{i=1}^{N} (q_i \cup s_i))$$

Equation 11: Number of Unique Descriptor Sets. $R$ denotes the decision set, and $q$ denotes features.

$$NumberOfUniqueDescriptorSets(R) = |dset(R)|, \text{ where } dset(R) = \bigcup_{i=1}^{N} (q_i)$$

Equation 12: Feature Overlap. $R$ denotes the decision set, $q$ denotes features in descriptor sets, and $s$ denotes operators.

$$FeatureOverlap(R) = \sum_{i=1}^{N} FeatureOverlap(q, s_i)$$

A different set of metrics is considered by [37], who for tree-based models measured the mean path length, the mean number of distinct features in a path, the number of nodes, and the number of nonzero features. Finally, [48] reported assessing explainability methods based on the total number of runtime operation counts performed by the model when computing the forecast for a given input.
4 Applications, Use Cases and Open Issues

Though multiple XAI methods exist, they do not suffice by themselves to provide human-understandable explanations. They are built into frameworks and applications that provide a convenient interface and additional context to achieve that goal. One such framework is bLIMEy[33], which decomposes surrogate models into three steps: interpretable data representation (transform data from the original to the interpretable domain), data sampling, and explanation generation. [34] follows a similar approach and describes the IBEX (Interactive Black-box EXplanation system) framework with two components: an explainer that produces explanations based on user’s needs, and a sampling component, that selects appropriate inputs to create the explanation. [32] describes AI Explainability 360, an extensible toolkit developed that provides contextual explainers based on the stage of the AI model development pipeline, kind of model, and explanation requirements. [33] explores the usage of semantic technologies to abstract relevant concepts encoded in the features, avoid exposing sensitive details regarding the forecasting model, and provide higher-level information to the users. The authors complement model explanations with information regarding real-world events reported in the media that likely influenced the variables of interest. [35] developed an ontology to model user’s feedback based on a given forecast and provided explanations. [37] developed an intelligent assistant for manufacturing, which creates directive explanations for the users using heuristics and domain knowledge. The application tracks user’s implicit and explicit feedback regarding local forecast explanations, enabling application-grounded evaluations.

The integration of explainability methods into applications enables providing relevant information regarding model forecasts to different stakeholders. For instance, data scientists and machine learning engineers require low-level data to monitor the AI model behavior, identify corner cases, and work towards a more accurate and robust model. On the other side, employees and supervisors require high-level insights that convey reasons behind the model forecasts, can interactively explore different “what-if” scenarios, and provide feedback regarding the explanations provided. We envision explainability methods can be useful in a wide range of manufacturing use cases, such as automatic defect detection (inform the user on the image regions influencing the decision), production planning (provide an insight on the cost of the opportunity given different scheduling decisions), or demand forecasting (provide insights why we expect demand will take place and which factors affect the quantity estimates).

Several explainability techniques have been implemented in the manufacturing domain and specifically the predictive quality management domain (Quality 4.0) to boost the transparency of AI deployed models. [33] used XAI techniques such as CAM and Contrastive gradient-based saliency maps to explain black-box classifiers in the area of quality welds in ultrasonically welded battery tabs. They produced heatmaps where they visualized several color maps to gain insights into true positive versus false-positive predictions. [36] implemented several XAI methods to provide explanations for domain experts in the area of 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 as they produced well-distributed heatmaps. Moreover, by fitting the model into a decision tree and converting the prediction results into human interpretable text, the authors achieved the maximum level of explainability when they presented the results to domain experts for evaluation purposes. In the area of manufacturing cost estimation, [37] 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 XAI method was applied.

Cybersecurity in a transversal concern related to all smart manufacturing cases. XAI techniques were successfully applied in the cybersecurity domain, to support the exploration of model vulnerabilities [16][19], and identify perturbed data samples [10].
In the European Horizon 2020 project STAR (Safe and Trusted Human Centric Artificial Intelligence in Future Manufacturing Lines), XAI is used to provide insights on most relevant features to each forecast, explore model vulnerabilities and help identify potential data poisoning. While providing accurate explanations to forecasts provides the users additional elements for decision-making, the vulnerabilities assessment and early data poisoning identification ensures the system is secure, enhancing users trust in the system.

5 Conclusion

The new industrial revolution relies on AI to enable higher production efficiency, and safer operations. XAI techniques provide means to reduce black-box models opaqueness, and increase trust in the system. In this contribution, we introduce the field of XAI. We list several taxonomies found in the literature alongside state-of-the-art methods and techniques to interpret AI models. We also include metrics with different qualitative and quantitative characteristics as a means of evaluating the above methods. Finally, we list applications of XAI, describe several use cases in the manufacturing domain, and open opportunities.

XAI requires a multi-disciplinary approach. Special consideration needs to be given to understand how domain experts and end-users operate. Users must be involved in the XAI outcomes validation. The integration of XAI into manufacturing processes will be paramount for the transition into the fifth industrial revolution.

Acknowledgements

This work has been carried out in the H2020 STAR project, which has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 956573.

