Computer Assisted Methods in Engineering and Science
29(4): 297–356, 2022, doi: 10.24423/cames.518

Interpretability versus Explainability: Classification for Understanding Deep Learning Systems and Models

Ivars NAMATĒVS[1)*], Kaspars SUDARS[1)], Artis DOBRĀJS[2)]

[1)] *Institute of Electronics and Computer Science, Riga, Latvia*

[2)] *Mondot Ltd., Riga, Latvia; e-mail: artis.dobrajs@mondot.lv*

[*] *Corresponding Author e-mail: ivars.namatevs@edi.lv*

The techniques of explainability and interpretability are not alternatives for many real-world problems, as recent studies often suggest. Interpretable machine learning is not a subset of explainable artificial intelligence or vice versa. While the former aims to build glass-box predictive models, the latter seeks to understand a black box using an explanatory model, a surrogate model, an attribution approach, relevance importance, or other statistics. There is concern that definitions, approaches, and methods do not match, leading to the inconsistent classification of deep learning systems and models for interpretation and explanation. In this paper, we attempt to systematically evaluate and classify the various basic methods of interpretability and explainability used in the field of deep learning. One goal of this paper is to provide specific definitions for interpretability and explainability in Deep Learning. Another goal is to spell out the various research methods for interpretability and explainability through the lens of the literature to create a systematic classifier for interpretability and explainability in deep learning. We present a classifier that summarizes the basic techniques and methods of explainability and interpretability models. The evaluation of the classifier provides insights into the challenges of developing a complete and unified deep learning framework for interpretability and explainability concepts, approaches, and techniques.

Keywords: explainable artificial intelligence, interpretable machine learning, deep learning, deep neural networks, interpretability, explainability.

1. Introduction

Exciting advances in the field of artificial intelligence (AI) have led to a variety of machine learning (ML) models now being used in complex and wide-ranging applications. At the same time, interest in the research and practical applications of explainable AI (XAI) is growing [1, 2]. Many deep learning (DL) models have borrowed heavily from current thinking in cognitive science and have rekindled interest in neural networks. DL approaches and, in particular, breakthrough results with deep neural networks (DNNs) include a number of successful technical applications such as computer vision [3, 4], speech recognition [5], video analysis [6], and precision medical diagnostics [7]. Despite the extensive research on the interpretability and explainability of DNNs in recent years, modeling interpretations and explanations through the use of various interpretation and explanation methods remain a challenge, e.g., axioms and metrics for explanations [8], self-explanatory models [9], or attribution-based methods [10].

Despite recent successes in XAI, it is still unclear how a particular DNN comes to a certain decision, how confident it is in its decision, whether and when we can trust or distrust it, or when it needs to be corrected. The main reason for this is that DNNs result in complex black-box models [11–19], i.e., where only the input features and the output predictions are known, making it difficult to understand the nature of learning within their structure. There are high-impact, high-risk domains where knowledge about the decision-making process is particularly important, e.g., healthcare decisions [20], autonomous driving [21], criminal justice [22], and other high-risk domains where the cost of poor decisions can have a large impact on human society [23]. The questions we should be thinking about here are "Can we trust decisions made by DL models?" or "How does a DL model make its decision?"

The growing interest of researchers in XAI has led the DL research community to focus on methods of interpretability and explainability of DNNs. The main problem with explainability, according to DARPA [1], is to provide sufficient justification for AI/ML conclusions such that users know why a conclusion has been reached or not, allow the user to know when an algorithm will succeed or fail, and when it can be trusted. To build trust and confidence in DL systems and models [24, 25], stakeholders need to gain insight into the decision-making process of a system or model by learning exactly why and how the DL system, model, and algorithm arrived at a particular solution.

There are several important reasons, either technical or methodological, why interpretable machine learning (IML) and XAI techniques and methods to be used in DL are a hot topic today and are becoming an active research area [18]. It can be assumed that the two key words "interpret" and "explain" mentioned above are never completely interchangeable in language use, especially not in technical and scientific language [26]. This also refers to the ability to interpret and explain DL systems and models in a human-friendly way. From the literature, we recognize several current research challenges we face regarding interpretability and explainability in DL systems and models:

- There are no clear differences between the two main concepts of XAI – interpretation and explanation [27].
- There is a lack of agreement on the vocabulary and different definitions of concepts and terms in IML and XAI [28]. For example, there is absolute ambiguity when it comes to explainable visualization, because terms such

as heatmaps, saliency maps, and salient maps are used interchangeably. Another example is that "feature relevance" and "feature importance" are assigned to the same concept.

- Based on the recent successful research methods for IML and XAI, it is worthwhile to create a classifier for interpretability and explainability methods for DL.

Answering these questions is important for the AI research community because it could enable the human-friendly use of practical DL applications to prove that decision behavior is based on plausible features [29] rather than speculative correlations of artifacts [30]. The goal of this paper is to attempt to define the boundaries between interpretability and explainability and provide a classification through a systematic review of interpretability and explainability methods and techniques. Overall, the novel contribution of this study can be summarized as follows:

1. We investigate the terminological and contextual difference between the two main concepts: interpretability and explainability and show that there is a clear difference between the two. The definitions of these concepts are presented in this article.
2. We address the problem of the difference between IML and XAI in DL, through the scope of the black-box and glass-box approach, system modeling, and a methodological perspective.
3. After thoroughly examiningrecently deployed XAI techniques and methods, we present the overall classification of IML and XAI for understanding DL systems and models. In addition to the classification, we display a brief description of the most prominent and commonly used IML and XAI methods and techniques and, our definition of each proposed category.

The rest of this article is organized as follows. Section 2 discusses the properties of a black box compared to a glass box, and shows how they relate to XAI and IML. Section 3 comments on the different approaches to interpretability. Section 4 addresses the methodological issues related to explainability. The taxonomies and classifications of XAI and IML, as well as a brief description of the existing categories of methods, are summarized in Sec. 5. The most common XAI methods are presented along with a brief description in Sec. 6. In Sec. 7, the paper classifies the domain of IML (intrinsic models) for interpretability. Section 8 presents toolboxes for explainability. Finally, conclusions and future directions are presented in Sec. 9.

2. Black box versus glass box

The use of DL is promising because it can process complex sets of data and model highly nonlinear internal representations of data. The models of DL can be composed of hundreds of layers, e.g., deep residual networks (ResNet) [31], which are over 200 layers deep and can encompass a vast parameter space with millions of parameters managed by hyperparameters [32]. DNNs have been shown to be particularly useful in tasks that indirectly or directly affect humans by producing classifications, regressions, predictions, or decisions.

One of the most prominent examples is convolution neural networks (CNNs) [3, 33], which have enabled unmatched developments in a wide range of computer vision tasks: from image recognition [34], image classification [3, 35], object detection and semantic segmentation [36, 37], image captioning [38, 39] to visual question answering [40–42] and, more recently, embodied question answering [43–45] and visual dialogue [46, 47]. It is well known that CNNs consist of an extremely complex internal structure and are therefore very difficult to explain. Fortunately, because human cognitive abilities prefer understanding visual data, these types of deep models can be more easily explained by visualizing their feature space. To understand the decision process of using CNNs, there are two approaches [28]: mapping the output back to the input space to see which parts of the input were crucial to the output, or trying to delve inside the network to interpret how the intermediate layers see the world, not necessarily with a particular input, but more generally.

Other examples are recurrent neural networks (RNNs) [48] and long short term memory (LSTM) [49], which are used for prediction problems on inherently sequential data, especially in natural language processing and language modeling [50], or for the analysis of temporal data series [51]. Interpretation of RNN models can be divided into two groups: using explainability by understanding what the RNN model has learned, mainly through feature relevance [52], and interpretability by modifying the RNN architecture to gain insight into the decisions they make [28]. The explanation of neural ensemble networks is another challenging task where specific explanation methods have recently been sought [53].

Research in DL modeling has traditionally focused on improving the quality, algorithmics, or speed of prediction of a neural network model [54]. At the same time, DNNs are generally considered as black-box models due to their multilayer nonlinearity and deeply nested structure, which are often criticized as opaque and incomprehensible to humans [55]. Since they are trained and not directly programmed, it can be difficult to discern how exactly they arrive at their decisions.

The term “black box” refers to a model that accepts a sequence of query inputs and produces corresponding outputs, while hiding internal states such as the model architecture [56]. There are black-box explanation methods that attempt to explain existing DL models without considering the internal structure of the model. This class of explanation methods is model-agnostic and can be easily integrated into DL models, from decision-trees to complex neural networks.

Black-box explanation methods are also referred to as *post hoc* methods because they can be used to interrogate DL models after training and deployment without knowing the training procedures. Explanations derived in this way are not in some way guaranteed to be human-friendly, useful, or enforceable, and may be dangerous in high-stakes decisions to the degree that others are not.

The opposite of a black box is an incremental system or “glass box” [17, 57], which is inherently transparent. Interpretive modules or tasks are integrated in advance into the DL architecture and algorithms and are referred to as *ante hoc*. The advantage of this approach is that practitioners can translate the models on DL, detect data and/or labeling errors, and in some cases, edit the models' decisions when they do not match the values or domain knowledge. This type of approach solves the problem of a trade-off between accuracy and interpretation, which is the problem of black box and *post hoc* models.

Black-box models, unlike glass-box models, are opaque, counterintuitive, and difficult to understand [58]. Due to their nonlinear structure, it is difficult to project the function they represent back into their input space and make sense of it. Moreover, they are not transparent, i.e., the provision of information, accountability, and prediction results of deep models is difficult to interpret and explain [20]. In the context of explainability, a black box is considered a function that is too complicated for a human to understand, e.g., a highly recursive function, a proprietary function, or even both [15].

There is a growing interest in understanding how these DL models arrive at their successful predictive tasks. Work on explaining these black-box networks has focused on understanding how a fixed deep model leads to a particular prediction [59]. The challenges in using DL models to explain their decisions are mainly due to the following: 1) the lack of transparency [57, 60], 2) the lack of explainability [61–63], 3) the large complexity and computational resources of current deep learning models [64], 4) the lack of robustness to adversarial attacks [65], and 5) the inability to explain decisions and actions in a way that humans can understand [66].

If we are trying to figure out what kind of input causes a particular DNN behavior – whether it is the firing of an internal neuron or the final output layer – we can use derivations to iteratively modify the input against that goal [67]. The question is what caused the trained DNN model to make a particular decision and how we can understand and quantify its inference. If we understand and trust the model, we need to clarify the decisions through interpretations [68]. From the perspective of developers and engineers, an explanation can help us to better understand the data and the problem, and provide a rational solution for using the system. Regulators consider explanations of algorithmic AI systems as a claim for compliance with the EU's General Data Protection Regulation (GDPR) [69].

These questions can be answered by a subset of AI – XAI, which aims to create a collection of methods that produce more explainable models while maintaining high levels of search, learning, planning, and reasoning through optimization, accuracy, and precision. The main idea behind XAI is to break down the black-box algorithm, explain how the black box made decisions, and account for the steps and models involved. These black-box functions can be extremely complex and too complicated for humans to understand. Referring to Gunning *et al.* [70], the purpose of an XAI system is to make its behavior more understandable to humans by providing explanations. The challenge for XAI is to produce explanations that are both complete and interpretable [9]. XAI systems should be able to provide historically scalable explanations of what the system has done, what it is doing now, and what will happen next, as well as reveal the key information to which it is responding.

XAI is concerned with implementing transparency and traceability of black-box statistical methods in ML, in particular DL [71]. Issues of explanation to their audience include at least four components: users, laws and regulations, explanations, and algorithms [72]. XAI assumes that there are many types of end-users or end-user groups, e.g., business owners, customer service representatives, IT and system operators, developers, data scientists, and policymakers. Each of these groups may differ based on the different times of system development and use [73]. Measuring and evaluating XAI systems requires evaluation frameworks, reasoning [74], mutual understanding, and diverse thinking [75].

Due to factors driving the improvement of AI systems, such as new algorithms and DL methodology [76, 77], GPU cards and on-chip neural networks, data availability, and cloud infrastructure, it has recently become possible to explain the actions of DL systems and evaluate the "explainability" of a system [18, 68]. These properties constitute explainable DL (XDL) and enable humans to understand, appropriately trust, and effectively manage DL systems and models using interpretability and explainability.

