XAI Method Properties: A (Meta-)study*

Gesina Schwalbe[1,2][0000−0003−2690−2478] and Bettina Finzel[2][0000−0002−9415−6254]

[1] Continental AG, Regensburg, Germany
gesina.schwalbe@continental-corporation.com
[2] Cognitive Systems Group, University of Bamberg, Germany
bettina.finzel@uni-bamberg.de

Abstract. In the meantime, a wide variety of terminologies, motivations, approaches and evaluation criteria have been developed within the scope of research on explainable artificial intelligence (XAI). Many taxonomies can be found in the literature, each with a different focus, but also showing many points of overlap. In this paper, we summarize the most cited and current taxonomies in a meta-analysis in order to highlight the essential aspects of the state-of-the-art in XAI. We also present and add terminologies as well as concepts from a large number of survey articles on the topic. Last but not least, we illustrate concepts from the higher-level taxonomy with more than 50 example methods, which we categorize accordingly, thus providing a wide-ranging overview of aspects of XAI and paving the way for use case-appropriate as well as context-specific subsequent research.

Keywords: Explainable Artificial Intelligence · Taxonomy · Meta-Analysis · Survey · Methods

1 Introduction

Machine learning models offer the great benefit that they can deal with hardly specifiable problems as long as these can be exemplified by data samples. This has opened up a lot of opportunities for promising automation and assistance systems, like highly automated driving, medical assistance systems, text summaries and question-answer systems, just to name a few. However, many types of models that are automatically learned from data will not only exhibit high performance, but also be black-box, hiding information on the learning progress, internal representation, and final processing in a format not or hardly interpretable by humans.

There are now diverse use-case specific motivations for allowing humans to *understand* a given software component, *i.e.* to build up a mental model approximating the algorithm in a certain way. This starts with legal reasons, like

* The research leading to these results is funded by the BMBF ML-3 project Transparent Medical Expert Companion (TraMeExCo), FKZ 01IS18056 B, 2018–2021, and by the German Federal Ministry for Economic Affairs and Energy within the project "KI Wissen – Automotive AI powered by Knowledge". The authors would like to thank the consortium for the successful cooperation.

the General Data Protection Regulation [37] adopted by the European Union in recent years. Another example are domain specific standards, like the functional safety standard ISO 26262 [50] requiring assessibility of software components in safety critical systems, which is detailed to a requirement for explainability of machine learning based components in the ISO/TR 4804 [49] draft standard. Many further reasons of public interest like fairness or security, and business interests like ease of debugging, knowledge retrieval, or appropriate user trust have been identified [4,61]. This need to translate behavioral or internal aspects of black-box algorithms into a human interpretable form gives rise to the broad topic of explainable artificial intelligence (XAI).

In recent years, the topic of XAI methods has received an exponential boost in research interest [4,64,1,121]. For practical application of XAI in human-AI interaction systems, it is important to ensure a choice of XAI method(s) appropriate for the corresponding use-case. While thorough use-case analysis including the main goal and derived requirements is one essential ingredient here [61], we argue that a necessary foundation for choosing correct requirements is a complete knowledge of the different *aspects* (traits, properties) of XAI methods that may influence their applicability. Well-known aspects are *e.g.* portability, so whether the method requires access to the model internals or not, or locality, so whether single predictions are explained or some global properties of the model to explain. As will become clear from our literature analysis in section 2, this only just scratches the surface of application relevant aspects of XAI methods.

This paper aims to help for one practitioners seeking a categorization scheme for choosing an appropriate XAI method for their use-case, and secondly researchers in identifying desired combination of aspects that have not or little been considered so far. For this, we provide a complete collection and a structured overview in the form of a taxonomy of XAI method aspects, together with method examples for each aspect. The method aspects are obtained from an extensive literature survey on categorization schemes for explainability and interpretability methods, resulting in the first meta-study on XAI surveys of our knowledge. Other than similar work, we do not aim to provide a survey on XAI methods, but rather gather the valuable work done so far into a good starting point for in-depth understanding of sub-topics of XAI research, and research on XAI methods themselves.

Our main contributions are:

- A detailed and complete taxonomy containing and structuring application relevant XAI method aspects so far considered in literature (see Figure 2).
- A large collection of more than 50 surveys on XAI methods as starting material for research on the topic (see Tables 1, 2, 3, 4). To our knowledge, this represents the first meta-study on XAI methods.
- A large and diverse collection of more than 50 XAI methods presented as examples for the method aspects with a final detailed categorization by main method aspects (see Table 5).

The rest of the paper reads as follows: In the following subsections we first introduce in more depth some basic notions of explainable AI for readers less familiar with the topic (subsection 1.1), and then give some details on our review approach (subsection 1.2). The remainder of this work in section 2 then details XAI method aspects and the suggested taxonomy thereof, following a procedural approach. The aspects are each accompanied by illustrating examples, which are finally summarized and sorted into the main aspects of the taxonomy in Table 5.

1.1 What is XAI?

In order to overcome the opaqueness and black-box character of end-to-end machine learning approaches, various methods for explainable artificial intelligence (XAI) have been developed and applied during the last years. From the development of new techniques emerged the usage of different terms and concepts to distinguish XAI methods.

In the following we shortly introduce important terms and concepts in order to give the reader, who may not be familiar with XAI, a general glance on this field. We are not aiming for presenting a comprehensive summary and refer to state-of-the-art surveys instead.

The concept of XAI exists already since many decades, but it was only against the background of increasing demand for trustworthy and transparent machine learning ([1]) that the term XAI was introduced to the research community in 2017 by the Defense Advanced Research Projects Agency (DARPA), see [39].

According to DARPA, XAI efforts aim for two main goals. The first one is to create machine learning techniques that produce models that can be explained (their decision-making process as well as the output), while maintaining a high level of learning performance. It further should convey a user-centric approach, to enable humans to understand their artificial counterparts. As a consequence, XAI aims for increasing the trust in learned models and to allow for an efficient partnership between human and artificial agents ([39]).

In order to reach the first goal, DARPA proposes three strategies: deep explanation, interpretable models and model induction.

Deep explanation refers to combining deep learning with other methods in order to create hybrid systems that produce richer representations of what a deep neural network has learned and that enable to extract underlying semantic concepts ([39]).

Interpretable models are defined as techniques that learn more structured representations or that allow for tracing causal relationships.

The strategy of **model induction** summarizes techniques, which are used to infer an approximate explainable model by observing the input-output behaviour of a model that is explained.

The second, more user-centric, goal requires a highly inter-disciplinary perspective, based on fields such as computer science, social sciences as well as psychology in order to produce **more explainable models**, suitable **explanation interfaces** and to **communicate explanations effectively** under consideration of psychological aspects.

According to [12] literature on user-centric perspectives describes aspects that add up to a process involving recognition, understanding and explicability/explainability as well as interpretability. The output of such a process is *local interpretability* for explainable artificial intelligence methods and *global interpretability* for interpretable machine learning. Systems that follow both tracks of interpretability are called *comprehensible artificial intelligence* (see [12]). An overview is given in Fig. 1 which was taken from [12]).

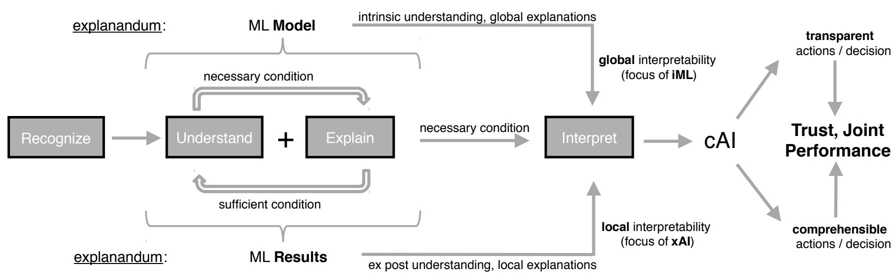

Fig. 1. A framework for comprehensible artificial intelligence [12]

According to the authors, understanding is described as the ability to recognize correlations, the context, and is a necessary precondition for explanations. Explaining can take place for two reasons: explicability or explainability. Explicability refers to making properties of a model transparent. For example in [87] a visual explanation method is applied to make explicit on which regions in an image a deep neural net focused to recognize distinct facial expressions. The analysis shows that in some cases the neural net looked at the background rather than the face for certain facial expressions and human participants. Through explicability domain experts can extract information and verify results. Explainability goes one step further and aims for comprehensibility. This means that the reasoning, the model or the evidence for a result can be explained such that the context can be understood by a human. The ultimate goal of such systems would be, to reach *ultra-strong machine learning*, where machine learning would help humans to improve in their tasks. For example [74] examined the comprehensibility of programs learned with Inductive Logic Programming and [92] showed that comprehensibility of such programs could help laymen to understand how and why a certain prediction was made. Both, understanding and explainability can be seen as necessary preconditions to fulfill interpretability. Interpretability can be reached on two different levels: globally and locally. While the global perspective explains the model and its logic as a whole ("How was the conclusion derived?"), local approaches aim for explaining individual decisions or predictions ("Why was this example classified as a car?"), independent of the model's internal structure ([1]). According to the literature, transparent and comprehensible artificial intelligence relies on interpretability on one hand, and interactivity on the other hand [47]. Especially correctability is an enabler of understanding the internal working of a model and provides methods to model adaption (see Interactivity in section 2.2).

By combining different algorithmic approaches as well as by providing multi-modal explanations, various types of users and input data and divers use cases can be satisfied. Thus, different perspectives need to be taken into account. We therefore aim to provide an extensive overview on the topic of XAI in this paper and want to provide a starting point for further research on finding methods appropriate to specific use cases and contexts. Our approach on how we searched for relevant work is described in the next subsection, followed by the chapter, where we present our proposed taxonomy.

1.2 Approach

One goal of this paper is to provide a complete overview of relevant aspects or properties of XAI methods. In order to achieve this, a systematic and broad literature analysis was conducted on papers in the time range of 2010 to 2021.

Search: Work on XAI taxonomies Firstly, we identified common terms associated directly with XAI taxonomies (for abbreviations both the abbreviation and the full expression must be considered):

- machine learning terms: AI, DNN, Deep Learning, ML
- explainability terms: XAI, explain, interpret
- terms associated with taxonomies: taxonomy, framework, toolbox, guide

We then collected google scholar search results for combinations of these terms. Most notably, we considered the search phrases "explain AI taxonomy" (more than 30 pages of results), "XAI taxonomy toolbox guide" (8 pages of results), "explainable AI taxonomy toolbox guide" (more than 30 pages of results), and the combination of all search terms "explain interpret AI artificial intelligence DNN Deep Learning ML machine learning taxonomy framework toolbox guide XAI" (2 pages of results). For each search, the first 30 pages of results were scanned for the title, promising results then were scanned for the abstract.

Search: General XAI Surveys In another iteration we collected search results for XAI surveys not necessarily proposing, but possibly implicitly using, a taxonomy. For this, we again conducted a search, now for the more general terms "XAI" and "XAI survey", which again were scanned first by title, then by abstract. This resulted in a similar number of finally chosen and in-depth assessed papers as the taxonomy search (not counting duplicate search results).

Results The search resulted in over 50 surveys on XAI, most of them from the years 2018 to 2021, that were analysed for XAI method aspects, taxonomy structuring proposals, and suitable example methods for each aspect. A selection of surveys is shown in Tables $1, 2, 3, 4$. For the selection, first the citation count of the surveys was collected from the popular citation databases google scholar[3], semantic scholar[4], opencitations[5], and NASA ADS[6]. The highest result (mostly google scholar) was chosen for comparison. Finally, we used as selection criteria the citation score per year, the recency, and whether the specificity, so surveys focusing on a concrete sub-topic of explainability.

To exemplify the aspects of our proposed taxonomy, we selected again more than 50 concrete XAI methods that are reviewed in example sections for the corresponding XAI aspects. The selection focused on high diversity and recency of the methods, in order to establish a broad view on the XAI topic. Finally, each of the methods were analysed on main taxonomy aspects, which is summarized in the overview table Table 5. Also some examples of larger toolboxes are collected in Table 4.

Table 1. Selected surveys on XAI methods with a broad focus.

