COMPUTER
METHODS
IN MATERIALS
SCIENCE

2025, vol. 25, no. 1, 31–43
https://doi.org/10.7494/cmms.2025.1.1013

Review of XAI methods for application in heavy industry

Wojciech Jędrysik*, Piotr Hajder, Łukasz Rauch

AGH University of Krakow, Department of Applied Computer Science and Modelling, Krakow, Poland.

Abstract

In recent years, considerable progress has been made in the field of artificial intelligence and machine learning. This progress allows us to solve increasingly complex problems, but it also requires providing appropriate explanations to understand the actions taken by AI. For this purpose, research into the development of Explainable Artificial Intelligence has been initiated and interest in this topic is constantly growing. This review of XAI methods includes a justification for the need to introduce solutions to explain artificial intelligence models, describes the differences between various methods and presents example methods that work in different cases. The purpose of this paper is to solve a real problem occurring in heavy industry. The third chapter describes the challenges to be faced, the solution developed and the results of the work. The entire study concludes with a summary of the research findings.

Keywords: explainable artificial intelligence, machine learning, heavy industry

1. Introduction

In recent years, there has been massive progress in the field of machine learning and artificial intelligence. With the increasing use of artificial intelligence, there is also a growing need to understand the decisions it makes in order to gain more confidence in what it does and to obtain a better understanding of its mechanics. Machine learning models such as deep neural networks are difficult to interpret because they rely on complex mathematical calculations which is why they are often called "black boxes", when the input is given, their predictions are returned and the user does not even know on what basis the obtained result was calculated. Such a situation limits a human's understanding of why a given model makes specific suggestions or decisions.

The relationship between the effectiveness (complexity) of an ML model and its interpretability is shown in Figure 1 (Ciatto et al., 2020).

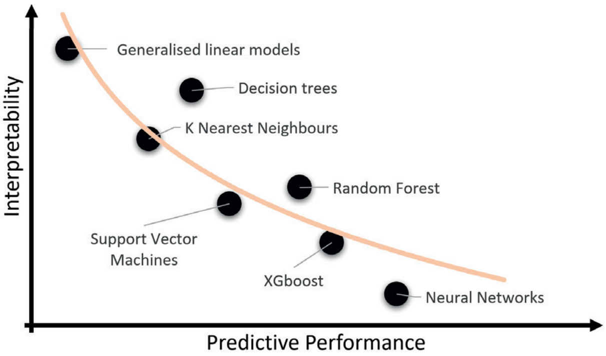
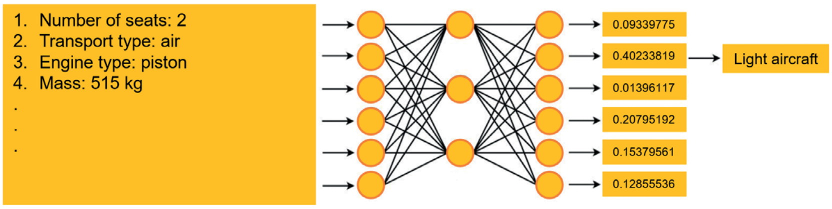


Fig. 1. Relationship between the effectiveness (complexity) of an ML model and its interpretability (developed with reference to the research performed by Ciatto et al. (2020))

Therefore, recently there has been an increasing interest in XAI (Explainable Artificial Intelligence), i.e. AI (Artificial Intelligence) capable of explaining its decisions in a way that is understandable to humans. An example showing the difference between the traditional AI model and the XAI model is shown in Figure 2.

* Corresponding author: wjedrysik@agh.edu.pl

ORCID ID's: 0009-0008-1454-3766 (W. Jędrysik), 0000-0002-0437-3328 (P. Hajder), 0000-0001-5366-743X (Ł. Rauch)



AI	XAI
Prediction: Light aircraft	Prediction: Light aircraft
Why? The value of the neuron in the final classification layer, which is responsible for the „Light aircraft" category, was the highest (0.40233819).	Why? Because: •Light aircrafts mainly accommodate 2 to 4 people. •A light aircraft is a type of airplane. •Light aircrafts are powered by piston engines. •The empty weight of a light aircraft usually does not exceed 1000 kg.

Fig. 2. Example showing the difference between the traditional AI model and the XAI model

1.1. Explainability vs interpretability

The explainability of AI models involves the possibility of using appropriate algorithms designed to give users insight into the mechanisms that artificial intelligence uses to obtain a result. Such techniques make it possible to explain AI decisions in a way that is understandable to humans. Interpretability, on the other hand, tells us to what extent the AI model itself is readable to humans, i.e. to what extent the user is able to understand how the model works, the criteria and principles influencing the result, by looking at the structure of this model. We can also talk about interpretability in the context of combining an AI model and appropriate XAI techniques to assess how understandable and sufficient the generated explanations are for the target audience.

What is the motivation for explaining AI models' decisions? Some sample aims are listed below:

– acceptance from contemporary society;
– better results of human cooperation with AI;
– ensuring safety in critical areas such as medicine (diagnoses) or automotive (autonomous cars);
– legal regulations;
– debugging AI models;
– human curiosity.

The development of AI can also bring many benefits to heavy industry because the appropriate optimization of processes or increasing efficiency can significantly reduce the production costs of various products. Moreover, heavy industry is increasingly using advanced AI systems to monitor, control or optimize complex processes, but in the case of malfunction, it is difficult to identify the causes and understand why this happened.

In this paper, we will look at different methods for explaining AI models, starting with the presentation of other papers describing selected techniques, then some of them will be described in more detail with examples of their application. The next step will be to present an actual problem related to heavy industry, then the data on which it was based will be described, and the solution to the task and the results will be presented. In conclusion, it will be demonstrated that XAI represents the next stage in the evolution of artificial intelligence models, encompassing a diverse range of methodologies and proving its suitability for applications in heavy industry.

2. XAI methods

There are many XAI methods available, and this number is dictated by the variety of problems and AI models used to solve them. There is no single method that can explain the decisions made by each possible model in different ways, so new techniques continue to be developed to fill the gaps in this ever-evolving area.

2.1. Differences between XAI methods

XAI methods differ from each other in many aspects and can be divided in several ways. This division, along with examples, is presented in Table 1 (Sofianidis et al., 2021; Vilone & Longo, 2020).