3. Interpretability

Let us clarify the discrepancies between the two main concepts widely used in XAI – interpretability and explainability. These two concepts are often used interchangeably in the literature. However, this is no longer the case for the new third wave of AI, that is, explainable AI applications, offering great benefits to a wide range of domains [70] (first wave – symbolic AI, second wave – statistic AI). Compared to task-centric AI systems, XAI systems are designed to perform specific tasks that lead to explanations or the creation of explanatory models that address black-box problems or IML systems, where interpretations or understandable models should be glass-box. This can lead toward *trustwor-*

thy artificial intelligence, the methodology for implementing human-friendly AI methods that focus on model transparency, fairness, and accountability [28]. This is how companies, regulators, and users view one of the key options for future AI. Thus, the distinction between interpretability and explainability is not only a crucial task, but also a challenging one. Therefore, we can argue that there are important reasons to distinguish between these two key concepts of XAI. Another question that arises is what is the difference between the two concepts and which is a more general concept, or could each of the concepts consist of sub-concepts [57]?

Montavon *et al.* [78] define interpretation as "the mapping of an abstract concept, for instance, a predicted class, into a domain that the human can make sense of". Gilpin *et al.* [9] define the goal of interpretability as describing the internals of a system in a way that is understandable to a human. A similar definition has been given by Doshi-Velez and Kim [25]: "interpretability is defined as the ability to explain or to provide the meaning in understandable terms to a human". There is no mathematical definition of interpretability. However, Doshi-Velez and Kim [25] outline an approach to measuring interpretability. A non-mathematical definition by Miller [79] is "interpretability is the degree to which a human can understand the cause of a decision".

Interpretability refers to a model's passive property and the level at which a particular model makes sense to a human in the context of *transparency*. By contrast, explainability can be considered a model's active character, which refers to the discovery of the model'd internal functions [28]. That is, interpretability is the result of the DL model, but explainability is the tool that must "open" this result. The higher the interpretability of a DL model, the easier it is for someone to understand why certain decisions or predictions have been made. A model is more interpretable than another model if its decisions are easier for a human to understand than the decisions of the latter one. From that, we can conclude that the concept of interpretability has a broader perspective compared to the concept of explainability.

Researches [2, 7, 14] hold a view on how to evaluate human interpretability. An approach that has emerged is that a proposed classification must consist of three main constituents: application-grounded evaluation, human-grounded evaluation and functional-grounded evaluation. A more frequent and clear option referring to Girshick *et al.* [80] is the scope of interpretability, divided into global interpretability and local interpretability. Together with the two, time limitation and nature of user expertise are added as a scope to better disentangle interpretability. Time limitation means that an important aspect is how much time that the user has or can spend on understanding an explanation. The extent of the user's experience in the task is another key aspect in the perception of interpretability of a model.

Global interpretability is about understanding how the overall deep model makes decisions that result from the prediction. This can be done by examining the complex structure and parameters of the entire model, which input patterns are captured, and how they are transformed to produce the output [81]. Local interpretability examines locally the reasons for the behavior of a deep model given a particular prediction. It is achieved by identifying the mapping of each feature in a specific input for the prediction made by the DNN [18]. Local interpretability helps to uncover the causal relationships between a specific input and the corresponding model prediction.

Along with human interpretability, which helps people understand machines, there is also machine interpretability. This refers to how machines "understand". For example, consider automatic care for the elderly or how an autonomous vehicle waiting at an intersection should reliably perceive and respond when a human in another car across the street signals to continue [25].

Identifying how to measure what interpretability means is not a trivial task, and there is little consensus on how to evaluate it. Logically, it would be clear that the higher the interpretability of a DL model, the easier it would be for someone to understand its decisions or predictions. A model is more interpretable than another model if its decisions are easier for a person to understand than the decisions of the other model. Lipton [57] points out that interpretability is not a monolithic concept, but reflects several distinct ideas and has a quasi-scientific character. Looking at the goal of DL from a modeling perspective, interpretability is closely related to the key end-user audiences from a system perspective. From the above, we can focus our attention on four specific concepts that describe interpretability: understanding, transparency, decision, and domain.

Understandability (also intelligibility) is the property of a model to make a human understand its operation without elucidating its internal structure or the internal operations by which the model processes data [82]. This property refers to the question: how does the deep model work? According to the previously mentioned machine interpretability and human interpretability, we can see that there is a machine (model) understandability and a human understandability. From the latter, we can see that the audience is the cornerstone of XAI when it comes to understanding the model [28]. When it comes to understanding DNNs, we have to deal with two views of understandability: a mechanistic understanding, i.e., what mechanism the network uses to solve a problem or implement a function, and a functional understanding, i.e., how the network relates the input variables to the output variables. From the above, we can conclude that comprehensibility is the most important concept in interpretable deep learning (IDL).

Transparency addresses the system's property to explain how it functions even when it behaves unexpectedly. A very important problem we are trying to solve for the ground truth of interpretability depends on complete transparency, and how a deep model arrives at its decision [83]. Roscher *et al.* [84] proposed to differentiate transparency into *model transparency*, *design transparency*, and *algorithmic transparency*. There are DNNs systems, where transparency as well as explainability are not the key requirements as long as the overall performance of these systems is good enough. If the transparency and trust requirements are not high, the system might lead to wrong decisions and even pose a threat.

On the other hand, if we create a trustworthy DNN system, there should be faith in the system and the prediction performance [85]. Lipton [57] divides transparency according to the property of understandability of a model. Referring to his proposal, models are divided into three levels: simulatable models (the level of the whole model), decomposable models (the level of individual components), and algorithmically transparent models (the level of training algorithms). In some contributions, interpretability is understood as a prerequisite for trust [57]. This indicates that the main idea of interpretability is to help people understand and trust the prediction tasks of DL models. These prediction tasks include decision support, e.g., tumor diagnosis, ranking, forecasting, and detecting anomalies.

In the context of XDL, interpretability can be considered to consist of three categories [90]. Firstly, *data interpretability*: which dimensions of the data are most relevant for the task. Secondly, *model interpretability*: what pattern belonging to a certain category typically looks like according to the model. Thirdly, *prediction interpretability*, explains why a certain pattern x has been classified in a certain way $f(x)$.

From the above, we can focus on specific concepts that describe interpretability: understanding, transparency (trust) and decision. Our definition, based on an extensive literature review, is as follows: **Interpretability means the ability for a human to understand and trust the decision of the DL model's results.**

For example, suppose we have a model that classifies patients for COVID-19 infection. Our DNN model has learned to classify patients' radiographs into three classes: normal radiographs, COVID-19 radiographs, and radiographs with pneumonia. The results are presented to the physician for further decision-making. If, after receiving the model results, the physician can see, read, listen and understand how the results were obtained and trust them, we could say that the predictions are interpretable. There is a logical path between the data, the image input, and the output results to understand and trust what concepts DNN has learned.

4. Explainability

The ability to verify the decisions of a DL system or models is very important to promote both trust and understanding in situations where the DL systems practically make the decisions, e.g., autonomous driving, and in situations where they play a supporting role, e.g., medical diagnosis [18]. Typically, standard models of DL take an input x and convert it to an output y, e.g., a predicted label that constitutes a further decision. In other words, let $f : \mathcal{X} \to \mathcal{Y}$ be the black-box function we are interested in explaining. For example, we might train a DNN, e.g., a classifier, to predict whether to accept or reject a medical diagnosis. If our request is rejected, we would like to know why. Because each decision of a DNN is a combination of thousands of neurons and weights interacting with each other, an explanation can be a very difficult task.

There is a great need to ask explanatory questions and know what, why, and how [79] an algorithm of a DNN makes a certain prediction and arrives at exactly that decision. The innate goal of explanation algorithms is to facilitate human understanding [86]. If users of a deep model understand the explanations, they will be more inclined to trust and adopt DL systems. From the perspective of the developers and researchers of DL systems, the explanations provided can help them better understand the problem and the data, understand why a model might fail, and even increase the system's security [87]. The safety of a DL system depends on explanations, fairness, security/privacy, and model debugging [88].

Probably the best definition of the explanation comes from Montavon *et al.* [78], who write that "an explanation is the collection of features of the interpretable domain, that have contributed for a given example, to produce a decision". Explanations can be *full* or *partial* [70]. Models that are fully explainable provide a complete explanation and are transparent [89]. Models that are only partially explainable reveal only important parts of their reasoning process. In scientific research, a scientific explanation should include at least two parts [90]: the object to be explained and the content of the explanation. Referring to Gunning [1], there are four types (modes) of explanation: analytical (didactic) statements, cases, visualization, and alternative choices. Another definition by Hendricks *et al.* [91] states that explanation should consider visual evidence, which includes two types of visualization: they must be class discriminative and accurately describe a specific image instance. One can distinguish between explanation systems with introspection, which explain how a model determines its final output, and explanation systems with justification, which produce sentences detailing how visual evidence is consistent with a system output.

Visual explanations highlight the regions of the DNN that are descriptive for the classes of interest, or, more generally, visualize the behavior of the model [92]. Visual explanations must satisfy two criteria. They must be class discriminative and accurately describe a particular image instance [91]. Textual explanations [93] are natural language statements that are verbally formulated or described. Textual explanations can be either template-based [86] or rule-based [94]. Explanations by example [57, 95] select particular instances underlying the data distribution to explain the behavior of the DL model. Explanations by simplification [96] involve dividing the complex feature space into simpler, explainable domains. Feature relevance attribution [97] assigns an importance score to each feature for specific input.

A good explanation should at least be faithful and interpretable [15]. A faithful explanation is an accurate characterization of the behavior of a model, while an interpretable explanation is easy for a human expert to understand [18]. There are two types of explanations that answer the faithful model: what has the network function learned to do, and how does it do it? The "what" question refers to the external properties of the function, such as whether it is invariant to the input. The "how" question refers to the internal functioning, i.e., how the hidden units process information to achieve invariance [15]. Hansen and Rieger [98] describe five general desiderata for a useful explanation of a DL system: fidelity, understandability, sufficiency, low construction overhead and efficiency.

There are different types of explanations, e.g., feature-based, instance-based, and language-based [27]. To find explanations, we need to define an explanation rule for a black box $f(x)$. First, we need to define which variations of the input x should be used to explore $f(x)$. The search for explanations can be formulated as the problem of learning meta-predictors that predict the behavior of a model [99]. There are explanation methods that differ in their approach. The goal they pursue is the same – to evaluate input variables according to their importance for prediction [18]. Montavon [8] consider three axioms of an explanation that can be traced to individual neurons of the network: conservation, continuity, and implementation invariance.

Araya *et al.* [27] argue that the explainability of an ML model is usually inversely related to its ML performance. Often DL models are the most powerful but least explainable, but this is not a rule of thumb. Decision trees are the most explainable, but with the least accuracy [70]. A different view is held by Watcher *et al.* [69], who argue that explanations do not have to satisfy explanatory accuracy. They say that counterfactuals are sufficient for an explanation. It is assumed that the higher the predictive accuracy, the lower the explanatory power of a model [9, 100], which is the case for black-box models.

Nevertheless, Murdoch *et al.* [101] describe another conflict between the descriptive and predictive power of a model. Yang *et al.* [102] argue that the explainability of DNN can be recognized by three main aspects. The first involves a complex function decomposition into sparse additive subnetworks. The second involves projection indexing in subnetworks that tend to be less confounded with each other. The third aspect involves subnetworks that can be used to better explain the functional relationship.

Using such an approach leads to an XDL that can balance the model‘s prediction performance and explainability. For instance, an example of sparse additive decomposition is given in [103], designated by the authors as an additive index model approach. Our definition based on an extensive literature survey is the following: **Explainability means the ability by which a human can justify the cause of the explanatory rule of the DL model's results.**

Let us refer to our COVID-19 example. Our physician wants to know not only the results of the classified radiographs, but also what the cause is and how exactly the model arrives at its prediction. Knowing what patterns, which pixels, and where in the network they are responsible for a particular prediction can help the physician make an informed decision based on the explanatory power of the model.

As for the explanation to the user, the explanation of the DL models of should be explained at all levels of the model state, which can be done through the ***scope of explanation***, ***modeling perspective***, and ***method viewpoint of explanation***.

The scope of the explanation consists of the following:

- **Global.** Global explanations attempt to focus on the entire model. The entire model can be explained, and reasoning can be followed from input to every possible outcome, e.g., the importance of features to all training data. This scope allows us to get a better picture of an entire model. This can be, for example, visualizing the weight distribution in a DNN or visualizing deep network layers propagating through the network.
- **Local.** Local explanation attempts to capture individual outcomes, e.g., to explain each prediction. The goal is to explain why a black box makes a particular prediction based on local features, e.g., pixels. These explanatory methods can be used for a small portion of a network, e.g., when considering a single filter in a deep network. Local explainability deals with a situation where it is possible to understand only the reasons for a particular decision.