General XAI method collections

[64]	Linardatos et al.	2021	Extensive survey on XAI methods with code and toolbox references
[48]	Islam et al.	2021	Shallow taxonomy with some example methods explained for each aspect, a short meta-study of XAI surveys, and a collection of future perspectives for XAI
[73]	Mueller et al.	2021	Design principles and survey on metrics for XAI systems
[121]	Zhou et al.	2021	Detailed review on XAI metrics with a shallow taxonomy both for methods and metrics
[70]	Molnar	2020	Book on interpretability methods including details on many transparent and many model-agnostic methods
[4]	Arrieta et al.	2020	Extensive and diverse XAI method collection for responsible AI
[111]	Xie et al.	2020	Introduction to XAI with wide variety of examples of standard methods
[23]	Das & Rad	2020	Presents a review and taxonomy for local and global explanations based on backpropagation and pertubation-based methods (model-specific versus model-agnostic)
[104]	Vilone & Longo	2020	Extensive survey on methods, with overview tables mapping methods to (few) properties
[11]	Benchekroun et al.	2020	Presents a preliminary taxonomy that includes pre-modelling explainability as an approach to link knowledge about data with knowledge about the used model and its results; motivates standardization
[14]	Carvalho et al.	2019	extensive collection of different XAI aspects, especially metrics, with some examples
[40]	Gunning et al.	2019	Very short low-level introduction to XAI and open research directions
[2]	Adadi & Berrada	2018	quite extensive literature survey of 381 papers related to XAI
[35]	Gilpin et al.	2018	Extensive survey on different kinds of XAI methods including rule extraction and references to further more specialized surveys

[3] https://scholar.google.com
[4] https://www.semanticscholar.org/
[5] https://opencitations.net/
[6] https://ui.adsabs.harvard.edu/

Table 2. Selected meta-studies on the topic of explainable AI.

XAI meta-studies		
[61] Langer et al.	2021	Reviews XAI with respect to stakeholder understanding and satisfaction of stakeholder desiderata; discusses the context of stakeholder tasks
[19] Chromik & Schüßler	2020	Rigorous taxonomy development for XAI methods from an HCI perspective
[30] Ferreira et al.	2020	Survey that refers to further XAI surveys and presents a taxonomy of XAI among computer science and human-computer-interaction communities
[72] Mueller et al.	2019	Detailed DARPA report respectively meta-study on state-of-the-literature on XAI including a detailed list of XAI method aspects and metrics (Chap. 7, 8)
[65] Lipton	2018	Small generic XAI taxonomy with discussion on desiderata for interpretability
[27] Doshi-Velez & Kim	2017	collection of latent dimensions of interpretability with recommendations on how to choose and evaluate a method

2 Taxonomy

In this section, a taxonomy of XAI methods is established by selecting key dimensions for their classification. The categorization is done in a procedural manner: One usually should start with the *problem definition* (subsection 2.1), before detailing the *explanator properties* (subsection 2.2). Lastly, we discuss different *metrics* (subsection 2.3) that can be applied to explanation systems. For an overview of the taxonomy, see Figure 2. The presented aspects are illustrated by selected example methods (marked in gray). The selection is by no means complete but rather should give an impression about the wide range of XAI methods and how to apply our taxonomy to both some well-known and less known but interesting methods. An overview over valuable further sources is given in Tables 1, 2, 3, 4.

In the following we will be using as nomenclature:

Explanandum *(what is to be explained)* The complete oracle to be explained. This usually encompasses a model (*e.g.* a deep neural network), which may or may not encompass the actual object of explanation.

Explanator *(the one who explains)* This is the system component providing explanations.

Explainee *(the one to whom is explained)* This is the receiver of the explanations. Note that this often but not necessarily is a human. Explanations may also be used *e.g.* in multi-agent systems for communication between the agents and without a human in the loop in most of the information exchange scenarios.

Human-AI system A system containing both algorithmic components and a human actor that have to cooperate for achieving a goal. We here consider in specific explanation systems, *i.e.* such human-AI systems in which the

Table 3. Selected domain specific surveys on XAI methods.

Domain specific XAI surveys

[46]	Heuillet et al.	2021	Survey on XAI methods for reinforcement learning
[118]	Zhang & Chen	2020	Survey on recommendation systems with good overview on models deemed explainable
[103]	Tjoa & Guan	2020	Survey with focus on medical XAI, sorting methods into a shallow taxonomy
[21]	Cropper et al.	2020	Survey on inductive logic programming methods for constructing rule-based transparent models
[96]	Singh et al.	2020	Survey and taxonomy of XAI methods for image classification with focus on medical applications
[22]	Danilevsky et al.	2020	Survey on XAI methods for natural language processing
[13]	Calegari et al.	2020	Overview of the main symbolic/sub-symbolic integration techniques for XAI
[9]	Baniecki & Biecek	2020	Presents challenges in explanation, traits to overcome these as well as a taxonomy for interactive explanatory model analysis
[63]	Li et al.	2020	Review of state-of-the art metrics to evaluate explanation methods and experimental assessment of performance of recent explanation methods
[81]	Puiutta & Veith	2020	Review with short taxonomy on XAI methods for reinforcement learning
[90]	Samek et al.	2019	Short introductory survey on visual explainable AI
[38]	Guidotti et al.	2018	Review of model-agnostic XAI methods with a focus on XAI for tabular data
[117]	Zhang & Zhu	2018	Survey on visual XAI methods for convolutional networks
[76]	Nunes & Jannach	2017	Very detailed taxonomy of XAI for recommendation systems (see Fig. 11)
[41]	Hailesilassie	2016	Review on rule extraction methods

Table 4. Examples of XAI toolboxes. For a more detailed collection we refer the reader to [64] and [23].

Examples of XAI toolboxes

[98]	Spinner et al.	2020	explAIner toolbox
[3]	Alber et al.	2019	Interface and reference implementation for some standard saliency map methods
[5]	Arya et al.	2019	IBM AI explainability 360 toolbox with 8 diverse XAI methods
[75]	Nori et al.	2019	Microsoft toolbox InterpretML with 5 model-agnostic and 4 transparent XAI methods

cooperation involves explanations about an algorithmic part of the system (the explanandum) by an explanator to the human interaction partner (the explainee) resulting in an action of the human.

2.1 Problem Definition

The following aspects consider the concretion of the explainability problem. The first step in general should be to determine the use-case specific *requirements* for aspects of the explanator (see subsection 2.2), and possibly targeted metric values (see subsection 2.3). This should be motivated by the actual goal or desiderata of the explanation, which can be *e.g.* verifiability of properties like fairness, safety, and security, knowledge discovery, promotion of user adoption respectively trust, or many more. An extensive list of desiderata can be found in [61]. Next, when the requirements are defined, the *task* that is to be explained must be clear, and the solution used for the task, meaning the type of explanandum. For explainability purposes the level of *model transparency* is the relevant point here.

Task XAI methods out-of-the-box usually only apply to a specific set of *task types* of the to-be-explained model, and *input data types*. For white-box methods that access model internals, additional constraints may hold for the *architecture* of the model (cf. portability aspect in [113]).

Task type Typical task categories are unsupervised clustering (clu), regression, classification (cls), detection (det), semantic segmentation which is pixel-wise classification, or instance segmentation. Many XAI methods targeting a question for classification, *e.g.* "Why this class?", can be extended to det, seg, and temporal resolution via snippeting of the new dimensions: "Why this class in this spatial/temporal snippet?". It must be noted that XAI methods working on classifiers often require access to a continuous classification score prediction instead of the final discrete classification. Such methods can also be used on regression tasks to answer questions about local trends, *i.e.* "Why does the prediction tend into this direction?". Examples of regression predictions are bounding box dimensions in object detection.

Examples RISE [78] (Randomized Input Sampling for Explanation) is a model-agnostic attribution analysis method specialized on image data. It produces heatmaps for visualization by randomly dimming super-pixels of an input image to find those which have the greatest influence on the local class confidence when deleted. The extension D-RISE [79] to object detection considers not a one-dimensional but the total prediction vector for change measurement. Other than these local image-bound explanation methods, surrogate models produced using inductive logic programming (ILP) [21] require the binary classification output of a model. ILP frameworks require background knowledge (logical theory) as input together with positive and negative examples. From this, a logic program

RISE [78]

D-RISE [79]

ILP [21]

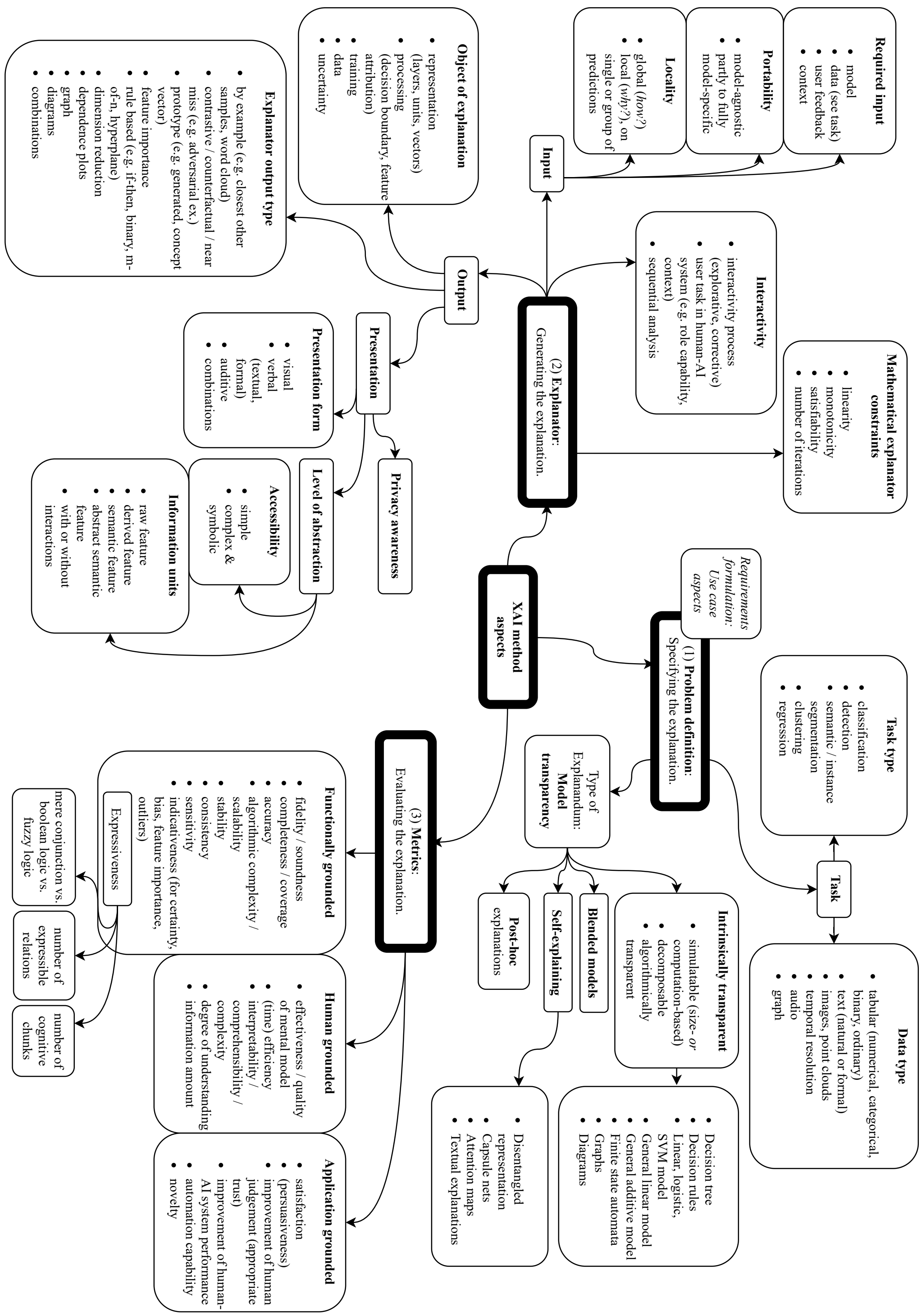

Fig. 2. Overview of the taxonomy presented in section 2

in the form of first-order rules is learned covering as many of the samples as possible. An example of an ILP surrogate model method is CA-ILP [83] (Concept Analysis for ILP): In order to explain parts of a convolutional image classifier with logical rules, they first learn global extractors for symbolic features which are then used for training an ILP surrogate model. Clustering tasks can often be explained by providing examples or prototypes of the final clusters, which will be discussed in subsection 2.2. CA-ILP [83]

Input data type Not every XAI method supports every input and output *signal type*, also called data type [38]. One input type is tabular (symbolic) data, which encompasses numerical, categorical, binary, and ordinary (ordered) data. Other symbolic input type are natural language or graphs, and non-symbolic types are images and point clouds (with or without temporal resolution), and audio.