References

1. Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamin, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

2. Vijay Arya, Rachel KE Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C Hoffman, Stephanie Houde, Q Vera Liao, Ronny Luss, Aleksandra Mojsilovic, et al. Ai explainability 360: An extensible toolkit for understanding data and machine learning models. *Journal of Machine Learning Research*, 21(130):1–6, 2020.

3. Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

4. Francesco Bodria, Fosca Giannotti, Riccardo Guidotti, Francesca Naretto, Dino Pedreschi, and Salvatore Rinzivillo. Benchmarking and survey of explanation methods for black box models. *arXiv preprint arXiv:2102.13076*, 2021.

5. Lok Chan. Explainable ai as epistemic representation. *Overcoming Opacity in Machine Learning*, page 7, 2021.
6. Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 839–847. IEEE, 2018.

7. Mark Craven and Jude Shavlik. Extracting tree-structured representations of trained networks. Advances in neural information processing systems, 8:24–30, 1995.

8. Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1703.06467, 2017.

9. Ethan R Elenberg, Alexandros G Dimakis, Moran Feldman, and Amin Karbasi. Streaming weak submodularity: Interpreting neural networks on the fly. arXiv preprint arXiv:1703.06467, 2017.

10. Gil Fidel, Ron Bitton, and Asaf Shabtai. When explainability meets adversarial learning: Detecting adversarial examples using shap signatures. In 2020 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE, 2020.

11. Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In Proceedings of the IEEE International Conference on Computer Vision, pages 3429–3437, 2017.

12. Claudia V Goldman, Michael Baltaxe, Debejyo Chakraborty, and Jorge Arinez. Explaining learning models in manufacturing processes. Procedia Computer Science, 180:259–268, 2021.

13. Jindong Gu and Volker Tresp. Contextual prediction difference analysis for explaining individual image classifications. arXiv preprint arXiv:1910.09086, 2019.

14. 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.

15. Wenbo Guo, Dongliang Mu, Jun Xu, Purui Su, Gang Wang, and Xinyu Xing. Lemna: Explaining deep learning based security applications. In proceedings of the 2018 ACM SIGSAC conference on computer and communications security, pages 364–379, 2018.

16. Patrick Hall, Navdeep Gill, Megan Kurka, and Wen Phan. Machine learning interpretability with h2o driverless ai. H2O ai. URL: http://docs.h2o.ai/driverless-ai/latest-stable/docs/booklets/MLIBooklet.pdf, 2017.

17. Clément Henin and Daniel Le Métayer. Towards a generic framework for black-box explanations of algorithmic decision systems. In IJCAI 2019 Workshop on Explainable Artificial Intelligence (XAI), 2019.

18. Clément Henin and Daniel Le Métayer. A generic framework for black-box explanations. In 2020 IEEE International Conference on Big Data (Big Data), pages 3667–3676. IEEE, 2020.

19. Clément Henin and Daniel Le Métayer. A multi-layered approach for tailored black-box explanations. 2021.

20. Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. arXiv preprint arXiv:1806.10758, 2018.

21. Been Kim, Oluwasanmi Koyejo, Rajiv Khanna, et al. Examples are not enough, learn to criticize! criticism for interpretability. In NIPS, pages 2280–2288, 2016.

22. Todd Kulesza, Simone Stumpf, Margaret Burnett, Sherry Yang, Irwin Kwan, and Weng-Keen Wong. Too much, too little, or just right? ways explanations impact end users’ mental models. In 2013 IEEE Symposium on visual languages and human centric computing, pages 3–10. IEEE, 2013.
23. Isaac Lage, Emily Chen, Jeffrey He, Menaka Narayanan, Been Kim, Sam Gershman, and Finale Doshi-Velez. An evaluation of the human-interpretability of explanation. arXiv preprint arXiv:1909.00006, 2019.
24. Isaac Lage, Andrew Slavin Ross, Been Kim, Samuel J Gershman, and Finale Doshi-Velez. Human-in-the-loop interpretability prior. arXiv preprint arXiv:1805.11571, 2018.
25. Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Interpretable & exploratory approximations of black box models. arXiv preprint arXiv:1707.01154, 2017.
26. Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. Nature communications, 10(1):1–8, 2019.
27. Minyoung Lee, Joohyoung Jeon, and Hongchul Lee. Explainable ai for domain experts: a post hoc analysis of deep learning for defect classification of tft–lcd panels. Journal of Intelligent Manufacturing, pages 1–13, 2021.
28. Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. arXiv preprint arXiv:1705.07874, 2017.
29. Yuxin Ma, Tiankai Xie, Jundong Li, and Ross Maciejewski. Explaining vulnerabilities to adversarial machine learning through visual analytics. IEEE transactions on visualization and computer graphics, 26(1):1075–1085, 2019.
30. Aniek F Markus, Jan A Kors, and Peter R Rijnbeek. The role of explainability in creating trustworthy artificial intelligence for health care: a comprehensive survey of the terminology, design choices, and evaluation strategies. Journal of Biomedical Informatics, page 103655, 2020.
31. Ramaravind K Morhilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pages 607–617, 2020.
32. Vincent C Müller. Deep opacity undermines data protection and explainable artificial intelligence. Overcoming Opacity in Machine Learning, page 18, 2021.
33. An-phi Nguyen and Maria Rodríguez Martínez. On quantitative aspects of model interpretability. arXiv preprint arXiv:2007.07584, 2020.
34. Cecilia Paniguttì, Alan Perotti, and Dino Pedreschi. Doctor xai: an ontology-based approach to black-box sequential data classification explanations. In Proceedings of the 2020 conference on fairness, accountability, and transparency, pages 629–639, 2020.
35. Vitali Petsiuk, Abir Das, and Kate Saenko. Risc: Randomized input sampling for explanation of black-box models. arXiv preprint arXiv:1806.07421, 2018.
36. Gregory Plumb, Denali Molitor, and Ameet Talwalkar. Model agnostic supervised local explanations. arXiv preprint arXiv:1807.02910, 2018.
37. Forough Poursabzi-Sangdeh, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Vaughan, and Hanna Wallach. Manipulating and measuring model interpretability. arXiv preprint arXiv:1802.07810, 2018.
38. 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.
39. Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018.
40. Marko Robnik-Šikonja and Igor Kononenko. Explaining classifications for individual instances. *IEEE Transactions on Knowledge and Data Engineering*, 20(5):589–600, 2008.