Explainability by dividing through the ***modeling*** perspective [57] of the DL systems can be divided into two fundamental stages:

- **Ante hoc** (also known as intrinsic). An *ante hoc* explanation (Latin: before this event) of the decision of a black-box model is incorporated in advance into its architecture or conceptual constraint. *Ante hoc* systems provide explanations that go from the beginning of the model or input toward the output [57].
- **Post hoc** (Latin: after the event) explanations for the decision of a black-box model can be given after the fact. *Post hoc* explanations are concerned with how the model behaves in ways that are not readily interpretable by design [28]. The *post hoc* approach requires the creation of a second model (explainer) that provides explanations [68], such as visual explanations,

text explanations, local explanations, explanations by example, explanations by simplification, and feature relevance attribution.

The problem with *post hoc* explanations is that they may not define exactly how a deep model works. Nonetheless, they should provide useful information to users. *Post hoc* models are about being able to explain the process in terms of its outcome, for example, by determining which part of the input data is responsible for the final output. *Post hoc* analysis techniques attempt to uncover the significance of the various parameters, a goal we summarize as transparency. *Post hoc* techniques entail incorporating the explainability into a model from its outcome, such as marking which part of the input data is responsible for the final decision, e.g., the surrogates modeling method LIME [85]. These modeling techniques are easier to apply to different DL models, but say less about the whole model in general. In the *post hoc* approach [78], we assume that we have access to the parameters and architecture of the network under study.

Post hoc explainability methods are specifically designed and adapted to explain different types of DL models [28]. We can divide them into two groups: shallow DL models, which do not depend on layered structures of neural processing units, and deep DL models, e.g., convolutional neural networks, recurrent neural networks, modular neural networks, and transparent models [104]. The *post hoc* model analysis is a very common approach to explain AI systems and models. The main difference between *ante hoc* and *post hoc* by Du *et al.* [68] lies in the trade-off between model accuracy and explanation fidelity.

Explainable models can be divided according to their ***method viewpoint of explanation***:

- **Model-specific.** Model-specific explainability methods are restricted to a specific class of models [2]. They attempt to understand the DL model by analyzing the internal components of the network [68] and how they interact by examining activation functions or backward pass activations on the input. Model-specific explainability can be linked with a particular type of black-box model or input data. It is only suitable for a single type of model, e.g., visualization of the layers of a neural network, which is only applicable to DNNs. Model-specific methods generally examine the distribution of input or output data. These methods aim to understand the DL model by analyzing the internal components and their interaction. For example, it is possible to examine the activation units of DNNs and link the internal activations to the input. This requires a thorough understanding of the network and is not applicable to other models.
- **Model-agnostic.** The black-box explainers are model-agnostic and generally require access only to a model's prediction function, while the glass-box explainers generally require access to a model's internals, such as its loss function. *Model agnostic explanation* methods do not care which type

of model you have and, in many cases, correlate with a *post hoc* explanation. The model-agnostic methods can be divided into three categories: model simplification, attribution estimation (feature relevance) and visualization methods. A model-agnostic explanation for *post hoc* explainability is intended to be applied to any DL model by means of obtaining some information from its prediction procedure.

To benefit from explainable DL, we need to get to know and apply its processes, axioms, and methods. To increase the explainability power of DL models, one must take into consideration at least several of the most important desiderata for DL explainability:

- *Causality*: the ability of a method to clarify the relationship between input and output in a specified context of use [57, 71, 79].
- *Correctability*: the ability of a method to make necessary corrections back to the learning model [73].
- *Effectiveness*: the ability of a method to support good decision-making [105].
- *Efficiency*: the ability of a method to support faster best-option for end-user decision-making [100].
- *Explicitness*: the ability of a method to provide explanations immediately and in an understandable manner [106, 107].
- *Faithfulness*: the ability of a method to provide explanations that indicate the true relevant features [106, 107].
- *Fidelity*: the ability of a method to agree with the input-output mapping of the deep model [108].
- *Informativeness*: The ability of the method to provide useful information to the end-user via its output [57, 109].
- *Stability*: the consistency of a method to provide similar explanations for similar or neighboring inputs [106, 107].
- *Transferability*: the ability of a method to generalize and transfer new knowledge to unfamiliar situations [57, 108, 110].
- *Robustness*: The persistence of a method to withstand small perturbations of the input that do not change the prediction of the model [65, 106, 107].
- *Persuasiveness*: The ability of a method to convince users to perform certain actions [106, 107].
- *Scrutability*: the ability of a method to inspect a training process that fails to converge or does not achieve an acceptable performance [106, 107].

From the explainability modeling perspective, there is a challenge to apply several so-called explanators or explainers which try to point out the connection between input and output to represent in a simplified way the inner structure of DL black boxes [14]. Besides black-box explainers, there are glass-box (also known as white-box) explainers [27].

5. Taxonomies and classifications of interpretability and explainability

Most of the research on explainability and interpretability has been conducted after 2016. There are several taxonomies in the literature to distinguish XAI methods (Table 1). The proposed taxonomies differ according to the scope of the explanation, the level of explanation, the type of explanation, the type of model or data that can be explained, or a combination of these methods [27, 116]. One group of XAI researchers distinguishes them into *ante hoc* and *post hoc* methods. Another important distinction, distinguished by a second group, is between explanation methods that attempt to explain the decision-making process of a model at a global level and those that focus on explaining a single data sample, i.e., at a local level [111]. A third group distinguishes XAI methods as model-specific or model-agnostic [112]. There is an approach to classify XAI techniques into a quadrate by vertically indicated global and local explanations and horizontally marked model-specific and model-agnostic explanations [17].

Gilpin *et al.* [9] present explanations that focus on DL and divide them into three categories: 1) DN processing that, includes methods for producing insights between inputs and outputs of the deep model. This category includes linear proxy models, decision trees, automatic rule extraction, and saliency mapping, 2) DN representation that, includes methods attempting to explain representations of the inside of the network. This category includes the role of layers, the role of individuals, and the role of representation vectors, and 3) explanation-producing systems, the combinatorial approach that attempts to merge different explainability methods. This category includes attention networks, generated explanations, and disentangled representations.

Samek *et al.* [18] divide methods of XAI into: 1) explaining with surrogates, e.g., LIME [85], 2) explaining with local perturbations, e.g., sensitivity analysis [123], and prediction difference analysis [124], 3) propagation-based approaches (leveraging structure), e.g., LRP [125], guided backpropagation [86], and 4) meta-explanations, e.g., spectral relevance analysis [117], and network dissection [126]. Guidotti *et al.* [111] identify and categorize explainability methods into four categories: 1) by the type of explanation model, 2) by the data used as input, 3) by the problem that makes up the method, and 4) by "opening" the black box.

Lucieri *et al.* [129] distinguish three neural network explanation methods using DNNs: 1) saliency-based, e.g., Grad-CAM [130], SmoothGrad [131], integrated gradient [132], and LRP [125], 2) text-based, which can be either template-

Table 1. Examples of the XAI taxonomy and classification methods collections.

Authors, year [Ref.]	Description of the taxonomy or classification
Bodria *et al.*, 2021 [113]	Present XAI taxonomy for explaining black boxes for tabular data, image data and text data.
Linardatos *et al.*, 2021 [114]	A survey on XAI methods, codes and toolbox with references.
Zhou *et al.*, 2021 [115]	An overview of explanation methods and quantitative metrics.
Arrieta *et al.*, 2020 [28]	An extensive survey of explainable methods is divided into transparent and *post hoc* explainability.
Belle, Papantonis, 2020 [116]	Present a taxonomy divided into transparent and opaque models, with mapping methods for subcategories.
Benchekroun *et al.*, 2020 [117]	Designate explainability taxonomy into three categories: pre-modeling, modeling, and post-modeling.
Chari *et al.*, 2020 [118]	Propose an explanation ontology incorporating different explanation types by the role of explanation, accounting for the system and user attributes in the process.
Das, Rad, 2020 [119]	Propose XAI taxonomy based on the explainability scope, methodology and usage.
Hase, Bansal, 2020 [120]	Summarize explainability methods dividing them into feature importance, gradient-based and case-based reasoning categories.
Vilone, Longo, 2020 [107]	Provide an overview of methods divided by stage (*ante hoc*, *post hoc*), scope (global, local), problem type, input data, or output format.
Xie *et al.*, 2020 [121]	Present an XAI taxonomy based on visualization methods, model distillation and intrinsic methods.
Arya *et al.*, 2019 [27]	Taxonomy is based on what is explained (data or model), how it is explained (*post hoc* or *ante hoc*) and at what level (global or local).
Carvalho *et al.*, 2019 [122]	A survey of ML explanation methods and metrics.
Adadi, Berrada, 2018 [2]	The explainability taxonomy is divided by scope (global, local), *post hoc* (model-agnostic) and intrinsic (model-specific) methods.
Gilpin *et al.*, 2018 [9]	Present explanations dividing them into three categories: DN processing, DN representation, and explanation producing systems.
Guidotti *et al.*, 2018 [111]	A summary of the methods for opening and explaining black boxes concerning the explanator adopted.

based [133] or rule-based [134], and 3) concept-based, e.g., concept activation vectors [135]. Murdoch *et al.* [101] developed a predictive, descriptive, relevant (PDR) framework that introduces metrics for explainability methods, predictive accuracy, descriptive accuracy, and relevance. The framework is based on the assumption that explainability is divided into categories, *post hoc* interpretation, and transparent models.

Arrieta *et al.* [28] begin by dividing the taxonomy into *post hoc* explainability and transparent models, with *post hoc* explainability subdivided into model-agnostic and model-specific. The authors identify four categories related to DNNs, 1) explanation of DN processing, 2) explanation of DN representation, 3) explanation of producing systems, and 4) hybrids of transparent and black-box methods. Hase and Bansal [120] divide the taxonomy of interpretability into three categories: 1) feature importance estimation, which includes, for instance, gradient-based approaches, 2) case-based reasoning, e.g., prototype models for computer vision, and 3) latent space traversal, which shows how the model behaves when its input changes.

Another taxonomy presented by Liao *et al.* [100] divides XAI methods according to their mapping to user question types: global (explain the model), local (explain the prediction), inspect the counterfactuals or based on an example. Nguyen and Martínez [136] propose a set of metrics for the programmatic evaluation of interpretation methods and divide the methods into example-based and feature attributions. Sokol and Flach [62] compiled a list of functional, operational, and user-friendly features of explanatory methods for predictive systems called the explainability fact sheet – a framework for systematic evaluation of explainability approaches. Lage *et al.* [137] focus on explanations in terms of decision sets (also known as rule sets) as a starting point for a study of explanatory methods. Xie *et al.* [121] list visualization methods, model distillation, and intrinsic methods for DL. Ras *et al.* [72] divide explanatory methods into three categories: rule extraction methods, attribution methods, and intrinsic methods.

More recently, Belle and Papantonis [116] presented a taxonomy of XAI with a map of explainability approaches, dividing XAI by model types, explainability categories, explainability principles, and common techniques. Cortez and Embrechts [138] write about two main approaches: extraction of rules and use of visualization. Some recent work on XAI taxonomy is presented by Bodria *et al.* [113], Linardatos *et al.* [114], and Zhou *et al.* [115]. Banchekroun *et al.* [117] propose three main categories of explainability: 1) pre-modeling explainability, which focuses on the study of the input, 2) modeling explainability, which focuses on the inner workings of the model (mathematical aspect), and 3) post-modeling explainability, which focuses on data-driven methods.

6. Explainability methods

This section aims to provide a concise overview of the classification of modern XAI attribution and distillation approaches and IML by summarizing their goals, along with some basic methods and their objectives, see Fig. 1.

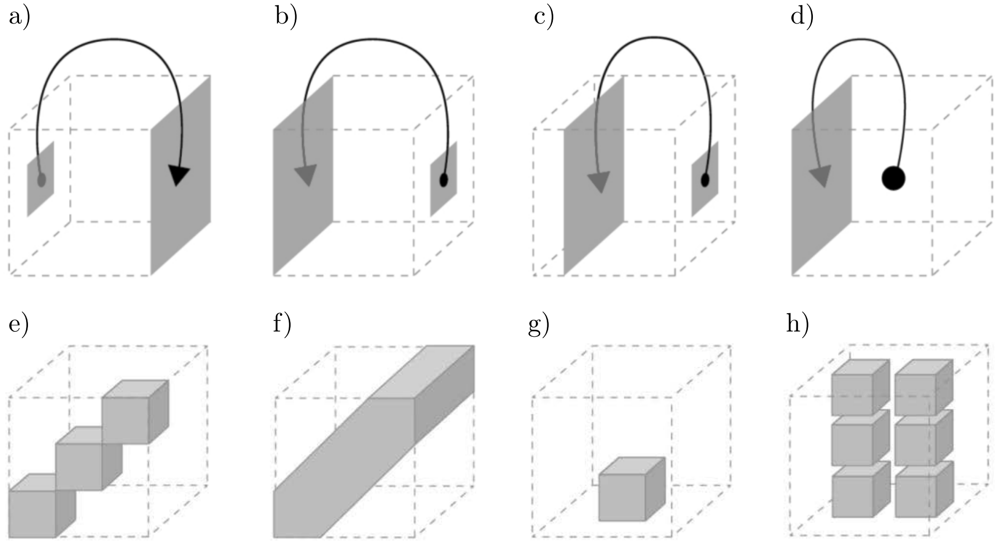

FIG. 1. Schematic conceptual examples of XDL approaches: a) local perturbation-based attribution, b) propagation-based attribution output to input, c) propagation-based attribution output to layers, d) propagation-based attribution neuron to input, e) decomposition, f) dissection, g) surrogate model, h) intrinsic approach.