Examples Typical examples for image explanations are methods producing heatmaps highlighting parts of the image that were relevant for (a part of) the decision. This highlighting of input snippets can also be applied to textual inputs where single words or sentence parts can be snippets. A prominent example of heatmapping both applicable to images and text inputs is the model-agnostic LIME [86] method (Local Interpretable Model-agnostic Explanations): It learns as local approximation a linear model on feature snippets of the input. For training, randomly selected snippets are removed. For textual inputs the snippets are words, for images they are super-pixels and blackened for removal. While LIME is suitable for image or textual input data, [38] provides a broad overview on model-agnostic XAI methods for tabular data. LIME [86]

Model transparency A model is considered to be transparent if its function is understandable without need for further explanation [4]. To obtain an explainable model, one can either *post-hoc* find a transparent surrogate model from which to derive the explanation (without changing the trained model), design the model to include *self-explanations* as additional output, or start with an *intrinsically transparent* model or *blended*, *i.e.* partly transparent, model from the beginning. Many examples for post-hoc methods are given later on, details on the other transparency types can be found below.

Intrisic transparency As introduced in [65], one can further differentiate between different levels of transparency: The model can directly be adapted as mental model by a human (*simulatable* [4]), or it can be split up into parts each of which is simulatable (*decomposable* [4]). Simulatability can either be measured based on the size of the model, or the needed length of computation. As a third category, *algorithmic transparency* is considered, which means the model can be mathematically investigated, *e.g.* the shape of the error surface is known.

Examples The following models are considered inherently transparent in the literature (cf. [70, Chap. 4], [38, Sec. 5], [75])

– Diagrams (cf. [46])

– Decision rules: This encompasses boolean rules as can be extracted from decision trees, or fuzzy or first-order logic rules. For further insights in inductive logic programming approaches to find the latter kind of rules see *e.g.* the recent survey [21].
– Decision trees
– Linear and logistic models
– Support vector machines
– General linear models (GLM): Here it is assumed that there is a linear relationship between the input features and the expected output value when this is transformed by a given transformation. For example, in logistic regression, the transformation is the logit. See *e.g.* [70, Sec. 4.3] for a basic introduction and further references.
– General additive models (GAM): It is assumed that the expected output value is the sum of transformed features. See the survey [15] for more details and further references. One concrete example of general additive models is the Additive Model Explainer [17]. They train predictors for a given set of features, and another small DNN predicting the additive weights for the feature predictors. They use this setup to learn a GAM surrogate models for a DNN, which also provides a prior to the weights: They should correspond to the sensitivity of the DNN with respect to the features.

Additive Model Explainer [17]

– Graphs (cf. [111])
– Finite state automata
– Simple clustering approaches, *e.g.* k-means clustering: The standard k-means clustering method [42] works with an intuitive model, simply consisting of k prototypes and a proximity measure, with inference associating new samples to the closest prototype representing a cluster. As long as the proximity measure is not too complex, this method can be regarded as an unsupervised inherently interpretable model.

k-means clustering [42]

Blended models Blended models consist partly of intrinsically transparent, symbolic models, that are integrated in sub-symbolic non-transparent ones. These kind of hybrid models are especially interesting for neuro-symbolic computing and similar fields combining symbolic with sub-symbolic models [13].

Logic Tensor Nets [26]

Examples An example of a blended model are Logic Tensor Networks [26]. Their idea is to use fuzzy logic to encode logical constraints on DNN outputs, with a DNN acting as fuzzy logic predicate. The framework in [26] allows additionally to learn semantic relations subject to symbolic fuzzy logic constraints. The relations are represented by simple linear models. Unsupervised deep learning can be made interpretable by approaches such as combining autoencoders with visualization approaches or by explaining choices of "neuralized" clustering methods [53] (*i.e.* clustering models translated to a DNN) with saliency maps. Enhancing an autoencoder was applied for example in the FoldingNet [112] architecture on point clouds. There, a folding-based decoder allows to view the reconstruction of point clouds, namely the warping from a 2D grid into the point

FoldingNet [112]

Neuralized clustering [53]

cloud surface. A saliency based solution can be produced by algorithms such as layer-wise relevance propagation which will be discussed in later examples.

Self-explaining models Self-explaining models provide additional outputs that explain the output of a single prediction. According to [35], there are three standard types of outputs of explanation generating models: *attention maps*, *disentangled representations*, and *textual or multi-modal explanations*.

Attention maps These are heatmaps that highlight relevant parts of a given single input for the respective output.
Examples The work in [56] adds an attention module to a DNN that is processed in parallel to, and later multiplied with, convolutional outputs. Furthermore, they suggest a clustering-based post-processing of the attention maps to highlight most meaningful parts.

Disentangled representations means that single or groups of dimensions in the intermediate output of the explanandum directly represent symbolic (also called semantic) concepts.
Examples One can by design force one layer of a DNN to exhibit a disentangled representation. One example are capsule networks [88], that structure the network not by neurons but into groups of neurons, the capsules, that characterize each one entity, *e.g.* an object or object part. The length of a capsule vector is interpreted as the probability that the corresponding object is present, while the rotation encodes properties of the object (*e.g.* rotation or color). Later capsules get as input the weighted sum of transformed previous capsule outputs, with the transformations learned and the weights obtained in an iterative routing process. A simpler disentanglement than alignment of semantic concepts with groups of neurons is alignment of single dimensions. This is done *e.g.* in the ReNN [107] architecture. They explicitly modularize their DNN to ensure semantically meaningful intermediate outputs. Other methods rather follow a post-hoc approach that fine-tunes a trained DNN towards more disentangled representations, like it is suggested for Semantic Bottleneck Networks [66]. These consist of the pretrained backbone of a DNN, proceeded by a layer in which each dimension corresponds to a semantic concept, called semantic bottleneck, and finalized by a newly trained front DNN part. During fine-tuning, first the connections from the backend to the semantic bottleneck are trained, then the parameters of the front DNN. Another interesting fine-tuning approach is that of concept whitening [18], which supplements batch-normalization layers with a linear transformation that learns to align semantic concepts with unit vectors of an activation space.

Capsule Nets [88]

ReNN [107]

Semantic Bottlenecks [66]

Concept Whitening [18]

Textual or multi-model explanations provide the explainee with a direct verbal or combined explanation that as part of the model output.
Examples An example are the explanations provided by [57] for the application of end-to-end steering control in autonomous driving. Their approach is two-fold: They add a custom layer that produces attention heatmaps similar to [56], and these are used by a second custom part to generate textual explanations of the decision which are (weakly) aligned with the model processing.

[57]

ProtoPNet [16]

ProtoPNet [16] for image classification provides visual examples rather than text. The network architecture is based on first selecting prototypical image patches, and then inserting a prototype layer that predicts similarity scores for patches of an instance with prototypes. These can then be used for explanation of the final result in the manner of "This is a sparrow as its beak looks like that of other sparrow examples". A truly multi-modal example is [43], which trains alongside a classifier a long-short term memory DNN (LSTM) to generate natural language justifications of the classification. The LSTM uses both the intermediate features and predictions of the image classifier, and is trained towards high class discriminativeness of the justifications. The explanations can optionally encompass bounding boxes for features that were important for the classification decision, making it multi-modal.

[43]

2.2 Explanator

The aspects of the explanator encompass *mathematical properties*, like linearity and monotonicity [14], requirements on the *input*, and properties of the *output* and the explanation generation, more precisely the *interactivity* of that process. Finally, we collect some *mathematical constraints* that can be desirable and verified on an explanator.

Input

Required Input The necessary inputs to the explanator may differ amongst methods [98]. While the explanandum, the *model* to explain, must usually provided to the explanator, many methods do also require valid *data* samples, or even *user feedback* (cf. section 2.2) or further situational *context* (cf. [24] for a more detailed definition of context).

Portability An important practical aspect for post-hoc explanations is whether or in how far the explanation method is dependent on access to internals of the explanandum model. This level of dependency is called portability, translucency, or transferability. In the following, we will not further differentiate between the strictness of requirements of model-specific methods. Transparent and self-explaining models are always model-specific, as the interpretability requires a special model type or model architecture (modification). Higher levels of dependency are:

model-agnostic also called *pedagogical* [113] or black-box means that only access to model input and output is required.
Examples A prominent example of model-agnostic methods is the previously discussed LIME [86] method for local approximation via a linear model. Another method to find feature importance weights without any access to model internals is SHAP [67] (SHapley Additive exPlanation). Their idea is to axiomatically ensure: local fidelity; features missing from the original input

SHAP [67]

have no effect; an increase of a weight also means an increased attribution of the feature to the final output; and uniqueness of the weights. Just as LIME, SHAP just requires a definition of “feature” or snippet on the input in order to be applicable.

model-specific also called *decompositional* [113] or white-box means that access is needed to the internal processing or architecture of the explanandum model, or even constraints apply.
Examples Methods relying on gradient or relevance information for generation of visual attention maps are strictly model-specific. A grandient-based method is Sensitivity Analysis [8]. They pick the vector representing the steepest ascend in the gradient tangential plane of a sample point. This method is independent of the type of input features, but can only analyse one one-dimensional output at once. Output-type-agnostic but dependent on a convolutional architecture and image inputs is Deconvnet [115] and its successors Backpropagation [95] and Guided Backpropagation [99]. They approximate a reconstruction of an input by defining inverses of pool and convolution operations, which allows to backpropagate the activation of single filters back to input image pixels (see [110] for a good overview). The idea of Backpropagation is generalized axiomatically by LRP [7] (Layer-wise Relevance Propagation): They require that the sum of linear relevance weights for each neuron in a layer should be constant throughout the layers (relevance is neither created nor extinguished from layer to layer). Methods that achieve this are *e.g.* Taylor decomposition or the back-propagation of relevance weighted by the forward-pass weights. The advancement PatternAttribution [58] fulfills the additional constraint to be sound on linear models.

Sensitivity Analysis [8]

Deconvnet [115]

Backprop [95]

Guided Backprop [99]

LRP [7]

PatternAttribution [58]

hybrid also called *eclectic* [113] or gray-box, means that the explanator only depends on access to parts of the model intermediate output, but not the full architecture.
Examples The rule extraction technique DeepRED [122] (Deep Rule Extraction with Decision tree induction) is an example of an eclectic method, so neither fully model-agnostic nor totally reliant on access to model internals. The approach conducts a backwards induction over the layer outputs of a DNN, between each two applying a decision tree extraction. While they enable rule extraction for arbitrarily deep DNNs, only small networks will result in rules of decent length for explanations.

DeepRED [122]

Explanation locality Literature differentiates between different ranges of validity of an explanation, respectively surrogate model. A surrogate model is valid in the ranges where high fidelity can be expected (see subsection 2.3). The range of input required by the explanator depends on the targeted validity range, so whether the input must representing a *local* or the *global* behavior of the explanandum. The general locality types are:

Local means the explanator is valid in a neighborhood of one or a group of given (valid) input samples. Local explanations tackle the question of *why* a given decision for one or a group of examples was made.

Examples Heatmapping methods are typical examples for local-only explanators, such as the discussed perturbation-based model-agnostic methods RISE [78], D-RISE [79], LIME [86], SHAP [67], as well as the model-specific sensitivity and backpropagation based methods LRP [7], PatternAttribution [58], Sensitivity Analysis [8], and Deconvnet and its successors [115,95,99].

Global means the explanator is valid in the complete (valid) input space. Other than the *why* of local explanations, global interpretability can also be described as answering *how* a decision is made.

Explanatory Graphs [116]

Examples A graph-based global explanator is generated by [116]. Their idea is that semantic concepts in an image usually consist of sub-objects to which they have a constant relative spatial relation (*e.g.* a face has a nose in the middle and two eyes next to each other), and that the localization of concepts should not only rely on high filter activation patterns, but also on their subpart arrangement. To achieve this, they translate the convolutional layers of a DNN into a tree of nodes (concepts), the *explanatory graph*. Each node belongs to one filter, is anchored at a fixed spatial position in the image, and represents a spatial arrangement of its child notes. The graph can also be used for local explanations via heatmaps: To localize a node in one input image, it is assigned the position closest to its anchor for which its filter activation is highest and for which the expected spatial relation to its children is best fulfilled. While most visualization based methods provide only local visualizations, a global, prototype-based, visual explanation is provided by Feature Visualizations [77]. The goal here is to visualize the function of a part of a DNN by finding prototypical input examples that strongly activate that part. These can be found via picking, search, or optimization. Other than visualizations, rule extraction methods usually only provide global approximations. An example is the well-known model-agnostic rule extractor VIA [102] (Validity Interval Analysis), which iteratively refines or generalizes pairs of input- and output-intervals. An example for getting from local to global explanations is SpRAy [62] (Spectral Relevance Analysis). They suggest to apply spectral clustering [105] to local feature attribution heatmaps of a data samples in order to find spuriously distinct global behavioral patterns. The heatmaps were generated via LRP [7].