41. Jože M. Rožanec, Patrik Zajec, Klemen Kenda, Inna Novalija, Blaž Fortuna, and Dunja Mladenčič. Xai-kg: knowledge graph to support xai and decision-making in manufacturing, 2021.

42. Jože M Rožanec and Dunja Mladenčič. Semantic xai for contextualized demand forecasting explanations. *arXiv preprint arXiv:2004.00432*, 2021.

43. Wojciech Samek and Klaus-Robert Müller. Towards explainable artificial intelligence. In *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 5–22. Springer, 2019.

44. A Carlisle Scott, William J Clancey, Randall Davis, and Edward H Shortliffe. Explanation capabilities of production-based consultation systems. Technical report, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1977.

45. 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 *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

46. Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.07113*, 2016.

47. Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

48. Dylan Slack, Sorelle A Friedler, Carlos Scheidegger, and Chitradeep Dutta Roy. Assessing the local interpretability of machine learning models. *arXiv preprint arXiv:1902.03501*, 2019.

49. Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.01982*, 2017.

50. Kacper Sokol and Peter Flach. Limetree: Interactively customisable explanations based on local surrogate multi-output regression trees. *arXiv preprint arXiv:2005.00427*, 2020.

51. Kacper Sokol, Alexander Hepburn, Raul Santos-Rodriguez, and Peter Flach. blimey: surrogate prediction explanations beyond lime. *arXiv preprint arXiv:1910.13016*, 2019.

52. Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

53. Mukund Sundararajan, Ankur Taly, and Qicai Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

54. Jasper van der Waa, Marcel Robeer, Jurriaan van Diggelen, Matthieu Brinkhuis, and Mark Neerincx. Contrastive explanations with local foil trees. *arXiv preprint arXiv:1806.07470*, 2018.

55. Feiyu Xu, Hans Uszkoreit, Yangzhou Du, Wei Fan, Dongyan Zhao, and Jun Zhu. Explainable ai: A brief survey on history, research areas, approaches and challenges. In *CCF international conference on natural language processing and Chinese computing*, pages 563–574. Springer, 2019.

56. Petri Ylioski and Jaakko Kuorikoski. Dissecting explanatory power. *Philosophical studies*, 148(2):201–219, 2010.
57. Soyoung Yoo and Namwoo Kang. Explainable artificial intelligence for manufacturing cost estimation and machining feature visualization. *arXiv preprint arXiv:2010.14824*, 2020.

58. Muhammad Rehman Zafar and Naimul Mefraz Khan. Dlime: A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems. *arXiv preprint arXiv:1906.10263*, 2019.

59. Patrik Zajec, Jože M Rožanec, Inna Novalija, Blaž Fortuna, Dunja Mladenić, and Klemen Kenda. Towards active learning based smart assistant for manufacturing. *arXiv preprint arXiv:2105.16777*, 2021.

60. Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

61. 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.

62. Jianlong Zhou, Amir H Gandomi, Fang Chen, and AndreasHolzinger. Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics*, 10(3):593, 2021.

63. 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.
Glossary

AI  Artificial Intelligence. 1, 2, 4, 6, 11, 12
CAM  Class Activation Mapping. 4, 6, 11
GradCAM  Gradient-weighted Class Activation Mapping. 4, 6
LIME  Local Interpretable Model-agnostic Explanations. 4, 5
LRP  Layer Wise Relevance Propagation. 4, 6, 11
ReLU  Rectified Linear Unit. 3
ROAR  RemOve And Retrain. 7
SHAP  SHapley Additive exPlanations. 5
XAI  Explainable Artificial Intelligence. 1–3, 6, 11, 12