Table 1. Division of XAI methods (Sofianidis et al., 2021; Vilone & Longo, 2020) along with three examples of each method type

XAI method type	Examples
Type of input data	
Numerical/categorical	LIMETree (Sokol & Flach, 2024), Local Foil Trees (Waa et al., 2018), LoRE (Guidotti et al., 2018)
Pictorial	All Convolutional Net (Springenberg et al., 2014), CAM (Zhou et al., 2016), GradCAM (Selvaraju et al., 2020)
Textual	Integrated Gradients (Sundararajan et al., 2017), k-LIME (Hall et al., 2024), Layer Wise Relevance Propagation (Bach et al., 2015)
Time series	DeepLIFT (Shrikumar et al., 2017), DICE (Mothilal et al., 2020), DLIME (Zafar & Khan, 2019)
Way of presenting explanations (output format)	
Numerical	Gradient (Simonyan et al., 2013), Gradient*Input (Shrikumar et al., 2017), MAPLE (Plumb et al., 2019)
Rules	LIMETree (Sokol & Flach, 2024), Local Foil Trees (Waa et al., 2018), LoRE (Guidotti et al., 2018)
Textual	Anchors (Ribeiro et al., 2018), LIME (Ribeiro et al., 2016), SHAP (Lundberg & Lee, 2018)
Visual	Deconvolutional Networks (Zeiler & Fergus, 2013), GradCAM++ (Chattopadhay et al., 2018), RISE (Petsiuk et al., 2018)
Mixed	Gradient (Simonyan et al., 2013), Gradient*Input (Shrikumar et al., 2017), Layer Wise Relevance Propagation (Bach et al., 2015)
Stage where explanations are generated in relation to the AI modeling process	
Ante-hoc	Decision Trees (Thombre, 2024), RuleFit (Friedman & Popescu, 2008), Supersparse Linear Integer Model (Ustun et al., 2013)
Post-hoc	Anchors (Ribeiro et al., 2018), Meaningfull Perturbation (Fong & Vedaldi, 2017), GradCAM (Selvaraju et al., 2020)
Dependence on the AI model architecture (in case of *post-hoc* methods)	
Model-agnostic	k-LIME (Hall et al., 2024), LoRE (Guidotti et al., 2018), STREAK (Elenberg et al., 2018)
Model-specific	CAM (Zhou et al., 2016), Smooth Grad (Smilkov et al., 2017), TREPAN (Craven & Shavlik, 1996)
Scope of explanations	
Global	MMD-critic (Kim et al., 2017), SHAP (Lundberg & Lee, 2018), TREPAN (Craven & Shavlik, 1996)
Local	Meaningfull Perturbation (Fong & Vedaldi, 2017), Prediction Difference Analysis (Zintgraf et al., 2017), Smooth Grad (Smilkov et al., 2017)
Problem type	
Classification	Contextual Prediction Difference Analysis (Gu & Tresp, 2019), Meaningfull Perturbation (Fong & Vedaldi, 2017), Smooth Grad (Smilkov et al., 2017)
Regression	Gradient (Simonyan et al., 2013), Layer Wise Relevance Propagation (Bach et al., 2015), MAPLE (Plumb et al., 2019)

In the next subsections, two selected methods are described in detail: GradCAM and Anchors. The first one was chosen because it was used to solve the problem in the heavy industry described in the next chapter, while Anchors is a method that works in a completely different way than GradCAM, which shows how diverse XAI methods can be in their approach to the problem and the way of working and presenting explanations. The motivation was the desire to describe two very different methods, therefore on the one hand there is a method focused on a specific type of problem, in this case, a method used to explain the decisions of convolutional neural networks for image analysis, and, on the other hand, there is a more general method that can be used for different types of problems and AI models. Other methods that are popular and can be used in many cases are also briefly described.

2.2. GradCAM

GradCAM (Gradient-weighted Class Activation Mapping) is an XAI method that can be classified according to the above criteria as one that uses visualization, showing the importance of individual features in the local space, and we define it as model-specific due to the need to use an appropriate AI model for classification tasks and architecture based on convolutional layers (Selvaraju et al., 2020).

It is used to make explanations on images by generating and applying appropriate heat maps to them, which are intended to indicate those areas in the image that had the greatest impact on the obtained prediction result, which helps to understand the key features of the objects presented in the image.

To use the GradCAM method, we need a previously trained classification model, and the entire operation of the algorithm can be presented in several steps (Selvaraju et al., 2020):

1. We pass a specific image through a previously trained model and obtain the prediction result.
2. After receiving the results, loss gradients are calculated in relation to the feature maps calculated in the last convolutional layer, which indicate how changes in pixel values in the feature maps affect the change in the prediction result for the resulting class (class selected by the model).
3. Global Average Pooling is performed to calculate the weighted average of the gradient, thus determining the weights for all feature maps for a given class.
4. The feature maps are multiplied by their corresponding calculated weights. In this way, maps that are more relevant to predicting a given class have a greater impact on the resulting visualization.
5. The resulting weighted feature maps are summed along the channel/feature axes to form one activation map for the predicted class.
6. The ReLU activation function is applied to the resulting map to remove negative values.
7. The result of all previous operations is a heat map that is superimposed on the input image, thus showing which areas of the image contributed most to classifying the image into a specific class.

A diagram presenting the operation of the described algorithm along with a shortened explanation in several steps is shown in Figure 3 (Selvaraju et al., 2020).

When receiving results from the model in the form of raw data, the Softmax function is used to normalize the set of real numbers in such a way that they can be interpreted as probabilities:

$$\sigma(\vec{\mathbf{z}})_y = \frac{e^{z_y}}{\sum_{k=1}^{n_c} e^{z_k}} \tag{1}$$

where:

$\sigma(\vec{\mathbf{z}})_y$ – the result of the Softmax function for the value included in the vector $\vec{\mathbf{z}}$ corresponding to class y,

e^{z_y} – the exponent of the z_y value, i.e. the z value (score) for class y,

$\sum_{k=1}^{n_c} e^{z_k}$ – the sum of the exponential values of all input elements for the Softmax function (all components of the vector $\vec{\mathbf{z}}$ corresponding to all classes taken into account).

GradCAM computes the prediction gradient relative to the last convolutional layer for each feature x for class y:

$$\frac{\partial z_y}{\partial A^x_{i,j}} \tag{2}$$

where:

z_y – the final raw value (score) for class y before Softmax,

$A^x_{i,j}$ – the activation value at position (i, j) on the feature map for channel (feature) x.

The whole is a gradient of the influence of activation on the result for a given class. After calculating the above formula for all points (i, j), a matrix A' is created for feature x and class y, so in total we can have $n_f \cdot n_c$ A' matrices.

Each such matrix A' is then subjected to Global Average Pooling to obtain a weight for each feature for class y. This value tells how much a given feature influences the model's decision:

$$\alpha^y_x = \frac{1}{H \cdot W} \cdot \sum_{i=1}^{H} \sum_{j=1}^{W} A'^{x,y}_{i,j} \tag{3}$$

where:

α^y_x – a weight determining the impact of the feature map A^x on the result for class y,

H, W – the spatial dimensions of the feature map,

$A'^{x,y}_{i,j}$ – the value of the matrix A' calculated using the gradient for feature x and class y at point (i, j).