Feature Visualization [77]

VIA [102]

SpRAy [62]

Output The output is characterized by several aspects: what is explained (the *object of explanation*), how it is explained (the actual *output type*), and how it is *presented*.

Object of explanation The object (or scope [70]) of an explanation describes which item of the development process should be explained. Items we identified in literature:

processing The objective is to understand the (symbolic) processing pipeline of the model, *i.e.* to answer parts of the question "How does the model work?".

This is the usual case for model-agnostic analysis methods. Types of processing to describe are *e.g.* the *decision boundary*, and *feature attribution* (or feature importance). Note that these are closely related, as highly important features usually locally point out the direction to the decision boundary. In case a symbolic explanator is targeted, one may need to first find a symbolic representation of input, output, or the model internal representation. Note that model-agnostic methods that do not investigate the input data usually target explanations of the model processing.

Examples Feature attribution methods encompass all the discussed attribution heatmapping methods (*e.g.* RISE [78], LIME [86], LRP [7]). LIME can be considered a corner case, as it both explains feature importance but also tries to approximate the decision boundary using a linear model on superpixels, which can itself serve directly as an explanation. A typical way to describe decision boundaries are decision trees or sets of rules, like extracted by the discussed VIA [102], and DeepRED [122]. Standard candidates for model-agnostic decision tree extraction are TREPAN [20] for M-of-N rules at the split points, and the C4.5 [82] decision tree generator for shallower but wider trees with interval-based splitting points. Concept tree [85] is a recent extension of TREPAN that adds automatic grouping of correlated features into the candidate concepts to use for the tree nodes. TREPAN [20] C4.5 [82] Concept Tree [85]

inner representation Machine learning models learn new representations of the input space, like the latent spaces representations found by DNNs. Explaining these inner representations answers “How does the model see the world?”. A more fine-grained differentiation considers whether *layers*, *units*, or *vectors* in the feature space are explained.

Examples

- units: One example of unit analysis is the discussed Feature Visualization [77]. In contrast to this unsupervised assignment of convolutional filters to prototypes, NetDissect [10] (Network Dissection) assigns filters to pre-defined semantic concepts in a supervised manner: For a filter, that semantic concept (color, texture, material, object, or object part) is selected for which the ground truth segmentation masks have the highest overlap with the upsampled filter’s activations. The authors also suggest that concepts that are less entangled, so less distributed over filters, are more interpretable, which is measurable with their filter-to-concept-alignment technique. NetDissect [10]
- vectors: Other than NetDissect, Net2Vec [31] also wants to assign concepts to their possibly entangled representations in the latent space. For a concept, they learn a linear 1×1-convolution on the output of a layer, which segments the concept in an image. The weight vector of the linear model for a concept can be understood as a prototypical representation (embedding) for that concept in the DNN intermediate output. They found that such embeddings behave like vectors in a word vector space: Concepts that are semantically similar feature embeddings with high cosine similarity. Similar to Net2Vec, TCAV [55] (Testing Concept Activation Vectors) also aims to find embeddings of NetDissect concepts. Net2Vec [31] TCAV [55]

They are interested in embeddings that are represented as a linear combination of convolutional filters, but in embedding vectors lying in the space of the complete layer output. In other words, they do not segment concepts but make an image-level classification whether the concept is present. These are found by using an SVM model instead of the 1×1-convolution. Additionally, they suggest to use partial derivatives along those concept vectors to find the local attribution of a semantic concept to a certain output. Other than the previous supervised methods, ACE [34] (Automatic Concept-based Explanations). does not learn a linear classifier but does an unsupervised clustering of concept candidates in the latent space. The cluster center then is selected as embedding vector. A super-pixeling approach together with outlier removal are used to obtain concept candidates.

ACE [34]

– layers: The works of [114] and IIN [28] (invertible interpretation networks) extend on the previous approaches and analyse a complete layer output space at once. For this, they find a subspace with a basis of concept embeddings, which allows an invertible transformation to a disentangled representation space. While IIN use invertible DNNs for the bijection of concept to latent space, [114] linear maps in their experiments. These approaches can be seen as a post-hoc version of the Semantic Bottleneck [66] architecture, only not replacing the complete later part of the model, but just learning connections from the bottleneck to the succeeding trained layer. [114] additionally introduces the notion of completeness of a set of concepts as the maximum performance of the model intercepted by the semantic bottleneck.

Concept completeness [114]

IIN [28]

development (during training) Some methods focus on assessing effects during training [70, Sec. 2.3]: "How does the model evolve during the training? What effects do new samples have?"

Examples One example is the work of [94], who inspect the model during training to investigate the role of depth for neural networks. Their findings indicate that depth actually is of computational benefit. An example which can be used to provide *e.g.* prototypical explanations are Influence Functions [59]. They gather the influence of training samples during the training to later assess the total impact of samples to the training. They also suggest to use this information as a proxy to estimate the influence of the samples to model decisions.

[94]

Influence Functions [59]

uncertainty [70] Capture and explain (*e.g.* visualize) the uncertainty of a prediction of the model. This encompasses the broad field of Bayesian deep learning [54] and uncertainty estimation [45]. It is argued in *e.g.* [80] for medical applications and in [69] for autonomous driving, why it is important to make the uncertainty of model decisions accessible to users.

data Pre-model interpretability [14] is the point where explainability touches the large research area of data analysis and feature mining.

Examples Typical examples for projecting high-dimensional data into easy-to-visualize 2D space are component analysis methods like PCA (Principal Component Analysis) [52]. A slightly more sophisticated approach is

PCA [52]

t-SNE [68]

t-SNE [68] (t-Distributed Stochastic Neighbor Embedding). In order to visualize a set of high-dimensional data points, they try to find a map from these points into a 2D or 3D space that is faithful on pairwise similarities. And also clustering methods can be used to generate prototype or example based explanations of typical features in the data. Examples here are k-means clustering [42] and the graph-based spectral clustering [105]. spectral clustering [105]

Output type The output type, also considered the actual explanator [38], describes the type of information presented to the explainee. Note that this ("what" is shown) is mostly independent of the presentation form ("how" it is shown). Typical types are:

by example instance *e.g.* closest other samples, word cloud

Examples The discussed ProtoPNet [16] is based on selecting and comparing relevant example snippets from the input image data.

contrastive / counterfactual / near miss including adversarial examples

Examples The perturbation-based feature importance heatmapping approach of RISE is extended in CEM [25] (Contrastive, Black-box Explanations Model). The do not only find positively contributing features, but also the features that must minimally be absent to not change the output. CEM [25]

prototype *e.g.* generated, concept vector

Examples A typical prototype generator is used in the discussed Feature Visualization [77] method: images are generated, *e.g.* via gradient descent, that represent the prototypical pattern for activating a filter. While this considers prototypical inputs, concept embeddings as collected in TCAV [55] and Net2Vec [31] describe prototypical activation patterns for a given semantic concept. The concept mining approach ACE [34] combines prototypes with examples: They search a concept embedding as prototype for an automatically collected set of example patches, that can be used to explain the prototype.

feature importance

Examples A lot of feature importance methods producing heatmaps have been discussed before (*e.g.* RISE [78], D-RISE [79], CEM [25], LIME [86], SHAP [67], LRP [7], PatternAttribution [58], Sensitivity Analysis [8], Deconvnet and successors [115,95,99]). One further example is the work in [32], which follows a perturbation-based approach. Similar to RISE, their idea is to find a minimal occlusion mask that if used to perturb the image (*e.g.* blur, noise, or blacken) maximally changes the outcome. To find the mask, backpropagation is used, making it a model-specific method. Some older but popular and simpler example methods are Grad-CAM [93] and its predecessor CAM [119] (Class Activation Mapping). While Deconvnet and its successors can only consider the feature importance with respect to intermediate outputs, (Grad-)CAM produces class-specific heatmaps, which are the weighted sum of the filter activation maps for one (usually the last) convolutional layer. For CAM, it is assumed the convolutional backend is finalized by a global average pooling layer that densely connects to the final classification [32] CAM [119] Grad-CAM [93] output. Here, the weights in the sum are the weights connecting the neurons of the global average pooling layer to the class outputs. For Grad-CAM, the weights in the sum are the averaged derivation of the class output by each activation map pixel. This is also used in the more recent [120], who do not apply Grad-CAM directly to the output but to each of a minimal set of projections from a convolutional intermediate output of a DNN that predict semantic concepts. Similar to Grad-CAM, SIDU [71] (Similarity Distance and Uniqueness) also adds up the filter-wise weighted activations of the last convolutional layer. The weights encompass a combination of a similarity score and a uniqueness score for the prediction output under each filter activation mask. The scores aim for high similarity of a masked predictions with the original one and low similarity to the other masked prediction, leading to masks capturing more complete and interesting object regions.

Concept-wise Grad-CAM [120]

SIDU [71]

rule based *e.g.* decision tree; or if-then, binary, m-of-n, or hyperplane rules (cf. [41])

Examples The mentioned exemplary rule-extraction methods DeepRED [122] and VIA [102], as well as decision tree extractors TREPAN [20], Concept Tree [85] and C4.5 [82] all provide global, rule-based output. For further rule extraction examples we refer the reader to the comprehensive surveys [41,108,6] on the topic, and the survey [109] for recurrent DNNs. An example of a local rule-extractor is the recent LIME-Aleph [84] approach, which generates a local explanation in the form of first-order logic rules. This is learned using inductive logic programming (ILP) [21] trained on the symbolic knowledge about a set of semantically similar examples. Due to the use of ILP, the approach is limited to tabular input data and classification outputs, but just as LIME it is model-agnostic. A similar approach is followed by NBDT [106] (Neural-Backed Decision Trees). They assume that the concept embeddings of super-categories are represented by the mean of their sub-category vectors (*e.g.* the mean of “cat” and “dog” should be “animal with four legs”). This is used to infer from bottom-to-top a decision tree where the nodes are super-categories and the leaves are the classification classes. At each node it is decided which of the sub-nodes best applies to the image. As embedding for a leaf concept (an output classes) they suggest to take the weights connecting the penultimate layer to a class output, and as similarity measure for the categories they use dot-product (cf. Net2Vec and TCAV).

LIME-Aleph [84]

NBDT [106]

dimension reduction *i.e.* sample points are projected to a sub-space

Examples Typical dimensionality reduction methods mentioned previously are PCA [52] and t-SNE [68].

dependence plots plot the effect of an input feature on the final output of a model.

Examples PDP [33] (Partial Dependency Plots, cf. [70, sec. 5.1]) calculate for one input feature and each vallue of this feature the expected model outcome averaged over the dataset. This results in a plot (for elach output) that indicates the global influence of the respective feature on the model. The local equivalent, ICE [36] (Individual Conditional Expectation, cf. [70,

PDP [33]

ICE [36]

sec. 5.2]) plots, obtain the PDP for generated data samples locally around a given sample.

graph

Examples The previously discussed Explanatory Graph [116] method provides amongst others a graph-based explanation output.

combinations

Presentation The presentation of information can be characterized by two categories of properties: the used *presentation form*, and the *level of abstraction* used to present available information. The presentation form simply summarizes the human sensory input channels utilized by the explanation, which can be: visual (the most common one including diagrams, graphs, and heatmaps), textual in either natural language or formal form, auditive, and combinations thereof. In the following the aspects influencing the level of abstraction are elaborated. These can be split up into (1) aspects of the smallest building blocks of the explanation, the *information units*, and (2) the *accessibility* or level of complexity of their combinations (the information units). Lastly, further filtering may be applied before finally presenting the explanation, including privacy filters.

Information units The basic units of the explanation, cognitive chunks [27], or information unit, may differ in the level or processing applied to them. The simplest form are unprocessed *raw features*, as used in explanations by example. *Derived features* capture some indirect information contained in the raw inputs, like super-pixels or attention heatmaps. These need not necessarily have a semantic meaning to the explainee, other than explicitly *semantic features*, *e.g.* concept activation vector attributions. The last type of information units are *abstract semantic features* not directly grounded in any input, *e.g.* prototypes. *Feature interactions* may occur as information units or be left unconsidered for the explanation.