The ***attribution*** approach focuses on measuring the feature relevance scores. We divide the explainability methods of this approach into three subcategories: **perturbation-based** explanations, **propagation-based** explanations, and **structural** explanations. The first and the second belong to the output-input of the deep network, and the third tries to use the internal structure of the deep network for an explanation. The distillation approach focuses on reducing the complexity of DNN models by transforming them into simple, easy-to-understand surrogate models. We divide surrogates into three subcategories: local approximation explanations, rule-based explanations, and tree-based explanations.

The proposed IML and XAI classification for understanding the models of DL is based on the fundamental algorithmic problem-solving approaches based on black-box model explanation and glass-box perspective using common basic *post hoc* and *ante hoc* methods. The task was to create a unified and generic classifier based on the concepts of interpretation and explanation as well as scope, modeling, and methodological perspective. The study of existing algorithms for the explanation of black-box models is divided into the level of features, layers and neurons. The proposed classification system is divided into *post hoc* and *ante hoc* (intrinsic), which must support the requirements of XAI and IML. The former is further divided into perturbation-based, propagation-based, surrogate models, and meta-explanations, and the latter into attention mechanisms and joint training of explanatory models. The set of methods covered here is not exhaustive, but sufficiently representative to discuss a number of issues referring to the taxonomy of explainability, possible applications and technical challenges.

6.1. Perturbation-based methods

Perturbation-based local methods, see Fig. 1a and Table 2, express an explanation by covering, removing, or changing the input features and/or activation units of a DNN to measure the corresponding change in the model output. The perturbation can be realized by *permutation*, which permutes the feature values, or by *uniform* distribution, where the feature values can be replaced by a random sample from a uniform distribution. This information is then used to evaluate the importance of the feature. *Perturbation-based forward propagation* approaches make perturbations to individual inputs or neurons and observe the effect on further network neurons, layers in the network and network output [139]. This means that perturbation-based methods try to evaluate the importance of input pixels by measuring the classifier, and how our network reacts to changes in the input. For example, if you mask a certain part of the input, how does it affect a classification, or not.

Therefore, the idea behind these methods is to assess feature relevance by testing the model response to the removal and perturbation of features [81]. For example, relevance measures the strength of the connection between the input or pixel to the specific network output [140]. While perturbation-based methods allow direct estimation of the marginal effect of a feature, they tend to be very slow as the number of features to test grows. What is more, given the nonlinear nature of DNNs, the result is strongly influenced by the number of features that are removed altogether at each iteration. Perturbation-based methods only require the propagation of one forward and/or backward pass through the CNN to generate an attribution visualization. In addition, perturbation-based methods are mostly iterative optimization-based approaches that require multiple passes through a network [141].

Definition: **Explainability of the DL model can be described through manipulation (altering, removing, deleting) of the input feature or intermediate layers (activations).**

The method of **occlusion** was proposed by Zeiler and Fergus [142]. The idea behind this method is to divide the input, e.g., an image, into a grid of regular, non-overlapping patches. Occlusion analyses depend on function value instead of a gradient. To calculate attribution (feature relevance), a function value is assigned to each patch when the patch region in the original image is perturbed or replaced by a specific baseline or reference value. Then the difference in network output is calculated. The probability of the correct class reduces significantly if the object of the image is occluded. When the patch covers the

Table 2. A summary of the **perturbation** explanation methods for explaining black box.

Category	Method	Authors, year [Ref.]	Code	Model components
Local perturbation-based explanation	Occlusion sensitivity (OS)	Zeiler, Fergus, 2013 [142]	NOC	IMG, CL, SM, PH, MA, L, CNN
	Meaningful perturbations (MP)	Fong, Vedaldi, 2017 [140]	NOC EC [link]	IMG, CL, SM, PH, MS/MA, L, DNN not specified
	Extremal perturbations (EXP)	Fong *et al.*, 2019 [143]	[link]	IMG, CL, SM, MA, CNN
	Prediction difference analysis (PDA)	Zintgraf *et al.*, 2017 [124]	[link]	IMG, CL, SM, IM, PH, L, CNN
	Randomized input sampling for explanation (RISE)	Petsiuk *et al.*, 2018 [144]	[link]	IMG, SM, MS, L, CNN
	Universal adversarial perturbations (UAP)	Moosavi-Dezfooli *et al.*, 2017 [145]	[link]	IMG, MA
	Epresentation erasure (RE)	Li *et al.*, 2016 [146]	NCC	EMB, CL, sequence tagging, HM of importance, SH, LSTM, RL

For every method, there is a reference link, code link: no official code (NOC), no community code (NCC), example code (EC). The data type on which it is possible to apply it: image (IMG), classification, (CL), embedding (EMB), explanation interface: saliency maps (SM), heatmaps (HM), importance maps (IM). If it is *post hoc* (PH) or *ante hoc* (AH) stage, model specific (MS) or model agnostic (MA), and local (L) or global (G) scope. Finally, for which type of DNN the method is appropriate: convolutional neural network (CNN), long-short-term memory (LSTM) or, reinforcement learning (RL).

critical area, the output prediction performance decreases significantly [78]. For features contained in multiple patches, the corresponding output differences are averaged to compute the attribution for that feature. The visualization shows the sensitivity range of an image for its classification label [78]. **The meaningful perturbations (MP)** method proposed by Fong and Vedaldi [140] applies meta-predictors as explainers. This is an optimization-based method observing how the output value of $f(x)$ changes as input x is penalized by deleting specific regions R. Attribution aims to identify which regions of an image x_0 are used to produce the output value $f(x_0)$. The idea is to not iterate over all possible perturbations but to search locally for the best perturbation mask m^*, i.e., to find the smallest deletion mask. The **extremal perturbations (EXP)** method [143] is where the perturbations are optimized by choosing smooth perturbation masks maximizing the model‘s confidence score. Extremal perturbations are regions of an input image that maximally affect the activation of a certain neuron in a DNN. **Prediction difference analysis (PDA)** [124] presents the method in which conditional and multivariate sampling are used within the pixel neighborhood of an analyzed feature to effectively remove information. **Randomized input sampling for explanation (RISE)** [144] explains actual DNN black-box models by estimating pixel saliency importance (importance map) of the input image regions. The importance of pixels is estimated by blurring them in random combinations, reducing their intensities to zero. The authors [145] develop a fast saliency detection method called **universal adversarial perturbations (UAP)** for image classifiers by manipulating the scores of classifiers by masking salient parts of the input image. Li *et al.* [146] propose the **representation erasure (RE)** method, which is used on the model by erasing various parts of the input word-vector dimensions, intermediate hidden units, or input words. Such representation erases NLP tasks, offers clear explanations about neural model decisions, and provides a way to conduct error analysis on the neural network.

6.2. Propagation-based methods

These types of methods, see Figs. 1b–1d and Table 3, try to explain any black-box function $f(x)$, e.g., a neural network object classifier. Since such a function is learned automatically from data, we would like to understand what it has learned to do and how it does so. For instance, to evaluate the importance of a model, a DNN can be regarded as a function, and we can compute gradients and sensitivity values or approximate the function using the functional perspective of explanation.

These methods use two fundamental axiomatic approaches to the information embedded in a neural network, i.e., gradients produced during the forward

Table 3. A summary of the **propagation** methods for explaining black box.

Category	Method	Authors, year [Ref.]	Code	Model components
Gradient-based explanation	Sensitivity analysis (SA)	Baehrens *et al.*, 2010 [147]	[link]	IMG, EMB, CAD, CL, HM, scatter plot, PH, MA, L
	Gradient * input (GI)	Shrikumar *et al.*, 2016 [148]	NOC	IMG
	Integrated gradients (IG)	Sundararajan *et al.*, 2017 [132]	[link]	IMG, TXT, CL, language tran., SM, PH, MS, L, CNN or NOSP
	Guided backpropagation (GBP)	Springenberg *et al.*, 2015 [151]	NOC	IMG, PH, L, CNN
	SmoothGrad (SG)	Smilkov *et al.*, 2017 [130]	[link]	IMG, SM, PH, MS, L
	Influence functions (IF)	Koh, Liang, 2017 [59]	[link]	IMG, PH, G, CNN
	Influence-directed explanations (IDE)	Leino *et al.*, 2018 [152]	NOC, CC [link]	IMG, SM
Activation-based explanation	Activation maximization (AM)	Erhan *et al.*, 2010 [67], Nguyen *et al.*, 2016 [153]	NOC	IMG, CL, PH, L, DBN, SDAE
	Class activation map (CAM)	Zhou *et al.*, 2015 [154]	NOC	IMG, TXT, CL, VQA, REG, localization, PH, L, CNN
	DeConvNet (DCN)	Zeiler, Fergus, 2013 [142]	NCC [link]	IMG, CL, PH, G, CNN
	DeepLIFT rescale (DLR)	Shrikumar *et al.*, 2017 [155]	NOC	IMG, DNA sequence, CL, visual, SM, PH, MS, L, CNN
Grad&activ explanation	Excitation backProp (EXB)	Zhang *et al.*, 2016 [156]	NOC	CNN
	Grad-CAM (GC)	Selvaraju *et al.*, 2016 [131]	[link]	IMG, CL, SM, PH, MS, L, CNN
	Grad-CAM (GC++)	Chattopadhyay *et al.*, 2018 [159]	[link]	IMG, CL, SM, PH, MS, L, CNN
	GradientSHAP (GSHAP)	Lundberg, Lee, 2017 [160]	[link]	IMG, PH, G

For every method, there is a reference link, code link: no official code (NOC), community code (CC), data type on which it is possible to apply it: image (IMG), embedding (EMB), categorical data (CAD), text data (TXT), problem: classification (CL), explanation interface: saliency maps (SM), heatmaps (HM). If it is *post hoc* (PH) or *ante hoc* (AH) stage, model specific (MS) or model agnostic (MA), and local (L) or global (G) scope. Finally, for which type of DNN the method is appropriate: convolutional neural network (CNN), deep belief nets (DBN) or, stacked denoising auto-encoders (SDAE).

pass and the backward pass, or a propagation-based algorithm to estimate the importance of the feature to be questioned. The first set, gradients (of the output concerning the input), is a rational analoge of the model coefficients for a DNN. Therefore, the product of the gradient and feature values is a reasonable starting point for an attribution (method) [127, 176]. Gradients provide a local explanation. The magnitude of the gradient shows the importance of the feature. The second set, the propagation-based methods, requires the propagation of one forward and/or backward pass through a neural network, e.g., the CNN, to generate an attribution visualization. These methods fall roughly into three categories [102]: 1) gradients, 2) activations, and 3) a combination of gradients and activations.

Definition: **Explainability of the DL model can be explained by considering the deep network as a function (each neuron or group of neurons) by using gradient and backpropagation axioms of the function of interest to define the explanatory rule.**

6.2.1. Gradient-based explanations. The **sensitivity analysis (SA)** method [147] studies the effect of different input features on the output values by changing the input values (features) and checking what happens in the output [27, 123]. Sensitivity analysis [81] explains a prediction based on the model's locally evaluated gradient (partial derivative) [62], some other local measures of variation as relevance scores [115] or class activation probabilities [122]. The **gradient * input (GI)** is the attribution method [148], where the information of the gradient of the neural network as a function (e.g., model) for each input dimension will increase if tiny steps in this direction are taken. The algorithm consists of an element-wise multiplication of the gradient times the input [83]. The gradients indicate the importance of a dimension, but the inputs suggest how strongly this dimension is expressed in the image. This method is preferable to gradients alone as it leverages the sign and strength of the input. In **integrated gradients (IG)** [132], instead of computing the gradients at the current value of the input, we can integrate the gradients. The inputs are scaled up from some starting value, e.g., all zero, to their current value [149]. The baseline input for the image networks could be the black image, while for text models, it could be the zero-embedding vector [150]. This method relies on integrating the gradients of the output prediction with respect to the input over a series of chosen variants of the input. The idea behind the **guided backprop (GBP)** [151], which is the feature backpropagation method, is that neurons act as detectors of a particular feature. During backpropagation, the negative gradients are removed. Guided Backprop is proposed for use in CNNs with ReLU to visualize pixel-space gradients with respect to the image. Smilkov *et al.* [130] propose the **SmoothGrad (SG)** method, where the discontinuous gradient is smooth with a Gaussian kernel.