Finally, each calculated matrix A' is multiplied by its weight α:

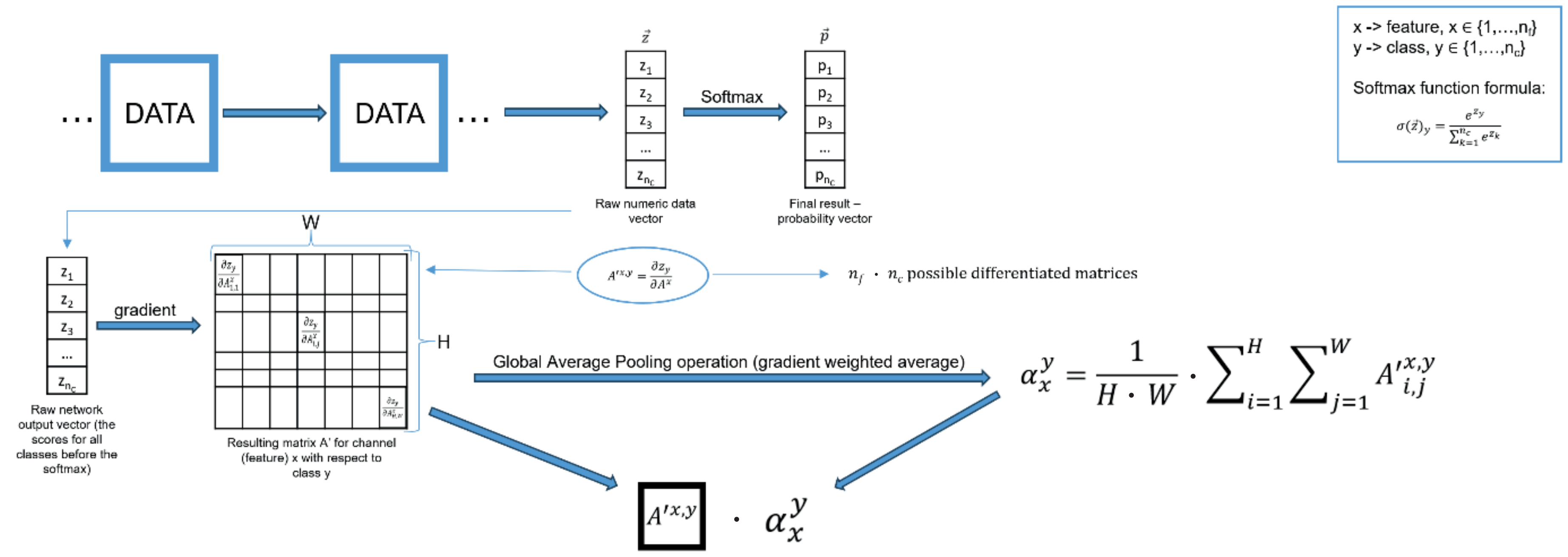
$$A'^{x,y} \cdot \alpha^y_x \tag{4}$$

In this way, specific features become more important for a given class. All weighted feature maps are summed and passed to the input of the ReLU function, which removes negative values that do not affect the visualization:

$$ReLU\left(\sum_{x=1}^{n_f} \alpha^y_x A'^{x,y}\right) \tag{5}$$

where n_f is the number of all features.

In this way, a heat map is calculated which, when applied to the input image, shows the key regions of the image that have the greatest impact on the decision made by the model.


…
DATA
DATA
…
$\vec{z}$
z_1
z_2
z_3
…
z_{n_c}
Raw numeric data vector
Softmax
$\vec{p}$
p_1
p_2
p_3
…
p_{n_c}
Final result – probability vector
x -> feature, x ∈ {1,…,n$_f$}
y -> class, y ∈ {1,…,n$_c$}
Softmax function formula:
$\sigma(\vec{z})_y = \frac{e^{z_y}}{\sum_{k=1}^{n_c} e^{z_k}}$
z_1
z_2
z_3
…
z_{n_c}
Raw network output vector (the scores for all classes before the softmax)
gradient
W
H
$\frac{\partial z_y}{\partial A_{1,1}^x}$
$\frac{\partial z_y}{\partial A_{i,j}^x}$
$\frac{\partial z_y}{\partial A_{H,W}^x}$
Resulting matrix A' for channel (feature) x with respect to class y
$A'^{x,y} = \frac{\partial z_y}{\partial A^x}$
$n_f \cdot n_c$ possible differentiated matrices
Global Average Pooling operation (gradient weighted average)
$\alpha_x^y = \frac{1}{H \cdot W} \cdot \sum_{i=1}^{H} \sum_{j=1}^{W} A'^{x,y}_{i,j}$
$\boxed{A'^{x,y}} \cdot \alpha_x^y$

How the GradCAM method works:

1. We perform the above operations for each feature (equivalents of x).
2. We pass the calculated sum through the ReLU activation function.
3. The obtained matrix is transformed into a heat map.
4. Heat map is applied to the input image, creating an explanation of the decision made (prediction).

Fig. 3. GradCAM algorithm operation (developed with reference to the research performed by Selvaraju et al. (2020)). The upper part of the image shows a simplified diagram of data flow in a convolutional neural network, the middle part shows the operation diagram of the GradCAM algorithm, and the steps described at the bottom of the image complete this diagram

2.3. GradCAM in other papers

GradCAM is used in medicine because the heat map it generates, when superimposed on an image, e.g. a photo of a given organ, helps to identify disease lesions in the human body, which facilitates the work of doctors when making a diagnosis and thus accelerates the detection of abnormalities, which affects higher chances of recovery among patients.

The paper written by Gulum et al. (2021) discusses the problem of cancer detection in medical images. The developed deep learning model copes well with the above-mentioned task, but in addition, a solution is needed that will present an explanation of the obtained results in a human-comprehensible way to increase the transparency of the so-called "black boxes" in order to increase trust among patients and doctors. The article distinguishes several categories of explanation methods depending on the aspects from which we look at the mechanisms of XAI methods. The authors then assessed the quality of the explanations received. Such an analysis provided them with the necessary data to decide what techniques to use in specific cases and, for example, to locate tumor lesions in the brain or diagnose breast cancer, GradCAM was used as an XAI method, which is effective in marking key places in photos. Thanks to such auxiliary materials, it is easier for physicians to make a diagnosis quickly and effectively, and to explain the problem and its causes to patients.

2.4. Anchors

The Anchors method is classified as a method presenting explanations in text form, showing key features that influence the prediction result, it can operate locally and globally, and is model-agnostic (Ribeiro et al., 2018).