Examples Some further notable examples of heatmapping methods for feature attribution are SmoothGrad [97] and Integrated Gradients [100]. One drawback of the methods described so far is that the considered loss surfaces that are linearly approximated tend to be "rough", *i.e.* exhibit significant variation in the point-wise values, gradients, and thus feature importance [89]. SmoothGrad [97] aims to mitigate this by averaging the gradient from random samples within a ball around the sample to investigate. Integrated gradients [100] do the averaging (to be precise: integration) along a path between two points in the input space. A technically similar approach but with a different goal is Integrated Hessians [51]. They intend not to grasp and visualize the sensitivity of the model for one feature (as derived feature), but their information units are interactions of features, *i.e.* how much the change of one feature changes the influence of the other on the output. This is done by having a look at the Hessian matrix, which is obtained by two subsequent integrated gradient calculations.

SmoothGrad [97]

Integrated Gradients [100]

Integrated Hessians [51]

Accessibility The accessibility, level of detail, or level of complexity describes how much intellectual effort the explainee has to bring up in order to understand the simulatable parts of the explanation. Thus, the perception of complexity heavily depends on the end-user, which is mirrored in the complexity metric discussed later in subsection 2.3. In general, one can differentiate between representations that are considered *simpler*, and such that are more *expressive but complex*. Because accessibility refers to the simulatable parts, this differs from the decomposable transparency level: For example, very large decision trees or very high-dimensional (general) linear models may be perceived as globally complex by the end-user. However, when looking at the simulatable parts of the explanator, like small groups of features or nodes, they are easy to grasp.

Examples Accessibility can indirectly be assessed by the complexity and expressivity of the explanation (see subsection 2.3). To give some examples: *Simple* presentations are *e.g.* linear models, general additive models, decision trees and Boolean decision rules, Bayesian models, or clusters of examples (cf. subsection 2.1; more *complex* are *e.g.* first-order or fuzzy logical decision rules.

Privacy awareness Sensible information like names may be contained in parts of the explanation, even though they are not necessary for understanding the actual decision. In such cases, an important point is privacy awareness [13]: Is sensible information removed if unnecessary, or properly anonymized if needed?

Interactivity The interaction of the user with the explanator may either be static, so the explainee is once presented with an explanation, or interactive, meaning an iterative process accepting user feedback as explanation input. Interactivity is characterized by the *interaction task* and the *explanation process*.

Interaction task The user can either inspect explanations or *correct* them. Inspecting takes place through *exploration* of different parts of one explanation or through consideration of various alternatives and complementing explanations, such as implemented in the *iNNvestigate* toolbox [3]. Besides, the user can be empowered within the human-AI partnership to provide corrective feedback to the system via an explanation interface, in order to adapt the explanator and thus the explanandum.

Examples State-of-the-art systems

CAIPI [101]

- enable the user to perform *corrections on labels* and to act upon wrong explanations through interactive machine learning (intML), such as implemented in the CAIPI approach [101],

EluciDebug [60]

- they allow for *re-weighting of features* for explanatory debugging, like the EluciDebug system [60],

Crayons [29]

- *adaption of features* as provided by Crayons [29], and

LearnWithME [91]

- correcting generated verbal explanations through user-defined constraints, such as implemented in the medical-decision support system LearnWithME [91].

Explanation process As mentioned above explanation usually takes place in an iterative fashion. Sequential analysis allows the user to query further information in an iterative manner and to understand the model and its decisions over time, in accordance with the user's capabilities and the given context.

Examples This includes combining different methods to create multi-modal explanations and involving the user into a dialogue, such as realized through a phrase-critic model as presented in [44].

Multi-modal explanations [44]

Mathematical Constraints Mathematical constraints encode some formal properties of the explanator that were found to be helpful for explanation receival. Constraints mentioned in literature are:

Linearity Considering a concrete proxy model as explanator output, linearity is often considered as a desirable form of simplicity [55,14,70].

Monotonicity Similar to linearity, one here again considers a concrete proxy model as output of the explanator. It is then considered a desirable level of simplicity if the dependency of that model's output on one input feature is monotonous.

Satisfiability This is the case if the explantor outputs readily allow application of formal methods like solvers.

Number of iterations While some XAI methods require a one-shot inference of the explanandum model (*e.g.* gradient-based methods), others require several iterations of queries to the explanandum. Since these might be costly or even restricted in some use cases a limited number of iterations needed by the explanator may be desirable in some cases. Such restrictions may arise from non-gameability [61] constraints on the explanandum model, *i.e.* the number of queries is restricted in order to guard against systematic optmization of outputs by users (*e.g.* searching for adversaries).

2.3 Metrics

By now, there is a considerable amount of metrics suggested to assess the quality of XAI methods with respect to different goals. This section details the types of metrics considered in literature. Following the original suggestion in [27], we categorize metrics by their level of human involvement required to measure them. For approaches to measure the below described metrics we refer the reader to [63]. They provide a good starting point with an in-depth analysis of metrics measurement for visual feature attribution methods.

Functionally Grounded Metrics Metrics are considered functionally grounded if the do not requiring any human feedback but instead measure formal properties of the explanator. This applies to the following metrics:

Fidelity or soundness [113], causality [13], or faithfulness [63], measures how accurately the behavior of the surrogate model used for the explanations conforms with that of the actual object of explanation. More simplification usually comes along with less fidelity, since corner cases are not captured anymore, also called the fidelity interpretability trade-off.

Completeness or coverage measures how large the validity range of an explanation is, so in which subset of the input space high fidelity can be expected. It can be seen as a generalization of fidelity to the distribution of fidelity.

Accuracy ignores the prediction quality of the original model and only considers the prediction quality of the surrogate model for the original task. This only applies to post-hoc explanations.

Algorithmic complexity and scalability measure the information theoretic complexity of the algorithm used to derive the explanator. This includes the time to convergence (to an acceptable solution), and is especially interesting for complex approximation schemes like rule extraction.

Stability or robustness [13] measures the change of explanator (output) given a change on the input samples. This corresponds to (adversarial) robustness of deep neural networks and a stable algorithm is usually also better comprehensible and desirable. Stability makes most sense for local methods.

Consistency measures the change of the explanator (output) given a change on the model to explain. The idea behind consistency is that functionally equivalent models should produce the same explanation. This assumption is important for model-agnostic approaches, while for model-specific ones a dependency on the model architecture may even be desirable. (*e.g.* for architecture visualization).

Sensitivity measures whether local explanations change if the model output changes strongly. The intuition behind this is that a strong change in the model output usually comes along with a change in the discrimination strategy of the model between the differing samples [63]. Such changes should be reflected in the explanations. Note that this may be in conflict with stability goals for regions in which the explanandum model behaves chaotically.

Indicativeness (or localization in case of visual feature importance maps [63]) means how well an explanation points out certain points of interest, *e.g.* by being sensitive to them and explicitly highlighting it for the explainee. Such points of interest considered in literature are certainty, bias, feature importance, and outliers (cf. [14]).

Expressiveness or the level of detail is interested in the expected information density felt by the user. It is closely related to the level of abstraction of the presentation. Several functionally grounded proxies were suggested to obtain comparable measures for expressivity:
 - the depth or amount of *added information*, also measured as the mean number of used information units per explanation
 - *number of relations* that can be expressed
 - the *expressiveness category* of used rules, namely mere conjunction, boolean logic, first-order logic, or fuzzy rules (cf. [113])

Human Grounded Metrics Other than functionally grounded metrics, human grounded metrics require human feedback on proxy tasks for their measurement. Often, proxy tasks are considered instead of the final application to avoid a need for expensive experts or application runtime (think of medical domains).

The goal of an explanation always is that the receiver of the explanation can build a *mental model* of (aspects of) the object of explanation. Human grounded metrics aim to measure some fundamental psychological properties of the XAI methods, namely quality of the *mental model*. The following are counted as such in literature:

Interpretability or comprehensibility or complexity measures how accurately the mental model approximates the explanator model. This measure mostly relies on subjective user feedback whether they "could make sense" of the presented information. It depends on background knowledge, biases, and cognition of the subject and can reveal use of vocabulary inappropriate to the user [35].

Effectiveness how accurately the mental model approximates the object of explanation. In other words, one is interested in how well a human can simulate the (aspects of interest of the) object after being presented with the explanations. Proxies for the effectiveness can be fidelity and accessibility [70, Sec. 2.4]. This may serve as a proxy for interpretability.

(Time) efficiency measures how time efficient an explanation is, *i.e.* how long it takes a user to build up a viable mental model. This is especially of interest in applications with a limited time frame for user reaction, like product recommendation systems [76] or automated driving applications [57].

Degree of understanding measures in interactive contexts the current status of understanding It helps to estimate the remaining time or measures needed to reach the desired extend of the explainee's mental model.

Information amount measures the total subjective amount of information conveyed by one explanation. Even though this may be measured on an information theoretic basis, it usually is subjective and thus requires human feedback. Functionally grounded related metrics are the complexity of the object of explanation, together with fidelity, and coverage. For example, more complex models have a tendency to contain more information, and thus require more complex explanations if to be approximated widely and accurately.

Application Grounded Metrics Other than human grounded metrics, application grounded ones work on human feedback for the final application. The following metrics are considered application grounded:

Satisfaction measures the direct content of the explainee with the system, so implicitly measures the benefit of explanations for the explanation system user.

Persuasiveness assesses the capability of the explanations to nudge an explainee into a certain direction. This is foremostly considered in recommendation systems [76], but has high importance when it comes to analysis tasks, where false positives and false negatives of the human-AI system are undesirable. In this context, a high persuasiveness may indicate a miscalibration of indicativeness.

Improvement of human judgement measures whether the explanation system user develops an appropriate level of trust in the decisions of the explained model. Correct decisions should be trusted more than wrong decisions *e.g.* because explanations of wrong decisions are illogical.

Improvement of human-AI system performance considers the end-to-end task to be achieved by all of explanandum, explainee, and explanator. This can *e.g.* be the diagnosis quality of doctors assisted by a recommendation system.

Automation capability gives an estimate on how much of the manual work conducted by the human in the human-AI system can be be automatized. Especially for local explanation techniques automation may be an important factor for feasibility if the number of samples a human needs to scan can be drastically reduced.

Novelty estimates the subjective degree of novelty of information provided to the explainee [61]. This is closely related to efficiency and satisfaction: Especially in exploratory use cases, high novelty can drastically increase efficiency (no repetitive work for the explainee), and keep satisfaction high (decrease the possibility for boredom of the explainee).