The authors attempt a stochastic approximation that takes random samples in a neighborhood of the input x and then averages their gradients. The method samples the neighbouhood of the input to approximate the gradient. **VarGrad (VG)** is a similar explanation method to SmoothGrad, taking into calculation the variance V with noise-adding [10, 130]. Koh and Liang [59] propose the **influence functions (IF)** method taken from statistics measuring the sensitivity of the model to changes in the distribution of the independent variable. IF shows how the deep model parameters change as we upweight a training point by an infinitesimal amount. The explanation is done using examples that the model finds most similar or useful by calculating the gradient concerning each training epoch. The perturbation of the input is done by applying a constant shift. The **internal influence (IIF)** method is proposed by Leino *et al.* [152] and it is like the integrating gradients method. The difference is that the integrating gradients refer to the network layer rather than the network input. Another method presented by Leino *et al.* [152] is the **influence-directed explanations (IDE)** method for deep networks. It peers inside the network to identify neurons with a high influence on the model's behavior and then uses visualization to explain the concepts the neurons represent. The authors provide a novel distribution influence measure to identify which neurons are most influential in determining the model's behavior in a given distribution of instances.

6.2.2. Activation-based explanations. The **activation maximization (AM)** [67, 153] is an *example-based* explanation method, where examples are used to explain the neural network. The input patterns can be explained by the activation of a unit. This method belongs to an optimization-based explanation with visualization of important features in any layer of DNN. Zhou *et al.* [154] describe the procedure of creating a **class activation map (CAM)** method, which can generate the localization maps for the prediction through the classification layer. The idea behind this method is that a predicted class score is mapped back to the previous convolution layer to generate the class activation map. The maps are created by using a global average pooling layer after the last convolution layers in CNNs and before the final fully-connected layer (FC). The maps highlight the class-specific discriminative image regions for image classification used by CNN. The **DeconvNet** method is a calculation of abackward convolutional network that reuses the weights at each layer from the output layer back to the input image. The method makes it possible to create feature maps of an input image that activate certain hidden units (hidden neurons) most linked to a particular prediction [93]. Zeiler and Fergus [142] introduce an image-based DeConvNet method that gives insight into the function of intermediate network layers and the operation of the classifier. Simonyan *et al.* [123] introduce a gradient-based DeconvNet method that computes the gradient of the output class score with respect to the input image. First, an image can be generated which maximizes the class score. Then a class saliency map is computed to be specific for a given input and output class. We assume that the salient regions are at locations with high gradient magnitude. Shrikumar *et al.* [155] propose the **DeepLIFT (deep learning importance features)** method to compute the relevance scores of features in a multilayer neural network. DeepLIFT compares the activation of each neuron j to its "reference activation" value j' and assigns a relevance score R according to the difference. The same approach is valid if one assigns the relevance scores to input features based on the difference between an input x and a "reference input" x'.

6.2.3. Gradient and activation-based explanations. The **excitation backprop (EXB)** [156] method modifies the backpropagation rules by passing the neurons along top-down in the network hierarchy such that a backpropagated signal is weighted by a convolution layers‘ activation, while bottom-up information is used to compute the winning probability of each neuron. The method exploits the selective tuning attention approach [157]. Selvaraju *et al.* [131] propose the **Grad-CAM (GC)** method, which is a generalization of CAM [154]. The method makes sense for convolutional models by providing visualization of the class-specific gradient information of the output concerning the given layer. GC and DeepResolve [158] are two gradient ascend-based methods [107]. The GC method uses the gradients of any target concept, i.e., car, flowing into the final convolutional layer to generate a heatmap that can generate the influential regions in the image for predicting the concept. The activations can be explained in any layer of a deep network. Such an attribution method is usually leveraged to the last convolution layer of the CNN, to produce a coarse localization map of the important regions in the image. The importance values can be assigned to each neuron for a particular decision of interest. Grad-CAM is a class-specific visualization for every class present in the image. The result is a class activation heatmap (attention map) for an image classification model. To compute more fine-grained feature importance, Selvaraju *et al.* [131] propose the **guided Grad-CAM (GGC)** method, which leverages an element-wise product between the scores obtained with guided backpropagation attributions, i.e., Guided Backpropagation and the scores obtained with layer up-sampling. Other similar methods are **Grad-CAM++ (GC++)** [159] a gradient-based visualization method and **gradient SHAP (GSHAP)** [160]. GradCAM++ is an extension of Grad-CAM that produces better visual explanations of the predictions of the CNN. This method is especially helpful in multi-label classification problems, while a different weight assigned to each pixel makes it possible to capture the importance of each pixel separately in the gradient feature map.

6.3. Structural-based methods

Dissection, see Fig. 1f, translates qualitative visualization of units of representations into quantitative explanations and measurements of explainability [128]. Dissection-based explanations provide information on individual units in a DNN. **Decomposition**, see Fig. 1e, [125, 161, 162] is the process of resolving a model relationship into its constituent individual components, e.g., input and output parameters. The output prediction is redistributed backward in the network to eventually assign relevance scores to each input variable [163]. While DNN models enable superior performance, their lack of decomposability into individual intuitive components makes them hard to interpret [57]. Typically, this category of methods is based on a decompositional approach [164] which can be applied to any neural network whose output function is monotone, e.g., sigmoid function. Decomposition splits the network at the neuron level. The neurons of a network can be transformed into logical formulas, then aggregated to represent the network as a whole. Such an approach can be used in time series by using perturbation techniques. Table 4 presents a summary of the **structural methods** for explaining black box.

Definition: **Explainability of the DL model can be explained by examining the inner structure of the deep network by decomposing or dissecting it on a unit, layer, and/or specific neuron level.**

6.3.1. Dissection-based explanations. Zhou *et al.* [165] develop the **network dissection** method, which explains neural networks by assigning meaningful labels to their representational units. The proposed method extracts and evaluates the semantics of the hidden units, i.e., it quantifies which concept these neurons encode [116, 166]. Network dissection quantifies the explainability of any network by measuring the degree of correspondence between the activation of the units and the ground-truth labels in a predefined dictionary of concepts [167]. The method defines the quality of explanation of concept c for a unit k by quantifying the ability of k to solve the segmentation problem given by c using the IoU score. The value of $IoU_{k,c}$ is the accuracy of unit k in recognizing concept c. Network Dissection translates the qualitative visualization of representation units into quantitative explanations. Kim *et al.* [135] introduce testing with the **concept activation vectors (TCAV)** method, which explains the internal state of a neural network through a combination of feature vectors. TCAV shows the importance of high-level concepts, e.g., race, gender, color, texture, etc., for a prediction class – similar to human communication. This method provides an explanation that generally applies to the class of interest, over and above an image. TCAV learns concepts from examples. The notation of TCAV is a way of translating between E_h and E_m, where E_h is a high-level human-interpretable

Table 4. A summary of the **structural methods** for explaining black box.

Category	Method	Authors, year [Ref.]	Code	Model components
Dissection-based explanation	Network dissection (ND)	Zhou *et al.*, 2018 [165]	NOC CC [link]	IMG, CNN
	Testing with concept activation vectors (TCAV)	Kim *et al.*, 2018 [135]	[link]	IMG, CAD, PH, MA, G, CNN
	Automatic concept-based explanations (ACE)	Ghorbani *et al.*, 2019 [170]	[link]	IMG, CAD, PH, MA, G, CNN
	GAN dissection (GAND)	Bau *et al.*, 2018 [171]	[link]	IMG, CL, PH, L, CNN
	Cluster explanations (CE)	Kauffmann *et al.*, 2019 [172]	NOC	‘neuralization-propagation’
	Explorative generative boundary (E-GBAS)	Jeon *et al.*, 2019 [173]	NOC	IMG, DGNN
Decomposition-based explanation	Layer-wise relevance backpropagation (LRP)	Bach *et al.*, 2015 [125]		ANY, FI, SR, PH, MA, L, multilayer network
	Deep Taylor decomposition (DTD)	Montavon *et al.*, 2015 [161]	NOC, CC [link]	IMG, CL, HM, PH, G
	PatternNet, PatternAttribution	Kindermans *et al.*, 2017 [175]	NOC, CC [link]	IMG, CL, PH, L
	Relative attribution propagation (RAP)	Nam *et al.*, 2019 [97]	[link]	IMG, network neural importance
Time-series explanation	N-BEATS	Oreshkin *et al.*, 2020 [176]	CC [link]	univariate forecasting interpretation
	Perturbation on time-series (PoTS)	Schlegel *et al.*, 2020 [177]	NOC	docker container

For every method, there is a reference link, code link: no official code (NOC), community code (CC), data type on which it is possible to apply it: image (IMG), embedding (EMB), categorical data (CAD), problem: classification (CL), explanation interface: saliency maps (SM), heatmaps (HM), feature importance (FI). If it is *post hoc* (PH) or *ante hoc* (AH) stage, model specific (MS) or model agnostic (MA), and local (L) or global (G) scope. Finally, for which type of DNN the method is appropriate: convolutional neural network (CNN) or, graph neural network (DGNN).

concept space and E_m is an input vector space. TCAV uses directional derivatives to compute classifier models, i.e., "conceptual sensitivity", across an entire class of inputs. The TCAV method considers directional derivatives in the space of activations. The directions correspond to higher-level concepts that can be interpreted by humans instead of input gradients [168]. These main concept directions and directional derivatives can quantify the influence of a concept on the prediction of a particular output class. Graziani *et al.* [169] redefine CAVs from a classification problem to a regression problem by computing regression concept vectors. Ghorbani *et al.* [170] develop the **automatic concept-based explanations (ACE)** method for unsupervised clustering of objects by segmenting a single object. First, the multi-resolution image is segmented into a pool of segments that are all from the same class. Then, the activation space of the CNN is used as a similarity spacer. After resizing each segment to the standard input size of the model, similar segments in the activation space are clustered. Finally, the importance score is calculated for each concept based on its example segments. Bau *et al.* [171] present **GAN dissection**, a segmentation-based network dissection method for decomposing networks to understand and visualize GANs at different levels of abstraction. An inferential process to understand a network can be built from each neuron level to each object. The contextual relationship between different objects is controlled by identifying units or groups of units that are related to semantic classes, e.g., ships in an image. This is the first systematic approach for understanding internal representations across different layers of GANs by using causal effects. The **cluster explanations (CE)** [172] method is based on a concept called "neuralization propagation". The cluster model, e.g., the k-means cluster model, is first transformed into a neural network. Then the output of the network is explained in terms of the input features using a reverse propagation approach. The **explorative generative boundary (E-GBAS)** [173] method defines generative boundaries that determine the activation nodes in the internal layers. The internal layers of the network are characterized by collecting samples within the region surrounded by generative boundaries.

6.3.2. Decomposition-based explanations. Bach *et al.* [125] propose the **layer-wise relevance backpropagation (LRP)** method to explain a neural network classifier decision by pixel-wise decomposition. The method explains individual predictions of DNN in terms of input variables such as text, images, or videos. Based on the connecting weights, the prediction confidence has been redistributed in the opposite direction,. For example, LRP calculates a single pixel to the prediction made by the network in the image classification tasks. Later the method has been extended to RNN architectures such as LSTMs and GRUs by proposing a specific propagation rule applicable to multiplicative connections. This method, based on the feature relevance technique, delivers insightful explanations in the form of input space relevance for understanding feed-forward neural networks [52]. The central idea of relevance propagation uses conservation property (conservation propagation or principle) to propagate the outcome decision back without using gradients [174]. More specifically, this property declares that each neuron receives a portion of the network output and redistributes it to its forerunner in an equal amount until the input features are reached [125]. We can evaluate the explainability of a deep model by using **Taylor decomposition**, which is a general function analysis tool in mathematics. The Taylor expansion is a method to decompose a composite function into its component functions associated with different combinations and degrees of an input variable. Montavon *et al.* [161] consider each neuron as an object that can be decomposed and expanded, then aggregated and backpropagated through the network [28]. In this case, we get a product of the gradient and input relative to the root point as a relevance score: $R_i = [f'(\widetilde{x})]_i \cdot (x_i - \widetilde{x}_i)$. Note that when the function is highly non-linear a Taylor-type decomposition is applied to one layer or a subset of layers, approximating the relevance propagation. **Deep Taylor decomposition (DTD)** [161] is a method that explains the model's decision by decomposing the function value $f(x)$ as a sum of relevance scores [78]. The key idea of DTD is that if a decision is too complex to explain, it is necessary to decompose the decision function learned by a deep network into a set of simpler subfunctions and explain each subfunction separately. This can be done either by structurally imposing the neural network connectivity or training and explaining each subfunction separately. **PatternNet and PatternAttribution** [175] aim at computing the contribution of the input "signal" direction per neuron by learning it from data. Such a signal contains information about the output class as well as filtering out the rest of the input, for example, the background of the image. PatternNet is a layer-wise back-projection method that projects back the likely signal to the input space. Whereas, PatternAttribution is a neuron-wise contribution method that produces explanations consisting of contributions of the estimated signal to the classification scores [110]. Pattern Attribution is based on deep Taylor decomposition. Nam *et al.* propose [97] the **relative attribution propagation (RAP)** method that decomposes the predictions of the output of DNN by separating the relevant (positive) and irrelevant (negative) attributions according to the relative influence between the layers.