Anchors is a method that explains the model's predictions in the form of the so-called "anchors". Anchors are rules built from conditions, so they describe what conditions must be met for the model to make a specific decision. This method can be used on data types such as text or tabular data.

There are two algorithms for generating anchors (Ribeiro et al., 2018):

1. Identifying the Best Candidate for Greedy,
2. Outline of the Beam Search.

To use the Anchors technique, we need a previously trained AI model, and the entire operation of the algorithm (Identifying the Best Candidate for Greedy) consists of the following steps (Ribeiro et al., 2018):

1. We pass a specific instance (input data) through a previously trained model for which we obtain a prediction.
2. The entire process of creating anchors is a loop in which we go through the stage of generating candidates and selecting the best one during each iteration:
   a) Candidate generation: involves creating combinations of features and their values for a selected case. Each candidate is built from the so-called predicates (conditions to be met). At this stage, we iterate through the available predicates (not yet included in the current final anchor) and check whether the anchor from the previous iteration (in the case of the first iteration, we start from the empty set) in combination with the next predicate meets the specified coverage condition. If the condition is met, the new candidate moves to the next stage (selecting the best candidate), otherwise the candidate is rejected.
   b) Selecting the best candidate: In this step, the precision of each candidate is calculated and then the candidate with the best score is selected. Accurate calculation of the precision of each candidate is very time-consuming and resource-intensive, therefore, in order to correctly find the candidate with the highest precision while reducing computational costs, the so-called "multi-armed bandit problem" is used, which is based on probability distributions and estimation calculations. The candidate with the highest precision, if it meets the minimum precision condition, moves on, otherwise it is rejected.
3. If we managed to find the candidate with the highest precision with a value that meets the previously assumed minimum precision condition, we set it as the current final anchor and move with it to the next iteration.
4. The entire loop ends when the current iteration does not provide any candidate that meets all the requirements described above (the iteration returns an empty set). The result of the algorithm and the final anchor is the anchor from the previous iteration. If no candidate was returned in the first iteration, it means that the algorithm did not find any anchor, i.e. solution (the described XAI method is not able to explain the result of the prediction made by the AI model).

A diagram presenting the operation of the described algorithm is shown in Figure 4 (Ribeiro et al., 2018). The comparison of precision and coverage is shown in Table 2 (Ribeiro et al., 2018).

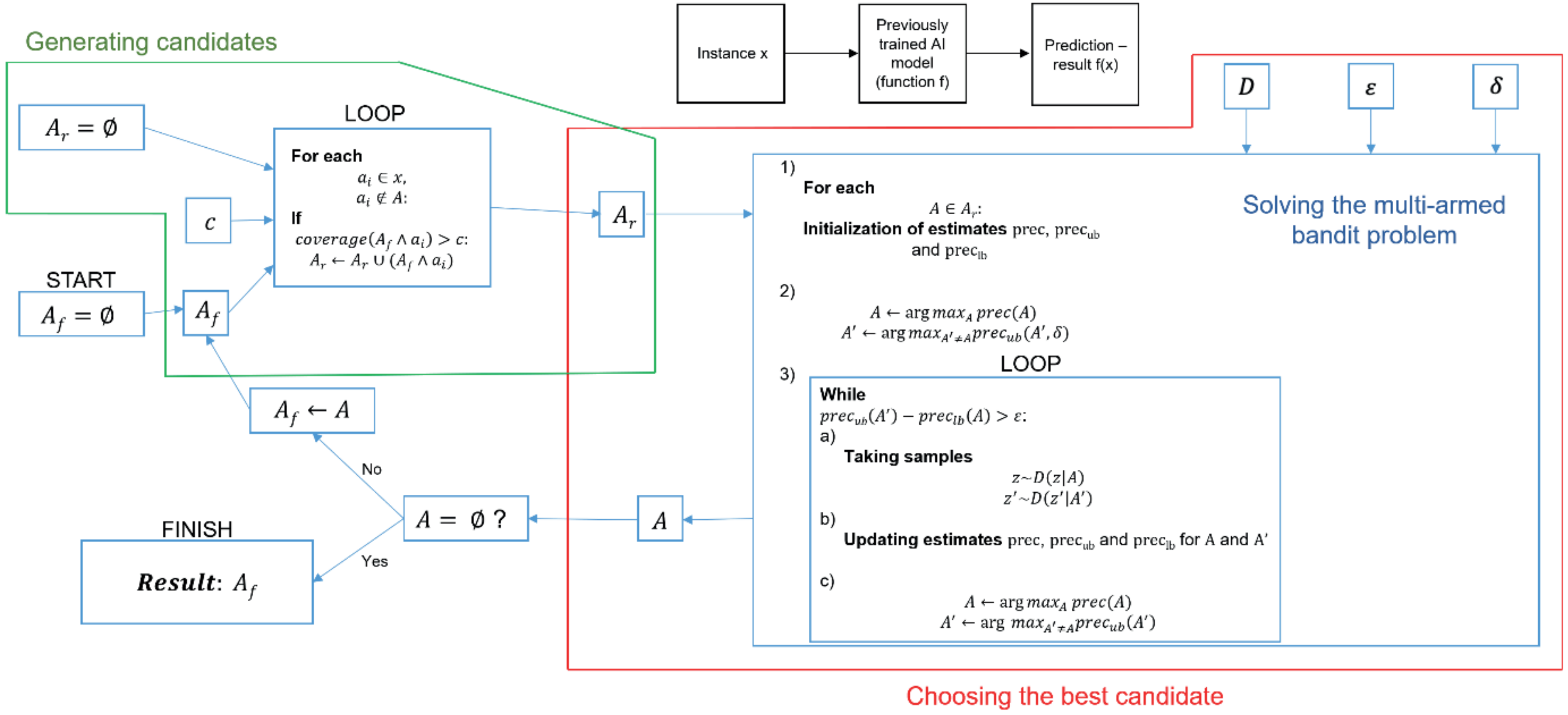

Instance x
Previously trained AI model (function f)
Prediction – result f(x)
Generating candidates
$A_r = \emptyset$
START
$A_f = \emptyset$
c
A_f
LOOP
For each
$a_i \in x,$
$a_i \notin A$:
If
$coverage(A_f \wedge a_i) > c$:
$A_r \leftarrow A_r \cup (A_f \wedge a_i)$
A_r
$A_f \leftarrow A$
No
Yes
$A = \emptyset$?
FINISH
Result: A_f
A
D
ε
δ
Solving the multi-armed bandit problem
1)
For each
$A \in A_r$:
Initialization of estimates prec, prec$_{ub}$ and prec$_{lb}$
2)
$A \leftarrow \arg max_A \, prec(A)$
$A' \leftarrow \arg max_{A' \neq A} prec_{ub}(A', \delta)$
3)
LOOP
While
$prec_{ub}(A') - prec_{lb}(A) > \varepsilon$:
a)
Taking samples
$z \sim D(z|A)$
$z' \sim D(z'|A')$
b)
Updating estimates prec, prec$_{ub}$ and prec$_{lb}$ for A and A'
c)
$A \leftarrow \arg max_A \, prec(A)$
$A' \leftarrow \arg \, max_{A' \neq A} prec_{ub}(A')$
Choosing the best candidate