Table 5: Review of an exemplary selection of XAI techniques according to the defined taxonomy aspects (without fully transparent models). Abbreviations by column: *image data*=img, *point cloud data*=pcl; *Trans.*=transparency, *post-hoc*=p, *transparent*=t, *self-explaining*=s, *blended*=b; *processing*=p, *representation*=r, *development during training*=t *data*=d; *visual*=vis, *symbolic*=sym, *plot*=plt; *feature importance*=fi, *contrastive*=con, *prototypical*=proto, *decision tree*=tree, *distribution*=dist

Name	Cite	Task	Model-agnostic?	Transp.	Global?	Obj. Expl.	Form	Type
Self-explaining and blended models								
-	[43]	cls		s		p	sym/vis	rules/fi
-	[57]	any		s		p	sym/vis	rules/fi
ProtoPNet	[16]	cls,img		s		p/r	vis	proto/fi
Capsule Nets	[88]	cls		s		r	sym	fi
Semantic Bottlenecks, ReNN, Concept Whitening	[66,107,18]	any		s		r	sym	fi
Logic Tensor Nets	[26]	any		b	✓	p/r	sym	rule
FoldingNet	[112]	any,pcl		b		p	vis	fi/red
Neuralized clustering	[53]	any		b		p	vis	fi
Black-box heatmapping								
LIME, SHAP	[86,67]	cls	✓	p		p	vis	fi/con
RISE	[78]	cls,img	✓	p		p	vis	fi
D-RISE	[79]	det, img	✓	p		p	vis	fi
CEM	[25]	cls,img	✓	p		p	vis	fi/con
White-box heatmapping								

Table 5: Review of an exemplary selection of XAI techniques according to the defined taxonomy aspects (without fully transparent models). Abbreviations by column: *image data*=img, *point cloud data*=pcl; *Trans.*=transparency, *post-hoc*=p, *transparent*=t, *self-explaining*=s, *blended*=b; *processing*=p, *representation*=r, *development during training*=t *data*=d; *visual*=vis, *symbolic*=sym, *plot*=plt; *feature importance*=fi, *contrastive*=con, *prototypical*=proto, *decision tree*=tree, *distribution*=dist

Name	**Cite**	**Task**	**Model-agnostic?**	**Transp.**	**Global?**	**Obj. Expl.**	**Form**	**Type**
Sensitivity analysis	[8]	cls		p		p	vis	fi
Deconvnet, (Guided) Backprop.	[115,95,99]	img		p		p	vis	fi
CAM, Grad-CAM	[119,93]	cls,img		p		p	vis	fi
SIDU	[71]	cls,img		p		p	vis	fi
Concept-wise Grad-CAM	[120]	cls,img		p		p/r	vis	fi
SIDU	[71]	cls,img		p		p	vis	fi
LRP	[7]	cls		p		p	vis	fi
Pattern Attribution	[58]	cls		p		p	vis	fi
-	[32]	cls		p		p	vis	fi
SmoothGrad, Integrated Gradients	[97,100]	cls		p		p	vis	fi
Integrated Hessians	[51]	cls		p		p	vis	fi
Global representation analysis								
Feature Visualization	[77]	img		p	✓	r	vis	proto
NetDissect	[10]	img		p	✓	r	vis	proto/fi
Net2Vec	[31]	img		p	(✓)	r	vis	fi
TCAV	[55]	any		p	✓	r	vis	fi
ACE	[34]	any		p	✓	r	vis	fi
-	[114]	any		p	✓	r	vis	proto
IIN	[28]	any		p	(✓)	r	vis/sym	fi
Explanatory Graph	[116]	img		p	(✓)	p/r	vis	graph
Dependency plots								
PDP	[33]	any	✓	p		p	vis	plt
ICE	[36]	any	✓	p	✓	p	vis	plt
Rule extraction								
TREPAN, C4.5, Concept Tree	[20,82,85]	cls	✓	p	✓	p	sym	tree
VIA	[102]	cls	✓	p	✓	p	sym	rules
DeepRED	[122]	cls		p	✓	p	sym	rules
LIME-Aleph	[84]	cls	✓	p		p	sym	rules
CA-ILP	[83]	cls		p	✓	p	sym	rules
NBDT	[106]	cls		p	✓	p	sym	tree
Interactivity								
CAIPI	[101]	cls,img	✓	p		r	vis	fi/con
EluciDebug	[60]	cls	✓	p		r	vis	fi,plt
Crayons	[29]	cls,img	✓	t		p	vis	plt

Table 5: Review of an exemplary selection of XAI techniques according to the defined taxonomy aspects (without fully transparent models). Abbreviations by column: *image data*=img, *point cloud data*=pcl; *Trans.*=transparency, *post-hoc*=p, *transparent*=t, *self-explaining*=s, *blended*=b; *processing*=p, *representation*=r, *development during training*=t *data*=d; *visual*=vis, *symbolic*=sym, *plot*=plt; *feature importance*=fi, *contrastive*=con, *prototypical*=proto, *decision tree*=tree, *distribution*=dist

Name	Cite	Task	Model-agnostic?	Transp.	Global?	Obj. Expl.	Form	Type
LearnWithME	[91]	cls	✓	t	✓	p, r	sym	rules
Multi-modal phrase-critic model	[44]	cls,img		p	✓	p	vis,sym	plt,rules
Inspection of the training								
-	[94]	any		p	✓	t	vis	dist
Influence functions	[59]	cls		p	✓	t	vis	fi/dist
Data analysis methods								
t-SNE, PCA	[68,52]	any	✓	p	✓	d	vis	red
k-means, spectral clustering	[42,105]	any	✓	p	✓	d	vis	proto

3 Conclusion

In this paper, we combined existing taxonomies and surveys on the topic of XAI into an overarching taxonomy and added other highly relevant concepts from the literature. Starting from the definition of the problem of XAI, we developed our taxonomy based on three main parts: the task, the explainer and metrics. We defined each of these parts and explained them using numerous example concepts and example methods from the most relevant as well as the most recent research literature. To provide a guide on the methods, we classified the presented methods according to seven criteria that are significant in the literature. We asked about the task, the form of transparency, whether the method is model-agnostic or model-specific, whether it generates global or local explanations, what the object of explanation is, in what form explanations are presented and the type of explanation. In our taxonomy, we highlighted that beyond the presented parts (task, explainer and metric), there are also other, use case specific, aspects to consider when developing, applying, and evaluating XAI to account for different stakeholders and their context. To date, there is no article in the current research literature that unifies taxonomies, illustrates them through a variety of methods, and also serves as a starting point for use case driven research.

References

1. Adadi, A., Berrada, M.: Peeking inside the black-box: A survey on explainable artificial intelligence (xai). IEEE Access **6**, 52138–52160 (2018)
2. Adadi, A., Berrada, M.: Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). In: IEEE Access. vol. 6, pp. 52138–52160 (2018). https://doi.org/10.1109/ACCESS.2018.2870052, https://ieeexplore.ieee.org/document/8466590/

3. Alber, M., Lapuschkin, S., Seegerer, P., Hägele, M., Schütt, K.T., Montavon, G., Samek, W., Müller, K.R., Dähne, S., Kindermans, P.J.: iNNvestigate Neural Networks! Journal of Machine Learning Research **20**(93), 1–8 (2019), http://jmlr.org/papers/v20/18-540.html
4. Arrieta, A.B., Rodríguez, N.D., Ser, J.D., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion **58**, 82–115 (2020). https://doi.org/10.1016/j.inffus.2019.12.012, http://www.sciencedirect.com/science/article/pii/S1566253519308103
5. Arya, V., Bellamy, R.K.E., Chen, P.Y., Dhurandhar, A., Hind, M., Hoffman, S.C., Houde, S., Liao, Q.V., Luss, R., Mojsilovic, A., Mourad, S., Pedemonte, P., Raghavendra, R., Richards, J.T., Sattigeri, P., Shanmugam, K., Singh, M., Varshney, K.R., Wei, D., Zhang, Y.: One explanation does not fit all: A toolkit and taxonomy of AI explainability techniques. CoRR **abs/1909.03012** (2019), http://arxiv.org/abs/1909.03012
6. Augasta, M.G., Kathirvalavakumar, T.: Rule extraction from neural networks — A comparative study. In: Proc. 2012 Int. Conf. Pattern Recognition, Informatics and Medical Engineering. pp. 404–408 (Mar 2012). https://doi.org/10.1109/ICPRIME.2012.6208380, https://ieeexplore.ieee.org/document/6208380
7. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLOS ONE **10**(7), e0130140 (Jul 2015). https://doi.org/10.1371/journal.pone.0130140, https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0130140
8. Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., Müller, K.R.: How to explain individual classification decisions. Journal of Machine Learning Research **11**, 1803–1831 (Aug 2010), http://portal.acm.org/citation.cfm?id=1859912
9. Baniecki, H., Biecek, P.: The Grammar of Interactive Explanatory Model Analysis. arXiv:2005.00497 [cs, stat] (Sep 2020), http://arxiv.org/abs/2005.00497, arXiv: 2005.00497
10. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: Quantifying interpretability of deep visual representations. In: Proc. 2017 IEEE Conf. Comput. Vision and Pattern Recognition. pp. 3319–3327. IEEE Computer Society (2017). https://doi.org/10.1109/CVPR.2017.354, http://arxiv.org/abs/1704.05796
11. Benchekroun, O., Rahimi, A., Zhang, Q., Kodliuk, T.: The Need for Standardized Explainability. arXiv:2010.11273 [cs] (Oct 2020), http://arxiv.org/abs/2010.11273, arXiv: 2010.11273
12. Bruckert, S., Finzel, B., Schmid, U.: The next generation of medical decision support: A roadmap toward transparent expert companions. Frontiers in Artificial Intelligence **3**, 75 (2020)
13. Calegari, R., Ciatto, G., Omicini, A.: On the integration of symbolic and subsymbolic techniques for XAI: A survey. Intelligenza Artificiale **14**(1), 7–32 (Jan 2020). https://doi.org/10.3233/IA-190036, https://content.iospress.com/articles/intelligenza-artificiale/ia190036
14. Carvalho, D.V., Pereira, E.M., Cardoso, J.S.: Machine learning interpretability: A survey on methods and metrics. Electronics **8**(8), 832 (Aug 2019).

https://doi.org/10.3390/electronics8080832, https://www.mdpi.com/2079-9292/8/8/832
15. Chang, C.H., Tan, S., Lengerich, B., Goldenberg, A., Caruana, R.: How interpretable and trustworthy are GAMs? CoRR **abs/2006.06466** (Jun 2020), https://arxiv.org/abs/2006.06466
16. Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., Su, J.: This looks like that: Deep learning for interpretable image recognition. In: Advances in Neural Information Processing Systems 32. vol. 32, pp. 8928–8939 (2019), https://proceedings.neurips.cc/paper/2019/hash/adf7ee2dcf142b0e11888e72b43fcb75-Abstract.html
17. Chen, R., Chen, H., Huang, G., Ren, J., Zhang, Q.: Explaining neural networks semantically and quantitatively. In: Proc. 2019 IEEE/CVF International Conference on Computer Vision. pp. 9186–9195. IEEE (Oct 2019). https://doi.org/10.1109/ICCV.2019.00928, https://ieeexplore.ieee.org/document/9008520/
18. Chen, Z., Bei, Y., Rudin, C.: Concept whitening for interpretable image recognition. CoRR **abs/2002.01650** (Feb 2020), https://arxiv.org/abs/2002.01650
19. Chromik, M., Schuessler, M.: A taxonomy for human subject evaluation of black-box explanations in XAI. In: Proc. Workshop Explainable Smart Systems for Algorithmic Transparency in Emerging Technologies. vol. Vol-2582, p. 7. CEUR-WS.org (2020)
20. Craven, M.W., Shavlik, J.W.: Extracting tree-structured representations of trained networks. In: Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, USA, November 27-30, 1995. pp. 24–30. MIT Press (1995), http://papers.nips.cc/paper/1152-extracting-tree-structured-representations-of-trained-networks
21. Cropper, A., Dumancic, S., Muggleton, S.H.: Turning 30: New ideas in inductive logic programming. CoRR **abs/2002.11002** (2020), https://arxiv.org/abs/2002.11002
22. Danilevsky, M., Qian, K., Aharonov, R., Katsis, Y., Kawas, B., Sen, P.: A Survey of the State of Explainable AI for Natural Language Processing. arXiv:2010.00711 [cs] (Oct 2020), http://arxiv.org/abs/2010.00711
23. Das, A., Rad, P.: Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey. arXiv:2006.11371 [cs] (Jun 2020), http://arxiv.org/abs/2006.11371, arXiv: 2006.11371
24. Dey, A.K.: Understanding and using context. Personal and Ubiquitous Computing **5**(1), 4–7 (Jan 2001). https://doi.org/10.1007/s007790170019, https://doi.org/10.1007/s007790170019
25. Dhurandhar, A., Chen, P.Y., Luss, R., Tu, C.C., Ting, P., Shanmugam, K., Das, P.: Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In: Advances in Neural Information Processing Systems 31. pp. 592–603. Curran Associates, Inc. (2018), http://papers.nips.cc/paper/7340-explanations-based-on-the-missing-towards-contrastive-explanations-with-pertinent-negatives.pdf
26. Donadello, I., Serafini, L., d'Avila Garcez, A.S.: Logic tensor networks for semantic image interpretation. In: Proc. 26th Int. Joint Conf. Artificial Intelligence. pp. 1596–1602. ijcai.org (2017). https://doi.org/10.24963/ijcai.2017/221, https://www.ijcai.org/Proceedings/2017/221
27. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. CoRR **abs/1702.08608** (Feb 2017), http://adsabs.harvard.edu/abs/2017arXiv170208608D