6.3.3. Time-series explanations. Oreshkin *et al.* [176] propose the **N-BEATS** method for a very deep neural network architecture to explain univariate time series points. The method is based on the principle of double residual stacking, which uses previously organized building blocks organized in stacks. The building block predicts the expansion coefficients both forward (forecast), θ^f, and backward (backcast), θ^b. The explanation of the time series was performed by backcasting and forward casting the residual links of the deep stacking of FC layers of the neural network. Schlegel *et al.* [177] propose a **perturbation on time series (PoTS)** technique for time series in which a time point is set to zero and time intervals around the relevant time points are acted upon.

6.4. Surrogate models

Distillation methods attempt to create surrogate models that must generate a separate, glass box model that is trained to mimic the input-output leverage of the internal representations of the original DNN [145]. We look for approximation of complex model f by the simple model g in the neighborhood of our data point. In other words, we need to get a good approximation in local neighborhoods. We divide surrogate model methods for DL into the following three categories: local approximation-based explanations, rule-based explanations, and tree-based explanations. A summary of the surrogate model methods for explaining black box is presented in Table 5.

Local explanations (also known as proxy or per-decision explanations) provide a separate explanation for each decision of the model, i.e., they explain only a single prediction [121]. Local explanations can be further divided into approximation-based and example-based explanations. In approximation-based explanation (approach), new data points near the data point whose prediction needs to be explained from the model are sampled (explanation data points), and then a linear model, e.g., LIME [85], is fitted or a rule set is extracted. **Rule extraction based** on explanations attempts to extract rules from DNNs. One of the most commonly used rule formats consists of "IF...THEN" or "IF...ELSE" statements with "AND/OR" or "YES/NO" statements to make decisions [178]. The rules are merged to create a set of rules that can explain the behavior of the network based on its inputs [72]. **Tree-based explanations** use types of mimic learning to distill knowledge from DL models with tree structure. Decision trees are structured as graphs where the internal nodes represent conditional tests on input features, and the leaf nodes represent model outcomes. Rule extraction is a common method for extracting rules from tree modules [179].

Definition: **The explainability of the DL model can be illustrated by approximating the original DL model with a simple model with a proxy function.**

6.4.1. Perturbation-based surrogate explanations. Ribeiro *et al.* [85] propose the **local interpretable model-agnostic explainer (LIME)**, which is based on generating and classifying a large set of randomly perturbated input images and recognizing how the prediction changes. This method is based on a local approximation of the function of an individual model [180]. Authors show that the LIME method can be used to identify regions of the input that are most influential for a decision in a variety of model types and problem domains. LIME is used for explaining image segmentation, random sampling, and multiple linear model fitting [181]. The data is used to construct a local linear model that serves as a simplified proxy for the entire input [9]. LIME explains a prediction by fitting a localized linear model to approximate the classification boundary for a given prediction [182]. The LIME method does not require access to internal weights, activations, or other hyperparameters of the network [56]. The output of LIME is a list of explanations reflecting the contribution of each feature to the prediction of a data sample, i.e., superpixels with the highest positive weights as explanations. This provides local explanatory power and allows one to determine which feature changes will have the greatest impact on the prediction.

Ribeiro *et al.* [183] present another method that explains complex models with high-precision rules called "**anchors**". An anchor explanation is a rule that "anchors" the prediction locally enough – so that changes in the other feature values of the instance do not matter. The **counterfactual impact evaluation method (CIE)** is a local explanation method for comparing different predictions of the DNN [184]. The method explains why one decision is made instead of another [14]. A counterfactual explanation is an alternative input where the model's input differs from the given input [121]. The counterfactual explanation of a prediction can be defined as the smallest change in feature values, which changes the prediction to a predefined outcome [14]. The most striking feature of the model is the explanation of the reasons for a particular classification result [185]. The method for DNN can be used with any data type [14]. **SHapley additive exPlanations (SHAP)** values, also known as Shapely values, use coalition game theory to distribute the payoffs of a game. When applying SHAP values to a DL problem, the "game" is a prediction of a DL model, the "players" in the game are input variables for a given instance, and the "payoff" is equal, subtracting the base value. The method applies an additive feature attribution principle to create a model that is an explainable approximation of the original model. The SHAP values assign an importance value to each feature, basically the linear combination of input features, for a given prediction [160]. The SHAP values measure the average marginal effect of including inputs over all possible orders in which inputs can be included [139]. In short, the Shapley values use coalitions to see what contribution a feature value makes to the final prediction. They explain how to get from the base value $E[f(z)]$, which would be predicted if we knew no features of the current output $f(x)$, to the outcome. The Shapley explanation values for the input features always sum to the function $f(x)$ of the model. The values never decrease for an input x_i if the feature is changed so that x_i makes a strong contribution to $f(x)$ [186]. Lundberg and Lee [160] combine the DeepLIFT method and Shapely values under the name Deep SHAP

Table 5. A summary of the **surrogate model** methods for explaining black box.

Category	Method	Authors, year [Ref.]	Code	Model components
Perturbation-based surrogate explanation	Local interpretable model-agnostic explainer (LIME)	Ribeiro *et al.*, 2016 [85]	[link]	IMG, TXT, TAB, CL, PH, MA, L
	Anchors (ANCH)	Ribeiro *et al.*, 2018 [183]	[link]	IMG, TXT, TAB, CAT, CL, structural predictions, text generation, PH, MA, G
	SHapley additive exPlanations (SHAP)	Lundberg, Lee, 2017 [160]	[link]	ANY, numerical explanation, PH, MA, G, L
	Counterfactual impact evaluation (CIE)	Bottou *et al.*, 2013 [184]	CC [link]	causal inference
	Streaming weak submodular maximization (STREAK)	Elenberg *et al.*, 2017 [187]	[link]	IMG, CL, MA
	Knowledge distillation (KD)	Hinton *et al.*, 2015 [188]	CC [link]	G, CL
Graph-based explanation	And–or graph	Zhang *et al.*, 2017 [189]	NOC, NCC	IMG, CL, MA, CNN
	Extracting explanatory graph (EEG)	Zhang *et al.*, 2020 [190]	NOC	IMG, CL, MA, CNN
Rule-based explanation	Adversarial black-box explainer generating latent exemplars (ABELE)	Guidotti *et al.*, 2020 [191]	[link]	IMG, CF, L
	Rule extraction from deep neural networks (DeepRED)	Zilke *et al.*, 2016 [192]	NOC	producing intermediate rule sets
	Conjunctive normal form (CNF)	Su *et al.*, 2016 [194]	NOC	simplification, MA
	Rule extraction by reverse engineering (RxREN)	Augasta, Kathirvalavakumar, 2012 [193]	NOC	AH, G, CL, R
	Word importance scores (WIS)	Murdoch, Szlam, 2017 [195]	NOC	TXT, CL, question answering, SA, PH, G, LSTM
	Local rule-based explainer (LORE)	Guidotti *et al.*, 2018 [191]	CC [link]	DTree, CF, PH, MA, L, multilayer network
	RuleMatrix (RM)	Ming *et al.*, 2018 [196]	CC [link]	TAB, PH, MA, G/L

Table 5. [Cont.].

Category	Method	Authors, year [Ref.]	Code	Model components
Decision trees	Continuous/discrete rule extractor via decision tree induction (CRED)	Sato, Tsukimoto, 2001 [197]	NOC	
	Soft decision tree (SDT)	Frosst, Hinton, 2017 [198]	CC [link]	IMG, CL, CNN
	Model extraction (ME)	Bastani *et al.*, 2018 [200]	NOC	CL, REG, RL, PH, MA, G, neural nets
	Global additive explanations (GAE)	Tan *et al.*, 2018 [201]	NOC	TAB, CL/REG, CNN
	Automated reasoning (AR)	Bride *et al.*, 2018 [202]	NC	CL, PH, MA, G
	Partial aware local model (PALM)	Krishnan, Wu, 2017 [203]	NC	CL, PH, MA, G

For every method, there is a reference link, code link: no official code (NOC), community code (CC), no community code (NCC), data type on which it is possible to apply it: image (IMG), text (TXT), tabular (TAB), categorical data (CAD), problem: classification (CL), regression (R), explanation interface: saliency maps (SM), heatmaps (HM). If it is *post hoc* (PH) or *ante hoc* (AH) stage, model specific (MS) or model agnostic (MA), and local (L) or global (G) scope. Finally, for which type of DNN the method is appropriate: convolutional neural network (CNN).

approximating SHAP values under the assumption that the input features are independent and the deep model is linear. Deep SHAP combines computed SHAP values for smaller components into values for the entire network. There are other SHAP modifications [160], such as kernel SHAP and tree SHAP. Elenberg *et al.* [187] propose the **streaming weak submodular maximization (STREAK)** method for black-box classifiers as a combinatorial maximization problem where a streaming algorithm is used to maximize a weakly submodular objective function (function maximization). This method is similar and faster than LIME and uses the superpixel approach. First, the image is segmented into regions. Then, for a subset of these regions, a new image containing only these regions is fed into the model. The algorithm explains the given label of the image in the model. Another local approximation method is **knowledge distillation (KD)** [188], which uses knowledge compression of an ensemble into a single model to approximate the predictions of a complex ensemble learned using a surrogate model.

6.4.2. Graph-based explanations. Zhang *et al.* [189] propose the **and–or graph** method, where explanations are represented in terms of extracted reasoning logic. The model provides explanations by distilling them into object parts of a graph for a pre-trained CNN. After the semantic patterns in the input are extracted, the graph for the explanation is created. The nodes in the graph represent specific parts of the object's pattern, while the edges represent co-activations between sub-patterns. The graph explains the semantic hierarchy of CNN entities by representing which object parts (nodes) are activated and where the parts are located in the corresponding feature maps. Zhang *et al.* [190] propose a different approach to distill into a graph that extracts an explanatory graph for interpreting CNN that automatically disentangles object parts from each filter without annotations. The proposed method can be used to learn explanatory graphs for various CNNs, e.g., VGGs, residual networks, and the encoder of a VAE-GAN.

6.4.3. Rule-based explanations. The **adversarial black-box explainer generating latent Exemplars (ABELE)** [191] is an explainer based on local image rules. It provides an explanation for the reasons for the proposed classification by exploiting the latent space learned by an adversarial autoencoder for the neighborhood generation process. The **rule extraction from deep neural networks (DeepRED)** method [192] by Zilke *et al.* deals with feature extraction from a neural network with multiple hidden layers. Rules are extracted and merged for each layer in the DNN by using simplification and decomposition strategies. Another commonly used method for extracting rules is **rule extraction by reverse engineering RxREN** [193], which prunes input and applies an algorithm, the C4.5 logical model [24], a statistical method for building a parsimonious decision tree. **Conjunctive normal form (CNF)** or **disjunctive normal form (DNF)** [194] is a two-step Boolean rule-based classification extraction method that connects a complex model to one. Murdoch and Szlam [195] introduce the **word importance scores** method to track the importance of given text input to the LSTM for given explainable text output. As a result, a rule-based classifier is created using extracted phrases to visualize the important words. Guidotti *et al.* [191] propose a **local rule-based explainer (LORE)** for the black-box outcome for tabular data. This method provides explanations in the form of logical rules and counterfactual rules. The counterfactual rule set shows the conditions that can be varied at instance x to change the decision for output y. Ming *et al.* [196] propose **RuleMatrix** to explain ML models using rule surrogates and matrix-style visualization. The rule-based explanation is composed of several steps.