Fig. 4. Anchors algorithm (Identifying the Best Candidate for Greedy) operation (developed with reference to the research performed by Ribeiro et al. (2018)). At the top of the image in the middle there is a simplified diagram showing the generalized operation of the AI model, the rest of the image is a diagram showing the operation of the Anchors algorithm

Legend: x – data instance under consideration; f – predictive model; A_f – rule (anchor) consisting of feature predicates; A_r – set of anchor candidates generated for the next round (iteration); c – threshold coverage value; a_i – individual feature predicate from instance x; D – perturbation distribution used to generate the modified instances of x; ε – precision error tolerance; δ – confidence level; *prec* – average of the rules precision estimated range; $prec_{ub}$ – upper bound of the rules precision estimated range; $prec_{lb}$ – lower bound of the rules precision estimated range; arg *max* – argument maximum; z – data sample from the A rule perturbation distribution; z' – data sample from the alternative A' rule perturbation distribution)

Table 2. Comparison of precision and coverage metrics used to find the best anchor candidate (Ribeiro et al., 2018)

Precision	Coverage
It tells us how likely it is that the model's predictions are unchanged under the conditions defined by the anchor	Refers to the proportion of data instances for which the anchor is applicable. It tells us how much of the data space is described by the anchor conditions, i.e., what is the probability that a randomly selected instance satisfies the anchor conditions
$Precision = \frac{True\ positive}{True\ positive + False\ positive}$ where: *True positive* – correctly predicted positive cases, *False positive* – incorrectly predicted positive cases	$Coverage = \frac{Total\ number\ of\ times\ the\ rule\ was\ applied}{Total\ number\ of\ positive\ cases}$
High precision means that we can trust the anchor as a solid explanation for the set of instances	High coverage means the anchor can be applied to more instances, increasing its usefulness

In addition to the described Identifying the Best Candidate for Greedy algorithm, there is also an algorithm based on searching multiple paths simultaneously (Outline of the Beam Search), the difference is that in each iteration, it tries to select more than one candidate, so you can analyze several solutions at the same time and finally select the anchor actually the best, because this algorithm helps to avoid local maxima, but at the cost of greater computational resource consumption (Ribeiro et al., 2018).

2.5. Anchors in other papers

The Anchors method is used, among others, in the financial industry. In the paper written by Thanathamathee et al. (2024) the topic of improving the mechanism of forecasting the chance of continuation of the enterprise's operations was discussed by using the XGBoost model with the attention mechanism in combination with the XAI Anchors method, which is able to generate understandable explanations regarding the predictions of the AI model. The financial analysis presented in the paper allows to better assess whether the company is able to continue its operations without the risk of bankruptcy. The authors of the paper think that traditional models often lose their ability to explain their predictions due to the high level of complexity of such a model, which results in a loss of trust among users. The introduction of XAI makes it possible to obtain more transparent forecasts, also highlighting key financial indicators that influence the forecast result.

2.6. Other XAI methods

There are many XAI methods, each with its own special features and working well in specific applications or being more universal. Two methods have been described in more detail, but it is also worth considering others, especially when the above two methods do not work in solving a specific problem. Very popular XAI methods are LIME and SHAP.

LIME (Local Interpretable Model-agnostic Explanations) explains the predictions of complex models locally by building a simple model (e.g. linear regression) in the vicinity of the analyzed example. Its range of applications is wide, it can be used in combination with any AI model for classification tasks (it is model-agnostic) (Ribeiro et al., 2016).

SHAP (Shapley Additive Explanations) is a game-theoretic method that uses Shapley values to assign each feature a contribution to the model's prediction. It explains individual forecasts and the overall operation of the model, indicating the features that have the greatest impact on the results. This method is used for various high-complexity models, such as neural networks (Lundberg & Lee, 2018).

In the paper written by Ahmad Khan et al. (2024) the topic of using machine learning and XAI to predict failures of the braking system in trucks was discussed. The main goal is to develop a Predictive Maintenance method, which is necessary to prevent failures and improve the safety and efficiency of trucks. The XAI methods used are SHAP and LIME, and in addition to generating explanations understandable to users, one of the reasons for using XAI was also to reduce computational complexity while maintaining high model accuracy. The authors managed to extract the 20 most important features out of 171, resulting in a significant simplification of the model and reducing training time while maintaining a similar level of accuracy. This shows that XAI can not only serve as a tool to help understand how AI models work, but can also support the optimization processes of these models.

3. Real problem and XAI solution

3.1. Problem description

A part of the case study was the problem of identifying ladles in images from CCTV cameras in the electrosteel plant. A ladle is a specialized container in which molten steel is transported and processed. It is made of refractory material that allows the steel to safely maintain high temperatures during all processes. Ladles must often be transported between different places in the electrosteel plant in order to go through subsequent stages of steel production, for this purpose special cranes are used on which the ladles can move.

There are many such ladles in the hall, they are at various stages of production and change their location from time to time, so it is important to properly monitor them to maintain control over them and the processes with which the ladles are related. There are also a lot of people doing their jobs in the workplace, so it is important to maintain order and proper arrangement among the machines for their safety.

Due to the above, there was a need to introduce a system for identifying all ladles in use by using CCTV cameras installed in the hall that continuously collect images of what is happening in the workplace. Additionally, having an AI model used to identify objects, we can combine it with the appropriate XAI method, which will allow us to understand what criteria the model used when making identification, and show us the image fragments that are crucial for the AI model in recognizing ladles among others objects.

3.2. Industrial data

The data used to develop the solution are image snapshots from CCTV cameras showing various places in the hall from different perspectives. Examples of pictures recorded by cameras and submitted for research are presented in Figure 5.

3.3. Solution

The implementation of the solution was divided into two steps: the first one concerns the identification of the ladles in the image, and the second one concerns the generation of heat maps explaining the detection results by indicating key fragments in the image.

3.3.1. Solution implementation, step 1: ladles identification – YOLO v3

The YOLO (You Only Look Once) v3 model was used to identify ladles in the image, based on a neural network called Darknet-53, it uses three different scales to detect objects (small, medium and large objects), so it can be said that the model follows three paths simultaneously. In addition, YOLO v3 can detect many classes of objects at the same time assigning them appropriate labels and frames limiting the areas of these objects in the image (Jiang et al., 2022). This model is fast and accurate, making it very popular.