28. Esser, P., Rombach, R., Ommer, B.: A disentangling invertible interpretation network for explaining latent representations. In: Proc. 2020 IEEE Conf. Comput. Vision and Pattern Recognition. pp. 9220–9229. IEEE (Jun 2020). https://doi.org/10.1109/CVPR42600.2020.00924, https://ieeexplore.ieee.org/document/9157170/
29. Fails, J.A., Olsen Jr, D.R.: Interactive machine learning. In: Proceedings of the 8th international conference on Intelligent user interfaces. pp. 39–45 (2003)
30. Ferreira, J.J., Monteiro, M.S.: What Are People Doing About XAI User Experience? A Survey on AI Explainability Research and Practice. In: Marcus, A., Rosenzweig, E. (eds.) Design, User Experience, and Usability. Design for Contemporary Interactive Environments. pp. 56–73. Lecture Notes in Computer Science, Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-49760-6_4
31. Fong, R., Vedaldi, A.: Net2Vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In: Proc. 2018 IEEE Conf. Comput. Vision and Pattern Recognition. pp. 8730–8738. IEEE Computer Society (2018). https://doi.org/10.1109/CVPR.2018.00910, http://openaccess.thecvf.com/content_cvpr_2018/html/Fong_Net2Vec_Quantifying_and_CVPR_2018_paper.html
32. Fong, R.C., Vedaldi, A.: Interpretable explanations of black boxes by meaningful perturbation. In: Proc. 2017 IEEE Intern. Conf. on Comput. Vision. pp. 3449–3457. IEEE Computer Society (2017). https://doi.org/10.1109/ICCV.2017.371, http://arxiv.org/abs/1704.03296
33. Friedman, J.H.: Greedy function approximation: A gradient boosting machine. The Annals of Statistics **29**(5), 1189–1232 (Oct 2001). https://doi.org/10.1214/aos/1013203451, https://projecteuclid.org/journals/annals-of-statistics/volume-29/issue-5/Greedy-function-approximation-A-gradient-boosting-machine/10.1214/aos/1013203451.full
34. Ghorbani, A., Wexler, J., Zou, J.Y., Kim, B.: Towards automatic concept-based explanations. In: Advances in Neural Information Processing Systems 32. pp. 9273–9282 (2019), http://papers.nips.cc/paper/9126-towards-automatic-concept-based-explanations
35. Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L.: Explaining explanations: An overview of interpretability of machine learning. In: Proc. 5th IEEE Int. Conf. Data Science and Advanced Analytics. pp. 80–89. IEEE (2018). https://doi.org/10.1109/DSAA.2018.00018, http://arxiv.org/abs/1806.00069
36. Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E.: Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation. Journal of Computational and Graphical Statistics **24**(1), 44–65 (Jan 2015). https://doi.org/10.1080/10618600.2014.907095, https://doi.org/10.1080/10618600.2014.907095
37. Goodman, B., Flaxman, S.: European union regulations on algorithmic decision-making and a “right to explanation”. AI Magazine **38**(3), 50–57 (Oct 2017). https://doi.org/10.1609/aimag.v38i3.2741, https://ojs.aaai.org/index.php/aimagazine/article/view/2741
38. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comput. Surv. **51**(5), 93:1–93:42 (Aug 2018). https://doi.org/10.1145/3236009
39. Gunning, D.: Explainable artificial intelligence (xai). Defense Advanced Research Projects Agency (DARPA), nd Web **2**(2) (2017)

40. Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., Yang, G.Z.: XAI—Explainable artificial intelligence. Science Robotics **4**(37) (Dec 2019). https://doi.org/10.1126/scirobotics.aay7120, https://robotics.sciencemag.org/content/4/37/eaay7120
41. Hailesilassie, T.: Rule extraction algorithm for deep neural networks: A review. CoRR **abs/1610.05267** (2016), http://arxiv.org/abs/1610.05267
42. Hartigan, J.A., Wong, M.A.: Algorithm AS 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics) **28**(1), 100–108 (1979). https://doi.org/10.2307/2346830, https://www.jstor.org/stable/2346830
43. Hendricks, L.A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., Darrell, T.: Generating visual explanations. In: Computer Vision – ECCV 2016. pp. 3–19. Lecture Notes in Computer Science, Springer International Publishing (2016). https://doi.org/10.1007/978-3-319-46493-0_1
44. Hendricks, L.A., Hu, R., Darrell, T., Akata, Z.: Grounding visual explanations. In: Proceedings of the European Conference on Computer Vision (ECCV) (September 2018)
45. Henne, M., Schwaiger, A., Roscher, K., Weiss, G.: Benchmarking uncertainty estimation methods for deep learning with safety-related metrics. In: Proc. Workshop Artificial Intelligence Safety. CEUR Workshop Proceedings, vol. 2560, pp. 83–90. CEUR-WS.org (2020), http://ceur-ws.org/Vol-2560/paper35.pdf
46. Heuillet, A., Couthouis, F., Díaz-Rodríguez, N.: Explainability in deep reinforcement learning. Knowledge-Based Systems **214**, 106685 (Feb 2021). https://doi.org/10.1016/j.knosys.2020.106685, http://arxiv.org/abs/2008.06693
47. Holzinger, A., Biemann, C., Pattichis, C.S., Kell, D.B.: What do we need to build explainable AI systems for the medical domain? arXiv:1712.09923 [cs, stat] (Dec 2017), http://arxiv.org/abs/1712.09923, arXiv: 1712.09923
48. Islam, S.R., Eberle, W., Ghafoor, S.K., Ahmed, M.: Explainable artificial intelligence approaches: A survey. arXiv:2101.09429 [cs] (Jan 2021), http://arxiv.org/abs/2101.09429
49. ISO/TC 22 Road vehicles: ISO/TR 4804:2020: Road Vehicles — Safety and Cybersecurity for Automated Driving Systems — Design, Verification and Validation. International Organization for Standardization, first edn. (Dec 2020), https://www.iso.org/standard/80363.html
50. ISO/TC 22/SC 32: ISO 26262-6:2018(En): Road Vehicles — Functional Safety — Part 6: Product Development at the Software Level, ISO 26262:2018(En), vol. 6. International Organization for Standardization, second edn. (Dec 2018), https://www.iso.org/standard/68388.html
51. Janizek, J.D., Sturmfels, P., Lee, S.I.: Explaining explanations: Axiomatic feature interactions for deep networks. CoRR **abs/2002.04138** (2020), https://arxiv.org/abs/2002.04138
52. Jolliffe, I.T.: Principal Component Analysis. Springer Series in Statistics, Springer-Verlag, second edn. (2002). https://doi.org/10.1007/b98835, https://www.springer.com/de/book/9780387954424
53. Kauffmann, J., Esders, M., Montavon, G., Samek, W., Müller, K.R.: From Clustering to Cluster Explanations via Neural Networks. arXiv:1906.07633 [cs, stat] (Jun 2019), http://arxiv.org/abs/1906.07633, arXiv: 1906.07633
54. Kendall, A., Gal, Y.: What uncertainties do we need in Bayesian deep learning for computer vision? In: Advances in Neural Information Processing Systems 30. pp. 5580–5590 (2017), http://papers.nips.cc/paper/7141-what-uncertainties-do-we-need-in-bayesian-deep-learning-for-computer-vision

55. Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., Sayres, R.: Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In: Proc. 35th Int. Conf. Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 2668–2677. PMLR (Jul 2018), http://proceedings.mlr.press/v80/kim18d.html
56. Kim, J., Canny, J.F.: Interpretable learning for self-driving cars by visualizing causal attention. In: Proc. 2017 IEEE Int. Conf. Comput. Vision. pp. 2961–2969. IEEE Computer Society (2017). https://doi.org/10.1109/ICCV.2017.320, http://arxiv.org/abs/1703.10631
57. Kim, J., Rohrbach, A., Darrell, T., Canny, J.F., Akata, Z.: Textual explanations for self-driving vehicles. In: Proc. 15th European Conf. Comput. Vision, Part II. Lecture Notes in Computer Science, vol. 11206, pp. 577–593. Springer (2018). https://doi.org/10.1007/978-3-030-01216-8_35
58. Kindermans, P.J., Schütt, K.T., Alber, M., Müller, K.R., Erhan, D., Kim, B., Dähne, S.: Learning how to explain neural networks: PatternNet and PatternAttribution. In: Proc. 6th Int. Conf. on Learning Representations (Feb 2018), https://openreview.net/forum?id=Hkn7CBaTW
59. Koh, P.W., Liang, P.: Understanding Black-box Predictions via Influence Functions. In: Proc. 34th Int. Conf. Machine Learning. pp. 1885–1894. PMLR (Jul 2017), http://proceedings.mlr.press/v70/koh17a.html
60. Kulesza, T., Stumpf, S., Burnett, M., Wong, W.K., Riche, Y., Moore, T., Oberst, I., Shinsel, A., McIntosh, K.: Explanatory debugging: Supporting end-user debugging of machine-learned programs. In: 2010 IEEE Symposium on Visual Languages and Human-Centric Computing. pp. 41–48. IEEE (2010)
61. Langer, M., Oster, D., Speith, T., Hermanns, H., Kästner, L., Schmidt, E., Sesing, A., Baum, K.: What Do We Want From Explainable Artificial Intelligence (XAI)? – A Stakeholder Perspective on XAI and a Conceptual Model Guiding Interdisciplinary XAI Research. Artificial Intelligence p. 103473 (Feb 2021). https://doi.org/10.1016/j.artint.2021.103473, http://arxiv.org/abs/2102.07817, arXiv: 2102.07817
62. Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., Müller, K.R.: Unmasking Clever Hans predictors and assessing what machines really learn. Nature Communications **10**(1), 1096 (Mar 2019). https://doi.org/10.1038/s41467-019-08987-4, https://www.nature.com/articles/s41467-019-08987-4
63. Li, X.H., Shi, Y., Li, H., Bai, W., Song, Y., Cao, C.C., Chen, L.: Quantitative Evaluations on Saliency Methods: An Experimental Study. arXiv:2012.15616 [cs] (Dec 2020), http://arxiv.org/abs/2012.15616, arXiv: 2012.15616
64. Linardatos, P., Papastefanopoulos, V., Kotsiantis, S.: Explainable AI: A review of machine learning interpretability methods. Entropy **23**(1), 18 (Jan 2021). https://doi.org/10.3390/e23010018, https://www.mdpi.com/1099-4300/23/1/18
65. Lipton, Z.C.: The mythos of model interpretability. Queue **16**(3), 31–57 (Jun 2018). https://doi.org/10.1145/3236386.3241340, http://arxiv.org/abs/1606.03490
66. Losch, M., Fritz, M., Schiele, B.: Interpretability beyond classification output: Semantic bottleneck networks. In: Proc. 3rd ACM Computer Science in Cars Symp. Extended Abstracts (Oct 2019), https://arxiv.org/pdf/1907.10882.pdf
67. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems 30. pp. 4765–4774. Curran Associates, Inc. (2017), http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