6.4.4. Tree-based explanations. The **CRED (continuous/discrete rule extractor via decision tree induction)** [197] method uses a decision tree approach to describe DNN behavior. This method has two steps, in each of which a decision tree is created. CRED is applied more for shallow networks than for deep networks. **Soft decision trees** [198] create binary trees of predetermined depth, where each branching node represents a hierarchical filter that influences the classification of input. As a first step toward explainable reinforcement learning, this method provides a form of explaining how deep control policy operates using a network distillation approach to transfer knowledge from a computationally complex environment into smaller, more explainable segments [188, 199]. Bastani *et al.* [200] propose the **model extraction (ME)** method for explaining the overall reasoning process performed by the model by approximation approach utilizing a much more explainable model. The explainers of the method are decision trees (DTs), which are generated by using the classification and regression tree algorithm (CART) logical model and Gaussian distribution fitted to the input data. Frosst and Hinton [198] propose **soft decision tree (SDT)** transforming a decision tree into DNN. The DNN is trained by stochastic gradient descent as a soft decision tree that mimics the input-output function. The decision tree makes hierarchical decisions based on input data and ultimately selects a particular statistical probability distribution over classes as its output. Tan *et al.* [201] propose **global additive explanation (GAE)**, distiling the black-box model into lower-dimensional components and study how the model parameters influence model performance. The method generates global additive explanations that describe the relationship between input features and model prediction. Other prominent methods worth to be mentioned are **automated reasoning (AR)** [202] and **partial aware local modeling (PALM)** [203].

7. Interpretability modeling

The **intrinsic** (also self-explained, predefined, or built-in), see Fig. 1h, approach aims to generate an interpretation of the model by rendering an explanation of the internal representations of DNNs along with their output using methods that are part of the DNN architecture [121]. In such an approach, the IML should be able to automatically offer human-friendly, interpretable explanations along with its predictions and be directly linked to domain knowledge [204]. Explanations should be intrinsic to the process of designing the model architecture and are, by definition, *ante hoc* [110]. This means that an explanation perspective is included in the neural network design and training before the neural network is modeled. This approach is, by definition, **model-specific** [2]. The design of self-explanatory DNN systems and models often consists of subnetworks called **modules** [205, 206] or special layers, e.g., the prototype layer in CNN, which should arrange tasks. Another term for modules found in the literature is a capsule. In capsules, each subnetwork is responsible for a specific high-level feature, and these features are combined to form even higher-level features, creating a word/parse tree structure [199, 207, 208]. Each module or such layer provides specific features for the interpretability of the whole DL system.

The intrinsic approach can be divided into the following interpretable method groups:

- **Attentional-based mechanism**. The attention mechanism deals with sequence-based tasks and uses a conditional distribution over a given input sequence of variable size by a weighted combination of all encoded input vectors, with the highest weights assigned to the most relevant vectors as attributes. When an attention mechanism is used to compute a representation of a single sequence, it is usually referred to as self-attention or intra-attention [208].
- **Joint training**. Joint training adds an “explanation task” to the original model task and trains the explanation task together with the original task. Joint training of the explanatory model means training the original model from DL on several different tasks simultaneously to optimize more than one function of the model. The additional task(s), called multi-tasks, are designed together with several types of formats for an explanation. The tasks can be organized directly or indirectly.

The attention-based mechanism group is divided into:

- **Monomodal interpretation. Single-module** or one-module interpretations consider the output of the attentional mechanism during a forward pass. The explainability score can be judged by how much attention is weighted by different input features at different stages of model inference. The multimodular explanatory approach attempts to divide the DNN structure into smaller parts of neural networks, with each part assigned to a specific task. Each independent neural network serves as a module and operates on separate inputs to perform a subtask of the tasks.

- **Multimodal interpretation.**

The **joint training** group is in the literature [121] divided into the interpretation of such additional tasks with:

- **Text explanations**. This type of intrinsic approach provides textual explanations in a natural language format.
- **Model prototyping**, Model prototyping is a task that explains the model based on the comparison between the model behavior and the prototype. These self-explanatory algorithms provide the class of model outputs as a weighted combination of prototypes. Prototype models are used for explainable computer vision, text and table classification tasks.
- **Association explanations**. Association explanation associates input features with objects and concepts that cannot be tested or interpreted by humans.

Definition: **The explainability of a DL model can be clarified by dividing a DNN into subnetworks (modules) by providing dedicated functionality for each subnetwork or by self-explanatory modeling through defined tasks.**

7.1. Attention-based mechanism

Table 6 presents a summary of the attention mechanism interpretation methods.

7.1.1. Monomodal interpretation. Vaswani *et al.* [209] present **image caption beneration with attention mechanism**, a "transformer" model for language translation as a sequence transduction model based on the transformer approach. The model is entirely built on a self-attention mechanism technique in encoder-decoder architecture adding additive model structures such as scaled dot-product. Recently Liu *et al.* [210] show that it is feasible to train the standard transformer with many layers. Using the adaptive model initialization (ADMIN) technique to stabilize training and unleash its full potential, they build very deep **transformer models for neural machine translation (NMT)** with up to 60 encoder layers and 12 decoder layers. Zhang *et al.* [211] propose the **self-attention generative adversarial network (SAGAN)** to model image generation tasks. SAGAN adds self-lttention layers to a general adversarial network (GAN) to strengthen both the generator and discriminator of the net-

Table 6. A summary of the **attention mechanism** explanation methods for explaining black box.

Category	Method	Authors, year [Ref.]	Code	Model components
Monomodal interpretation	Image captioning generation with attention mechanism (ICGAM)	Vaswani *et al.*, 2017 [209]	[link]	TXT, long translations, transformer
	Transformer models for neural machine translation (NMT)	Liu *et al.*, 2020 [210]	CC [link]	TXT, transformer
	Self-attention generative adversarial network (SAGAN)	Zhang *et al.*, 2019 [211]	[link]	IMG, CL, AM, CNN
	Simple neural attention meta-learner (SNAIL)	Mishra *et al.*, 2018 [212]	CC [link]	meta-learning, input: example-label pairs or observation-activation-reward tuples
	Visual analysis of transformer models (exBERT)	Hoover *et al.*, 2019 [213]	CC [link]	TXT, L, transformer
	Aspect-level sentiment classification (ASC)	He *et al.*, 2018 [214]	CC [link]	TXT, sentiment analysis, attention-based LSTM
	Bidirectional encoder representations from transformer (BERT)	Devlin *et al.*, 2019 [215]	[link]	TXT, language understanding, transformer
	Self-attention network (SANet)	Letarte *et al.*, 2018 [216]	CC [link]	TXT, sentiment analysis, self-attention network
	Reverse time attentIoN (RETAIN)	Choi *et al.*, 2017 [217]	[link]	sequential data

Table 6. [Cont.]

Category	Method	Authors, year [Ref.]	Code	Model components
Multimodal interpretation	Neural image caption with visual attention (NICVA)	Xu *et al.*, 2016 [218]	CC [link]	IMG, VAE, CNN + RNN/LSTM
	Multimodal interaction (MMI)	Vinyals *et al.*, 2015 [219]	CC [link]	IMG, VAE, CNN + RNN/LSTM comb
	Neural module network (NMN)	Andreas *et al.*, 2017 [205]	[link]	IMG, VQA, CNN + LSTM
	Explainable visual entailment (EVE)	Xie *et al.*, 2019 [220]	[link]	IMG, VQA, visual entitlement, CNN + MaskRNN + spec. modules
	Pointing and justification explanation (PJ-X)	Park *et al.*, 2016 [221]	NOC	IMG + TXT, AE, CNN + LSTM + spec. modules
	Transparency by design network (TbD-net)	Mascharka *et al.*, 2018 [222]	[link]	IMG, VQA, DNN + spec. modules
	Bottom-up and top-attention model (BTA)	Anderson *et al.*, 2018 [223]	[link]	IMG, TXT, VQA + captioning, faster R-CNN + LSTM
	Visual question answering model (VQAm)	Teney *et al.*, 2017 [224]	CC [link]	IMG, TXT, VQA

For every method, there is a reference link, code link: no official code (NOC), community code (CC), data type on which it is possible to apply it: image (IMG), text (TXT), embedding (EMB), categorical data (CAD), problem: classification (CL), explanation interface: saliency maps (SM), heatmaps (HM). If it is *post hoc* (PH) or *ante hoc* (AH) stage, model specific (MS) or model agnostic (MA), and local (L) or global (G) scope. Finally, for which type of DNN the method is appropriate: convolutional neural network (CNN) or, long-short term memory (LSTM).

work, improve the model's relationships between spatial regions, and capture global dependencies. The self-attention module in the image context is explicitly designed to learn and interpret the relationship between a pixel and all other pixels or pixel patches. Mishra *et al.* [212] introduce **simple neural attention meta-learner (SNAIL)** with a positioning option in the transformer model by combining the self-attention mechanism in "transformer" with temporal convolutions. SNAIL belongs to the field of meta-learning explanation methods and can be used not only for supervised learning but also for reinforcement learning tasks. Choi *et al.* [217] present a **RETAIN (reverse time attentIoN)** model used to explain time series. The model is based on a two-stage neural attention approach in which influential patterns from the past are detected and labeled. It uses two RNNs, each with an attentional mechanism responsible for explaining the focus of the neural network and how a choice was influenced. Other prominent solutions belonging to the group of monomodal explanations can be found in the works of Hoover *et al.* [213]: visual analysis of transfer models, He *et al.* [214]: the **aspect-level sentiment classification (ASC)**, Devlin *et al.* [215]: the bidirectional encoder **representations from transformer (BERT)** and Letarte *et al.* [216] present the **self-attention network (SANet),** an attention-based length-agnostic model for text classification.

7.1.2. Multimodal interpretation. Xu *et al.* [218] present the **neural image caption with visual attention (NICVA)** method for image caption generation that attempts to merge attentions. The algorithm of the method consists of two variants of the attention mechanism: a stochastic "hard" and a deterministic "soft" attention mechanism. The first is a stochastic "hard" attention mechanism that maximizes an appropriate lower bound. The second is a deterministic "soft" attention mechanism that uses backpropagation. A similar approach is proposed by Vinyals *et al.* [219]: the **multimodal interaction (MMI)** method. It combines computer vision and machine translation to generate natural language sentences to describe an image. This deep attention mechanism network consists of deep CNN for image classification tasks, followed by RNN language generation. An excellent example of a multimodal explanation is presented by Andreas *et al.* [205]: the **neural module network (NMN)**, which uses the VQA approach. The attention mechanism is applied to images to create the captions. The modularity of the proposed DNN architecture consists of the encoder-decoder approach. Xie *et al.* [220] propose **explainable visual entailment (EVE)**, which uses an attention mechanism to recover semantically meaningful regions across different feature spaces, e.g., between images and text. EVE combines image and ROI information to model fine-modal information. Such regions also correspond to the reason for a statement. Park *et al.* [221] present the **pointing and jus-**

tification explanation (PJ-X) method, which incorporates an explanatory attention mechanism to explain the responses to VQA tasks. The architecture of the multimodal explanation includes a textual justification consisting of two modules of two "pointing" mechanisms: answering with pointing and explaining with pointing. Other good solutions that belong to the multimodal explanation technique can be found in the work of Masharka *et al.* [222]: a transparency by design network (TbD-net) that assembles visual attention masks to answer a question about objects in a scene. Using these attentions, we can perform reasoning operations. Anderson *et al.* [223] implement a combined bottom-up and top attention model that can compute attention at the level of objects and other salient image regions. For each generated word, the model visualizes the attention weights on individual pixels and outlines the regions with the highest attention weight. Teney *et al.* [224] propose a model for visual question answering based on the principle of a joint embedding (GRU) of the input question and image, followed by a multi-label classifier over a set of response options.

7.2. Joint training

Table 7 presents a summary of the joint training interpretation methods.

7.2.1. Text explanations generation. Hendricks *et al.* [225] present the **visual explanation (VA)** method, which focuses on the discriminative properties of images, classifies images, and provides accurate textual explanations of why the image belongs to a particular class. This explanation-discriminative model generates images and provides class-relevant explanations for classification decisions. Their proposed model consists of the combination of two LSTM RNNs. The first RNN is trained on the image descriptions, i.e., generates words based on the previously generated word. The second one feeds the output of the first RNN with the image features and the image category predicted by the CNN and generates the next word based on this input. The architecture of the model consists of a recurrent explanation generator and a fine-grained visual feature extraction classifier. The visual explanation method describes the visual content of a particular image with appropriate information to explain why an image belongs to a particular category. Kim *et al.* [226] present the **attention alignment (AA)** method, which produces explanations in the form of attention maps with textual descriptions. This method consists of an attention-based video-to-text algorithm that generates textual explanations of the model as natural language text. The network architecture consists of two main components: controller and explanation matching. There are two approaches for directing attention: explanation with strongly directed attention (SAA) and explanation with weakly directed

Table 7. A summary of the **joint training** methods for explaining black box.