The construction of the used YOLO model starts from the input layer, then we go through the *yolo_darknet* layer, i.e. we use the Darknet-53 neural network. The next layers are the previously mentioned branches, which are responsible for detecting objects of a certain size. Each path consists of the following layers: *yolo_conv*, *yolo_output* and *yolo_boxes*. Finally, all three paths converge to the last layer of the YOLO model, *yolo_nms*.

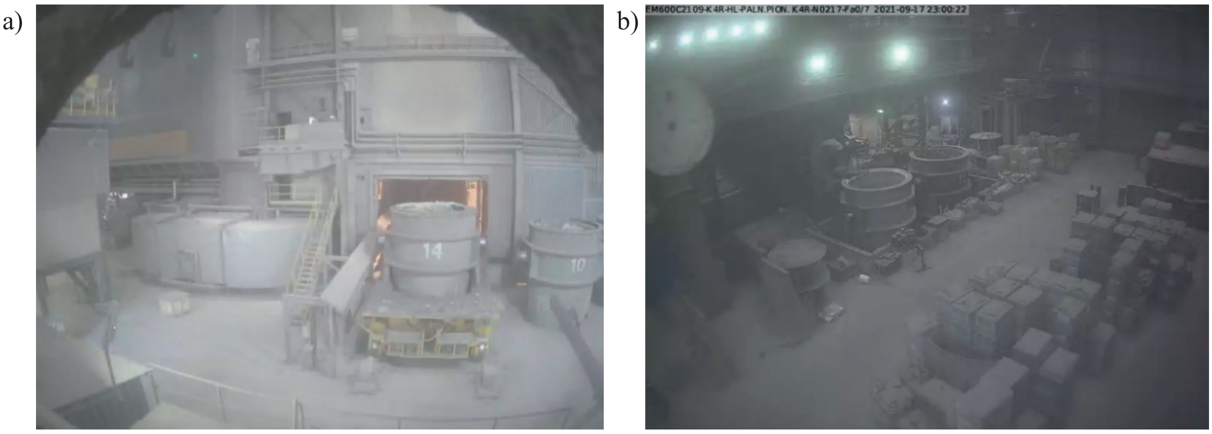

a)
14
10
b)
EM600C2109-K4R-HL-PALN.PION. K4R-N0217-Fa0/7 2021-09-17 23:00:22

Fig. 5. Pictures from CCTV cameras in the electrosteel plant of CMC Poland sp. z o.o. In the first one (a), two ladles are located in front of the camera on the right side of the image. In the second one (b), the view is presented from above, and two ladles are located on the other side of the room, approximately in the center of the image

3.3.2. Solution implementation, step 2: prediction explanation – GradCAM

GradCAM was used as the XAI method, which is usually used for classification tasks, and here the situation is slightly different, because the task of the mentioned XAI method is to explain what fragments in the image determined that the YOLO v3 model was able to identify ladles. In addition, the GradCAM method takes into account the last convolutional layer, and here the situation is more complicated, because YOLO in its algorithm follows three paths at the same time, where each path is responsible for the detection of objects with different dimensions, therefore we are dealing with three parallel convolutional layers, so GradCAM had to be adapted to all three, then three identical images were juxtaposed, but with different heat maps superimposed on them, and the results were compared.

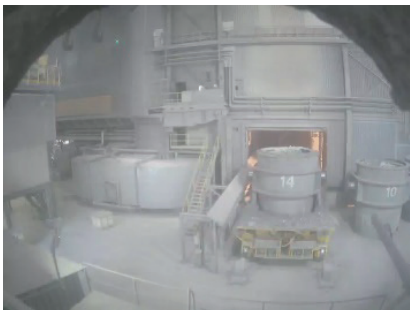
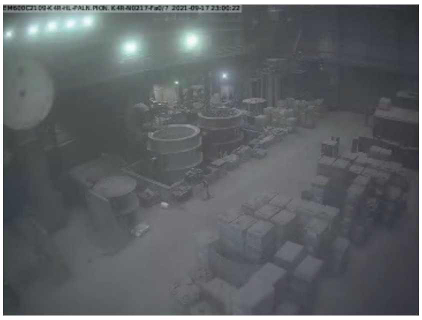
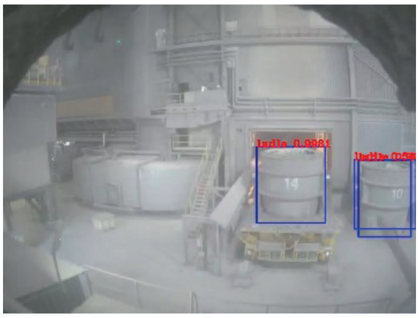
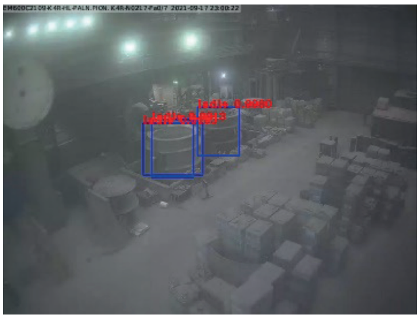
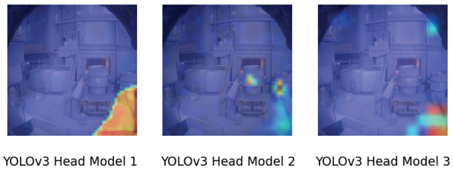
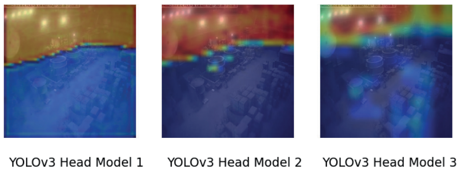
3.4. Results

The results show sample images recorded by CCTV cameras installed in the electrosteel plant hall compared to the same images after applying ladle detection using the YOLO v3 model and explaining the predictions by applying heat maps generated by the GradCAM method to the images (Tab. 3).

The YOLO model coped with the task of identifying appropriate objects, did not miss any ladle, but it can be noticed that in both cases one ladle was identified twice, as evidenced by two frames superimposed on the same object. These frames have different dimensions, which indicates that the dimensions of the ladle in the image were on the border of the two size scales distinguished by the YOLO model, i.e. two of the three paths the model followed when solving the problem identified the same object.