68. van der Maaten, L., Hinton, G.: Visualizing Data using t-SNE. Journal of Machine Learning Research **9**(86), 2579–2605 (2008), http://jmlr.org/papers/v9/vandermaaten08a.html
69. McAllister, R., Gal, Y., Kendall, A., van der Wilk, M., Shah, A., Cipolla, R., Weller, A.: Concrete problems for autonomous vehicle safety: Advantages of Bayesian deep learning. In: Proc. 26th Int. Joint Conf. Artificial Intelligence. pp. 4745–4753 (2017), https://doi.org/10.24963/ijcai.2017/661
70. Molnar, C.: Interpretable Machine Learning. Lulu.com (2020), https://christophm.github.io/interpretable-ml-book/
71. Muddamsetty, S.M., Jahromi, M.N.S., Ciontos, A.E., Fenoy, L.M., Moeslund, T.B.: Introducing and assessing the explainable AI (XAI) method: SIDU. arXiv:2101.10710 [cs] (Jan 2021), http://arxiv.org/abs/2101.10710
72. Mueller, S.T., Hoffman, R.R., Clancey, W., Emrey, A., Klein, G.: Explanation in human-AI systems: A literature meta-review, synopsis of key ideas and publications, and bibliography for explainable AI. arXiv:1902.01876 [cs] (Feb 2019), https://arxiv.org/abs/1902.01876v1
73. Mueller, S.T., Veinott, E.S., Hoffman, R.R., Klein, G., Alam, L., Mamun, T., Clancey, W.J.: Principles of explanation in human-AI systems. CoRR **abs/2102.04972** (Feb 2021), https://arxiv.org/abs/2102.04972
74. Muggleton, S.H., Schmid, U., Zeller, C., Tamaddoni-Nezhad, A., Besold, T.: Ultra-strong machine learning: comprehensibility of programs learned with ilp. Machine Learning **107**(7), 1119–1140 (2018)
75. Nori, H., Jenkins, S., Koch, P., Caruana, R.: InterpretML: A unified framework for machine learning interpretability. CoRR **abs/1909.09223** (Sep 2019), http://arxiv.org/abs/1909.09223
76. Nunes, I., Jannach, D.: A systematic review and taxonomy of explanations in decision support and recommender systems. User Modeling and User-Adapted Interaction **27**(3-5), 393–444 (Dec 2017). https://doi.org/10.1007/s11257-017-9195-0, https://doi.org/10.1007/s11257-017-9195-0
77. Olah, C., Mordvintsev, A., Schubert, L.: Feature visualization. Distill **2**(11), e7 (Nov 2017). https://doi.org/10.23915/distill.00007, https://distill.pub/2017/feature-visualization
78. Petsiuk, V., Das, A., Saenko, K.: RISE: Randomized input sampling for explanation of black-box models. In: Proc. British Machine Vision Conf. p. 151. BMVA Press (2018), http://bmvc2018.org/contents/papers/1064.pdf
79. Petsiuk, V., Jain, R., Manjunatha, V., Morariu, V.I., Mehra, A., Ordonez, V., Saenko, K.: Black-box explanation of object detectors via saliency maps. arXiv:2006.03204 [cs] (Jun 2020), http://arxiv.org/abs/2006.03204
80. Pocevičiūtė, M., Eilertsen, G., Lundström, C.: Survey of XAI in digital pathology. Lecture Notes in Computer Science **2020**, 56–88 (2020). https://doi.org/10.1007/978-3-030-50402-1_4, http://arxiv.org/abs/2008.06353
81. Puiutta, E., Veith, E.M.S.P.: Explainable reinforcement learning: A survey. In: Machine Learning and Knowledge Extraction - 4th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2020, Dublin, Ireland, August 25-28, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12279, pp. 77–95. Springer (2020). https://doi.org/10.1007/978-3-030-57321-8_5, https://doi.org/10.1007/978-3-030-57321-8_5
82. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Series in Machine Learning, Morgan Kaufmann (1993), https://kupdf.net/download/j-ross-quinlan-c4-5-programs-for-machine-learning-1993_5b095daee2b6f5024deefc30_pdf

83. Rabold, J., Schwalbe, G., Schmid, U.: Expressive explanations of DNNs by combining concept analysis with ILP. In: KI 2020: Advances in Artificial Intelligence. pp. 148–162. Lecture Notes in Computer Science, Springer International Publishing (2020). https://doi.org/10.1007/978-3-030-58285-2_11
84. Rabold, J., Siebers, M., Schmid, U.: Explaining black-box classifiers with ILP – empowering LIME with Aleph to approximate non-linear decisions with relational rules. In: Proc. Int. Conf. Inductive Logic Programming. pp. 105–117. Lecture Notes in Computer Science, Springer International Publishing (2018). https://doi.org/10.1007/978-3-319-99960-9_7, https://link.springer.com/chapter/10.1007/978-3-319-99960-9_7
85. Renard, X., Woloszko, N., Aigrain, J., Detyniecki, M.: Concept tree: High-level representation of variables for more interpretable surrogate decision trees. In: Proc. 2019 ICML Workshop Human in the Loop Learning. arXiv.org (2019), http://arxiv.org/abs/1906.01297
86. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should I trust you?": Explaining the predictions of any classifier. In: Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining. pp. 1135–1144. KDD '16, ACM (2016), https://arxiv.org/abs/1602.04938
87. Rieger, I., Kollmann, R., Finzel, B., Seuss, D., Schmid, U.: Verifying deep learning-based decisions for facial expression recognition (accepted). In: Proceedings of the ESANN Conference 2020 (2020)
88. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Advances in Neural Information Processing Systems 30. pp. 3856–3866. Curran Associates, Inc. (2017), http://papers.nips.cc/paper/6975-dynamic-routing-between-capsules
89. Samek, W., Montavon, G., Lapuschkin, S., Anders, C.J., Müller, K.R.: Toward interpretable machine learning: Transparent deep neural networks and beyond. CoRR **abs/2003.07631** (2020), https://arxiv.org/abs/2003.07631
90. Samek, W., Müller, K.R.: Towards explainable artificial intelligence. In: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Lecture Notes in Computer Science, vol. 11700, pp. 5–22. Springer (2019). https://doi.org/10.1007/978-3-030-28954-6_1, https://doi.org/10.1007/978-3-030-28954-6_1
91. Schmid, U., Finzel, B.: Mutual explanations for cooperative decision making in medicine. KI-Künstliche Intelligenz pp. 1–7 (2020)
92. Schmid, U., Zeller, C., Besold, T., Tamaddoni-Nezhad, A., Muggleton, S.: How does predicate invention affect human comprehensibility? In: International Conference on Inductive Logic Programming. pp. 52–67. Springer (2016)
93. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In: Proc. 2017 IEEE Int. Conf. Computer Vision. pp. 618–626. IEEE (Oct 2017). https://doi.org/10.1109/ICCV.2017.74, http://ieeexplore.ieee.org/document/8237336/
94. Shwartz-Ziv, R., Tishby, N.: Opening the Black Box of Deep Neural Networks via Information. CoRR **abs/1703.00810** (2017), http://arxiv.org/abs/1703.00810
95. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. In: 2nd Intern. Conf. Learning Representations, Workshop Track Proceedings. vol. abs/1312.6034. CoRR (2014), http://arxiv.org/abs/1312.6034

96. Singh, A., Sengupta, S., Lakshminarayanan, V.: Explainable Deep Learning Models in Medical Image Analysis. Journal of Imaging **6**(6), 52 (Jun 2020). https://doi.org/10.3390/jimaging6060052, https://www.mdpi.com/2313-433X/6/6/52
97. Smilkov, D., Thorat, N., Kim, B., Viégas, F.B., Wattenberg, M.: SmoothGrad: Removing noise by adding noise. CoRR **abs/1706.03825** (2017), http://arxiv.org/abs/1706.03825
98. Spinner, T., Schlegel, U., Schafer, H., El-Assady, M.: explAIner: A visual analytics framework for interactive and explainable machine learning. IEEE Transactions on Visualization and Computer Graphics **26**, 1064–1074 (2020). https://doi.org/10.1109/TVCG.2019.2934629, https://arxiv.org/pdf/1908.00087.pdf
99. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.A.: Striving for simplicity: The all convolutional net. In: 3rd Intern. Conf. Learning Representations, ICLR 2015, Workshop Track Proceedings. vol. abs/1412.6806. CoRR (2015), http://arxiv.org/abs/1412.6806
100. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: Proc. 34th Int. Conf. Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 3319–3328. PMLR (2017), http://proceedings.mlr.press/v70/sundararajan17a.html
101. Teso, S., Kersting, K.: Explanatory interactive machine learning. In: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society. pp. 239–245 (2019)
102. Thrun, S.: Extracting rules from artificial neural networks with distributed representations. In: Advances in Neural Information Processing Systems 7. pp. 505–512. MIT Press (1995), http://papers.nips.cc/paper/924-extracting-rules-from-artificial-neural-networks-with-distributed-representations.pdf
103. Tjoa, E., Guan, C.: A survey on explainable artificial intelligence (XAI): Toward medical XAI. IEEE Transactions on Neural Networks and Learning Systems pp. 1–21 (2020). https://doi.org/10.1109/TNNLS.2020.3027314, https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9233366
104. Vilone, G., Longo, L.: Explainable artificial intelligence: A systematic review. CoRR **abs/2006.00093** (2020), https://arxiv.org/abs/2006.00093
105. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing **17**(4), 395–416 (Dec 2007). https://doi.org/10.1007/s11222-007-9033-z, http://link.springer.com/10.1007/s11222-007-9033-z
106. Wan, A., Dunlap, L., Ho, D., Yin, J., Lee, S., Jin, H., Petryk, S., Bargal, S.A., Gonzalez, J.E.: NBDT: Neural-backed decision trees. arXiv:2004.00221 [cs] (Jun 2020), http://arxiv.org/abs/2004.00221
107. Wang, H.: ReNN: Rule-embedded neural networks. In: Proc. 24th Int. Conf. Pattern Recognition. pp. 824–829. IEEE Computer Society (2018). https://doi.org/10.1109/ICPR.2018.8545379, http://arxiv.org/abs/1801.09856
108. Wang, Q., Zhang, K., II, A.G.O., Xing, X., Liu, X., Giles, C.L.: A comparative study of rule extraction for recurrent neural networks. CoRR **abs/1801.05420** (2018), http://arxiv.org/abs/1801.05420
109. Wang, Q., Zhang, K., Ororbia II, A.G., Xing, X., Liu, X., Giles, C.L.: An empirical evaluation of rule extraction from recurrent neural networks. Neural Computation **30**(9), 2568–2591 (Jul 2018). https://doi.org/10.1162/neco_a_01111, http://arxiv.org/abs/1709.10380

110. Weitz, K.: Applying Explainable Artificial Intelligence for Deep Learning Networks to Decode Facial Expressions of Pain and Emotions. PhD Thesis, Otto-Friedrich-University Bamberg (Aug 2018), http://www.cogsys.wiai.uni-bamberg.de/theses/weitz/Masterarbeit_Weitz.pdf
111. Xie, N., Ras, G., van Gerven, M., Doran, D.: Explainable deep learning: A field guide for the uninitiated. CoRR **abs/2004.14545** (2020), https://arxiv.org/abs/2004.14545
112. Yang, Y., Feng, C., Shen, Y., Tian, D.: Foldingnet: Interpretable unsupervised learning on 3d point clouds. CoRR **abs/1712.07262** (2017), http://arxiv.org/abs/1712.07262
113. Yao, J.: Knowledge extracted from trained neural networks: What's next? In: Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2005, Orlando, Florida, USA, March 28-29, 2005. SPIE Proceedings, vol. 5812, pp. 151–157. SPIE (2005). https://doi.org/10.1117/12.604463
114. Yeh, C.K., Kim, B., Arik, S.O., Li, C.L., Pfister, T., Ravikumar, P.: On completeness-aware concept-based explanations in deep neural networks. CoRR **abs/1910.07969** (Feb 2020), http://arxiv.org/abs/1910.07969
115. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: 13th European Conf. Computer Vision - Part I. Lecture Notes in Computer Science, vol. 8689, pp. 818–833. Springer International Publishing (2014). https://doi.org/10.1007/978-3-319-10590-1_53
116. Zhang, Q., Cao, R., Shi, F., Wu, Y.N., Zhu, S.C.: Interpreting CNN knowledge via an explanatory graph. In: Proc. 32nd AAAI Conf. Artificial Intelligence. pp. 4454–4463. AAAI Press (2018), https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17354
117. Zhang, Q., Zhu, S.C.: Visual interpretability for deep learning: A survey. Frontiers of IT & EE **19**(1), 27–39 (2018). https://doi.org/10.1631/FITEE.1700808, http://arxiv.org/abs/1802.00614
118. Zhang, Y., Chen, X.: Explainable recommendation: A survey and new perspectives. Foundations and Trends in Information Retrieval **14**(1), 1–101 (2020). https://doi.org/10.1561/1500000066, http://www.nowpublishers.com/article/Details/INR-066
119. Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: 2016 IEEE Conf. Comput. Vision and Pattern Recognition. pp. 2921–2929. IEEE Computer Society (2016). https://doi.org/10.1109/CVPR.2016.319, https://arxiv.org/abs/1512.04150
120. Zhou, B., Sun, Y., Bau, D., Torralba, A.: Interpretable basis decomposition for visual explanation. In: Computer Vision – ECCV 2018. pp. 122–138. Lecture Notes in Computer Science, Springer International Publishing (2018). https://doi.org/10.1007/978-3-030-01237-3_8, https://link.springer.com/chapter/10.1007%2F978-3-030-01237-3_8
121. Zhou, J., Gandomi, A.H., Chen, F., Holzinger, A.: Evaluating the quality of machine learning explanations: A survey on methods and metrics. Electronics **10**(5), 593 (Jan 2021). https://doi.org/10.3390/electronics10050593, https://www.mdpi.com/2079-9292/10/5/593
122. Zilke, J.R., Loza Mencía, E., Janssen, F.: DeepRED – rule extraction from deep neural networks. In: Proc. 19th Int. Conf. Discovery Science. pp. 457–473. Lecture Notes in Computer Science, Springer International Publishing (2016). https://doi.org/10.1007/978-3-319-46307-0_29, https://link.springer.com/chapter/10.1007%2F978-3-319-46307-0_29