The GradCAM model generated heat maps for each of the three size scales considered by the YOLO model (Head Model 1 – small objects, Head Model 2 – medium-sized objects, Head Model 3 – large objects). We can do the most by analyzing the heat map for Head Model 2 – there we can notice that the model has started to recognize the ladles by the numbers they were marked with to identify them in the hall. GradCAM also tells us that for the YOLO v3 model the identified ladles are rather medium-sized, as it is at this scale that the heat maps are clearest and point to the identified objects.

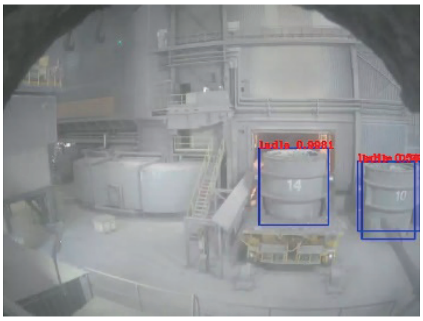
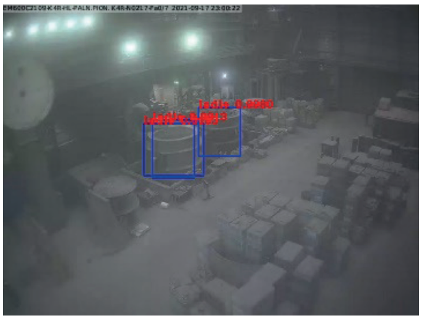
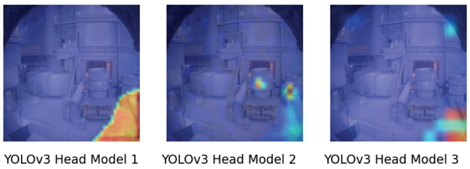
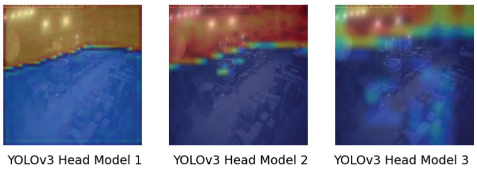
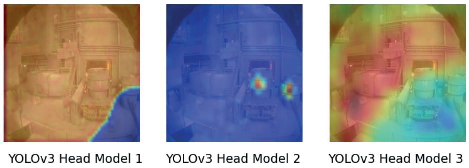
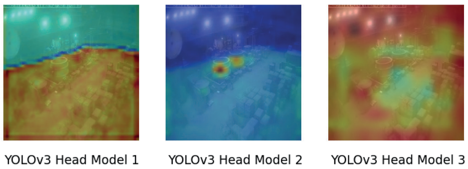
Table 3. Example results of the YOLO v3 model and the GradCAM method on input data in the form of camera images

	Example 1	Example 2
Input image		
Detection result (YOLO v3)		
Explanation result (GradCAM)	 YOLOv3 Head Model 1 YOLOv3 Head Model 2 YOLOv3 Head Model 3	 YOLOv3 Head Model 1 YOLOv3 Head Model 2 YOLOv3 Head Model 3

3.5. Use of an improved version of GradCAM – GradCAM++

GradCAM can, to some extent, generate a visual explanation indicating those image elements that most influenced the detection result, but the marked areas on the ladles are small and not very clear (blurred), therefore, for comparison, an improved version of the XAI method was also used, i.e. GradCAM++. It can calculate first- and second-order gradients, which allows it to capture more detailed location information and generate more precise heat maps (Soomro et al., 2024). The comparison of the results is presented in Table 4.

Table 4. Comparison of GradCAM and GradCAM++ results

	Example 1	Example 2
Detection result (YOLO v3)		
Explanation result (GradCAM)	 YOLOv3 Head Model 1 YOLOv3 Head Model 2 YOLOv3 Head Model 3	 YOLOv3 Head Model 1 YOLOv3 Head Model 2 YOLOv3 Head Model 3
Explanation result (GradCAM++)	 YOLOv3 Head Model 1 YOLOv3 Head Model 2 YOLOv3 Head Model 3	 YOLOv3 Head Model 1 YOLOv3 Head Model 2 YOLOv3 Head Model 3

In the case of GradCAM++, Head Model 2 is still able to tell us the most, because the corresponding convolutional layer is subject to greater activation than the others, while the generated heat maps are more extensive and detailed than in the case of the traditional GradCAM model.

4. Conclusions

XAI is another step forward towards the further development of algorithms that make predictions related to a specific problem. The emphasis on the need to explain decisions made by AI will increase our awareness of how these models work and what criteria they are guided by. There are many XAI methods which can be divided into different categories depending on the aspect we look at, and which algorithm to choose depends on individual needs.

XAI, just like conventional AI, can be used in heavy industry; as it develops, it will cope better with current tasks and will be used in increasingly complex challenges. Using GradCAM to generate explanations of the identification results made by the YOLO v3 model gave satisfactory results, it helped to see the key places thanks to which the YOLO model could distinguish the ladles from other objects in the image, while using the improved GradCAM++ method gave even more accurate marking of these areas.

Further research should focus on finding other methods that would be able to extract other features crucial to understanding the performance of the YOLO model in this case.

Acknowledgement

The financial support of Polish National Centre for Research and Development project no. POIR.01.01.01-00-0996/19 is acknowledged.

References

Ahmad Khan, M., Khan, M., Dawood, H., Dawood, H., & Daud, A. (2024). Secure Explainable-AI approach for brake faults prediction in heavy transport. *IEEE Access*, *12*, 114940–114950. https://doi.org/10.1109/ACCESS.2024.3444907

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, *10*(7), e0130140. https://doi.org/10.1371/journal.pone.0130140

Chattopadhay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. (2018). Grad-CAM++: generalized gradient-based visual explanations for deep convolutional networks. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 839–847. https://doi.org/10.1109/WACV.2018.00097

Ciatto, G., Schumacher, M. I., Omicini, A., & Calvaresi, D. (2020). Agent-based explanations in AI: towards an abstract framework. In D. Calvaresi, A. Najjar, M. Winikoff, & K. Främling (Eds.), *Lecture Notes in Computer Science: Vol. 12175. Explainable, Transparent Autonomous Agents and Multi-Agent Systems* (pp. 3–20). Springer. https://doi.org/10.1007/978-3-030-51924-7_1

Craven, M. W., & Shavlik, J. W. (1996). Extracting tree-structured representations of trained networks. In D. S. Touretzky, M. C. Mozer, M. E. Hasselmo (Eds.), *Advances in Neural Information Processing Systems 8* (pp. 24–30). MIT Press.

Elenberg, E. R., Dimakis, A. G., Feldman, M., & Karbasi, A. (2018). Streaming weak submodularity: interpreting neural networks on the fly. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30* (pp. 4047–4055). Curran Associates.

Fong, R. C., & Vedaldi, A. (2017). Interpretable explanations of black boxes by meaningful perturbation. In *2017 IEEE International Conference on Computer Vision* (pp. 3429–3437). https://doi.org/10.1109/ICCV.2017.371

Friedman, J. H., & Popescu, B. E. (2008). Predictive learning via rule ensembles. *The Annals of Applied Statistics*, *2*(3), 916–954. https://doi.org/10.1214/07-AOAS148

Gu, J., & Tresp, V. (2019). Contextual prediction difference analysis. *ArXiv*, arXiv:1910.09086. https://doi.org/10.48550/arXiv.1910.09086

Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., & Giannotti, F. (2018). Local rule-based explanations of black box decision systems. *ArXiv*, arXiv:1805.10820. https://doi.org/10.48550/arXiv.1805.10820

Gulum, M. A., Trombley, Ch. M., & Kantardzic, M. (2021). A review of explainable deep learning cancer detection models in medical imaging. *Applied Sciences*, *11*(10), 4573. https://doi.org/10.3390/app11104573

Hall, P., Gill, N., Kurka, M., & Phan, W. (2024). *Machine Learning Interpretability with H2O Driverless AI* (A. Bartz, Ed.). H2O.ai.

Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A review of YOLO algorithm developments. *Procedia Computer Science*, *199*, 1066–1073. https://doi.org/10.1016/j.procs.2022.01.135

Kim, B., Khanna, R., & Koyejo, O. (2017). Examples are not enough, learn to criticize! Criticism for interpretability. In D. D. Lee, U. Von Luxburg, R. Garnett, M. Sugiyama, I. Guyon (Eds.), *Advances in Neural Information Processing Systems 29* (pp. 2280–2288). Curran Associates.

Lundberg, S. M., & Lee, S.-I. (2018). A unified approach to interpreting model predictions. In U. Von Luxburg, I. Guyon, S. Bengio, H. Wallach, R. Fergus, S. V. N. Vishwanathan, R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30* (pp. 4765–4774). Curran Associates.

Mothilal, R. K., Sharma, A., & Tan, Ch. (2020). Explaining machine learning classifiers through diverse counterfactual explanations. In *FAT* ‘20: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (pp. 607–617). https://doi.org/10.1145/3351095.3372850

Petsiuk, V., Das, A., & Saenko, K. (2018). RISE: Randomized Input Sampling for Explanation of black-box models. *ArXiv*, arXiv:1806.07421. https://doi.org/10.48550/arXiv.1806.07421

Plumb, G., Molitor, D., & Talwalkar, A. (2019). Model agnostic supervised local explanations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), *Advances in Neural Information Processing Systems 31* (pp. 2515–2524). Curran Associates.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). “Why should I trust you?”: Explaining the predictions of any classifier. In *KDD ‘16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135–1144). https://doi.org/10.1145/2939672.2939778

Ribeiro, M. T., Singh, S., & Guestrin, C. (2018). Anchors: High-Precision Model-Agnostic Explanations. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1), 1527–1535. https://doi.org/10.1609/aaai.v32i1.11491

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2020). Grad-CAM: visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, *128*(2), 336–359. https://doi.org/10.1007/s11263-019-01228-7

Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning important features through propagating activation differences. In D. Precup, Y. Whye Teh (Eds.), *ICML’17: Proceedings of the 34th International Conference on Machine Learning* (vol. 70, pp. 3145–3153).

Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: visualising image classification models and saliency maps. *ArXiv*, arXiv:1312.6034. https://doi.org/10.48550/arXiv.1312.6034

Smilkov, D., Thorat, N., Kim, B., Viégas, F., & Wattenberg, M. (2017). SmoothGrad: removing noise by adding noise. *ArXiv*, arXiv:1706.03825. https://doi.org/10.48550/arXiv.1706.03825

Sofianidis, G., Rožanec, J. M., Mladenić, D., & Kyriazis, D. (2021). A review of explainable artificial intelligence in manufacturing. *ArXiv*, arXiv:2107.02295. https://doi.org/10.48550/arXiv.2107.02295

Sokol, K., & Flach, P. (2024). LIMEtree: Consistent and faithful multi-class explanations. *ArXiv*, arXiv:2005.01427. https://doi.org/10.48550/arXiv.2005.01427

Soomro, S., Niaz, A., & Nam Choi, K. (2024). Grad++ScoreCAM: Enhancing visual explanations of deep convolutional networks using incremented gradient and score-weighted methods. *IEEE Access*, *12*, 61104–61112. https://doi.org/10.1109/ACCESS.2024.3392853

Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: the all convolutional net. *ArXiv*, arXiv:1412.6806. https://doi.org/10.48550/arXiv.1412.6806

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. In D. Precup, Y. Whye Teh (Eds.), *ICML'17: Proceedings of the 34th International Conference on Machine Learning* (vol. 70, pp. 3319–3328). https://dl.acm.org/doi/10.5555/3305890.3306024

Thanathamathee, P., Sawangarreerak, S., & Nizam, D. N. M. (2024). Enhancing going concern prediction with anchor explainable AI and attention-weighted XGBoost. *IEEE Access*, *12*, 68345–68363. https://doi.org/10.1109/ACCESS.2024.3401007

Thombre, A. (2024). Explainable AI (XAI): Using decision trees to explain neural network model. *ResearchGate*. https://www.researchgate.net/publication/383898176_Explainable_AI_XAI_Using_decision_trees_to_explain_neural_network_model

Ustun, B., Tracà, S., & Rudin, C. (2013). Supersparse linear integer models for interpretable classification. *ArXiv*, arXiv:1306.6677. https://doi.org/10.48550/arXiv.1306.6677

Vilone, G., & Longo, L. (2020). Explainable Artificial Intelligence: a systematic review. *ArXiv*, arXiv:2006.00093. https://doi.org/10.48550/arXiv.2006.00093

Waa, J., van der, Robeer, M., Diggelen, J., van, Brinkhuis, M., & Neerincx, M. (2018). Contrastive explanations with local foil trees. *ArXiv*, arXiv:1806.07470. https://doi.org/10.48550/arXiv.1806.07470

Zafar, M. R., & Khan, N. M. (2019). DLIME: A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems. *ArXiv*, arXiv:1906.10263. https://doi.org/10.48550/arXiv.1906.10263

Zeiler, M. D., & Fergus, R. (2013). Visualizing and understanding convolutional networks. *ArXiv*, arXiv:1311.2901. https://doi.org/10.48550/arXiv.1311.2901

Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2921–2929. https://doi.org/10.1109/CVPR.2016.319

Zintgraf, L. M., Cohen, T. S., Adel, T., & Welling, M. (2017). Visualising deep neural network decisions: prediction difference analysis. *ArXiv*, arXiv:1702.04595. https://doi.org/10.48550/arXiv.1702.04595