Category	Method	Authors, year [Ref.]	Code	Model components
Text explanation	Visual explanation (VA)	Hendricks *et al.*, 2016 [225]	NOC, CC [link]	IMG/CL, L, LRCN
	Attention alignment (AA)	Kim *et al.*, 2018 [226]	[link]	IMG/control planning, CNN
	Generative explanation framework (GEF)	Liu *et al.*, 2019 [227]	NOC	TXT/CL, encoder predictor
	Visual commonsense reasoning (VCR)	Zellers *et al.*, 2019 [228]	CC [link]	IMG, TXT, R2C (recognition to cognition networks)
Model prototyping	PrototypeDL (PDL)	Li *et al.*, 2017 [229]	[link]	IMG/CL, autoencoder + prototyping layer combination
	ProtoPNet (PPN)	Chen *et al.*, 2019 [230]	[link]	ANY, PR, SM, G, CNN + prototyping layer
	HP net model (HPm)	Hase *et al.*, 2019 [231]	[link]	
Interpretation association	Rationales (RTL)	Lei *et al.*, 2016 [232]	[link]	TXT, SenA, encoder-generator
	Self-explaining neural network (SENN)	Alvarez-Melis, Jaakkola, 2018 [233]	[link]	IMG, CAT, TAB, self-explainable NN
	Video captioning (VC)	Dong *et al.*, 2017 [234]	NOC	video, captioning, encoder/decoder

For every method, there is a reference link, code link: no official code (NOC), community code (CC), data type on which it is possible to apply it: image (IMG), embedding (EMB), categorical data (CAD), problem: classification (CL), explanation interface: saliency maps (SM), heatmaps (HM). If it is *post hoc* (PH) or *ante hoc* (AH) stage, model specific (MS) or model agnostic (MA), and local (L) or global (G) scope. Finally, for which type of DNN the method is appropriate: convolutional neural network (CNN).

attention (WAA). The model predicts the control commands for the vehicle, i.e., a change in course, at each time step, whereas an explanation model generates natural language explanations for the justifications. For example, an explanation consists of an action description, e.g., "the car is driving in the left lane", supported by an action explanation, e.g., "to pass the school bus", and a visual explanation in the form of an attention map. Liu *et al.* [227] propose the **generative explanation framework (GEF)**, which makes classification decisions while generating textual explanations for feature labels. The "explanatory factor" in the model structure is intended to help make the generated explanations class-specific. Other prominent solutions belonging to text and image explanation methods can be found in the work of Zellers *et al.* [228] **visual commonsense reasoning (VCR)**. The purpose of this method is to provide explanations in a multichoice manner. The model, called recognition to cognition (R2C), is given an image with its objects, a semantic question, and four answer choices. The model must decide which answer is correct and provide interpretation accordingly. It must decide which is the best rationale that explains why its answer is correct.

7.2.2. Model prototyping. Model prototyping or prototyping-based reasoning is specifically designed for classification tasks and traces back to a classical form of case-based reasoning, e.g., nearest neighbor-based technique and, prototype-based technique [230]. Prototype-based reasoning refers to the task of predicting future events (i.e., new patterns) based on particularly informative known data points. A prototype-based classifier generates classifications based on the similarity between the given input and each prototypical observation in the dataset [229]. This is usually done by identifying prototypes, i.e., representative examples that are used to make a prediction. These methods are inspired by the fact that predictions are based on individual, previously seen examples that mimic human decision-making. In prototype classification, prototypes are not limited to one observation in the dataset, but can be generalized to a combination of multiple observations or a latent representation learned in the feature space [121]. To provide intrinsic explanations, the model architecture is designed to allow joint training of prototypes along with the original task. Model explainability is achieved by tracing the reasoning path for the given prediction to each prototype learned by the model.

Li *et al.* [229] propose an explainable prototype image classifier **prototypeDL (PDL)**, whose predictions are based on the principle of similarity of input to a small set of prototypes learned during neural network training. Observations are classified based on their proximity to a prototypical observation. The model contains the transformed input, which is derived through encoding, and the input reconstructed by decoding, which is used for prototyping the classifier. In addition, the prototype classifier network, consisting of the prototype layer p, the fully connected layer w, and the softmax layer s, generates the classification. Each prototype can be visualized by the decoder through the training process. Chen *e al.* [230] present the **prototypical part network (ProtoPNet)** method for image classification tasks. The network dissects the initial image by finding prototypical parts and combining the knowledge from the prototypes to make a final classification. The architecture of the explainable DNN consists of four components: a standard CNN, a prototype layer (a prototype classifier), a fully connected layer, and the output. ProtoPNet is not only able to reveal the parts of the input it is looking at, but also points us to prototypical cases that are similar to those parts. The prototypical layer, where the source image is learned, extracts various features of the output image. These are then compared with the training images to produce a recognition map indicating the similarity between the images. The prototype classifier generates model predictions based on the weighted sum of the individual similarity score between the image patch and a learned prototype. ProtoPNet has a built-in case-based reasoning process that generates explanations during classification. More recently, Hase *et al.* [231] have presented an **HP net** model that uses hierarchically organized prototypes to find explanations for predicting an image at each level of the class taxonomy.

7.2.3. Interpretation associations. Lei *et al.* [232] combine two modular components: the generator and the encoder, to extract parts of the input text called rationales and try to solve a sentiment classification problem. The generator gives a distribution over possible rationales, e.g., which text fragments might be candidates for rationales. Both the generator and the coder are trained together to obtain explanations using different regularizations. The coder assigns task-specific values to the text for prediction. The model's explanation is associated with a set of critical input words for prediction. Alvarez-Melis and Jaakkola [233] propose the **self-explaining neural network (SENN)** method, which is based on learnable explainable concepts that link input features to semantically based concepts. SENN gives a raw input that learns to generate both class prediction and explanations as input feature-to-feature mapping. SENN consists of three components. The first is a concept encoder that converts the input features into a small set of explainable features. The second is an input-dependent parameter that generates attribution scores. Finally, the third component is an aggregation function that produces a prediction. Dong *et al.* [234] provide an attentive encoder/decoder framework for video recording tasks that can automatically learn explainable features of each neuron of the deep network associated with a "topic". For example, the topic "road" would be a road with related words such as road, vehicle, or pedestrian.

8. Explanation toolboxes

During the last few years, a significant number of toolboxes for DL explanations have been proposed. A detailed collection of XAI toolboxes was referred to in 2018 by Alber *et al.* [235], in 2019 by Arya *et al.* [27], Nori *et al.* [236], and Spinner *et al.* [237], in 2020 by Das *et al.* [119], and in 2021 by Linardatos *et al.* [114], see Table 8.

Table 8. XAI Toolkits.

Software package	Available from
AI Fairness 360 (AIF360)	https://github.com/Trusted-AI/AIF360
AI Explainability 360 (AIX360)	https://github.com/Trusted-AI/AIX360
Alibi Explain	https://github.com/SeldonIO/alibi
Analysis by synthesis (ABS)	https://github.com/bethgelab/AnalysisBySynthesis
Captum	https://github.com/pytorch/captum
DALEX	https://github.com/ModelOriented/DALEX
DeepExplain	https://github.com/marcoancona/DeepExplain
Deep visualization tool	https://github.com/yosinski/deep-visualization-toolbox
ELI5	https://github.com/TeamHG-Memex/eli5
explainX	https://github.com/explainX/explainx
FAT Forensics	https://github.com/fat-forensics/fat-forensics
InterpretML	https://github.com/interpretml/interpret
iNNvestigate	https://github.com/albermax/innvestigate
H2O.ai	https://github.com/h2oai/mli-resources
L2X	https://github.com/Jianbo-Lab/L2X
Rectified gradient	https://github.com/1202kbs/Rectified-Gradient
Saliency relevance propagation	https://github.com/Hey1Li/Salient-Relevance-Propagation
Sensitivity analysis library (SALib)	https://github.com/SALib/SALib
Skater	https://github.com/oracle/Skater
tfexplain	https://github.com/sicara/tf-explain
treeinterpreter	https://pypi.org/project/treeinterpreter/
XAI	https://github.com/EthicalML/xai

9. Conclusions

The proposed classification for Deep Learning interpretability and explainability is based on and reflects the current discourse on interpretability and explainability principles, theories and methods. Explanation methods are a promising approach to uncover the hidden intelligence of how DNNs work. In the context of the complexity behind these methods, a lack of knowledge about which explainable method or which interpretable model is best suited for implementation, as well as a low level of abstraction, as many explainability methods, involve several explainable techniques, may prevent human users from using and implementing them for practical applications and research. To mitigate these shortcomings, it is important to provide a systematic classification of interpretability and explainability. With this in mind, we focused on three specific problems of explainability previously identified to enrich existing studies of explainability.

The first concern was to provide specific definitions for the concepts of interpretability and explainability in DL. Here we showed a reasonable argument to distinguish these two main concepts from XAI and IML. The second concern was to establish a classification for explaining and interpreting DL systems and models. Here we presented approaches, models and methods that form a classification for explaining and interpreting DL, especially DNNs. We hope that the proposed classification distills the important issues, related work, methods, and concerns related to the explainability of black-box models and creating interpretable glass-box. The third concern is to describe the definitions of each category of a proposed classification for explainable and interpretable DL.

This article did not provide a definitive answer to all problems of explainable and interpretable DL. For example, state-of-the-art performance across the range of DNNs understanding studies requires a variety of different explainability topologies, i.e., there is not one “best explainer” for all deep networks. We see several challenges for the future, not only for the exploration of explainable and interpretable DL but also for a comprehensive understanding and trust in the nature of transparent AI. The first issue concerns the human user: the use of explanations must be human-machine friendly interaction. The second issue is fairness. The introduction of these concepts at each step of deep modeling must be considered by design. The third issue is the evaluation of the quality and metrics of explanations. Another problem is the low level of abstraction of the explanation, i.e., the use of abstractions to simplify explanations. Finally, there is a need for a general theory of explainable AI: what approaches and techniques can be developed for well-developed explanation methods.

Ultimately, both deep model end-users and the DL research community should develop new, more friendly explainability models, methods and techniques in the future, and integrate explanations into a comprehensive AI system to mitigate complexity and improve system performance. We hope that the proposed explainable DL classification will be actively used as a step toward trustworthy AI and, as a prerequisite for artificial general intelligence.

Other challenges of XAI that are outside the scope of this article include objective metrics, assessing the quality of explanations, criteria for a good explanation, and how to recognize the importance of a feature that contributed to a particular prediction or that is locally relevant to that prediction. For example, the remove and retrain (ROAR) approach proposed by Hooker *et al.* [29] could be a reliable feature estimator. There is also great potential for intrinsic methods.

Acknowledgments

This work was funded by ERDF project No. 1.2.1.1/18/A/006 research No.1.5 "Efficient module for automatic detection of people and transport using surveillance cameras".

References

1. D. Gunning, *Explainable artificial intelligence (XAI)*, Technical Report, Defense Advanced Research Projects Agency (DARPA), 2017.
2. A. Adadi, M. Berrada, Peeking inside the black-box: A survey on explainable artificial intelligence (XAI), *IEEE Access*, **6**: 52138–52160, 2018.
3. A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems*, **25**: 1097–1105, 2012.
4. Y.A. LeCun, Y. Bengio, G.E. Hinton, Deep learning, *Nature*, **521**(7553): 436–444, 2015.
5. W. Xiong, L. Wu, F. Alleva, F. Droppo, X. Huang, A. Stolcke, The Microsoft 2017 conversational speech recognition system, [in:] *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5934–5938, 2018.
6. A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, [in:]: *IEEE CVPR*, pp. 1725–1732, 2014.
7. E. Tjoa, C. Guan, A survey on explainable artificial intelligence (XAI): Towards medical XAI, *arXiv*, 2019, arXiv:1907.07374v5.
8. G. Montavon, Gradient-based vs. propagation-based explanations: An axiomatic comparison, [in:] W. Samek *et al.* [Eds.], *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, Springer, pp. 253–265, 2019.
9. L.H. Gilpin, D. Bau, B.Z. Yuan, A. Bajwa, M. Specter, L. Kagal, Explaining explanations: An approach to evaluating interpretability of machine learning, *arXiv*, 2019, arXiv:1806.00069v3.
10. J. Seo, J. Choe, J. Koo, S. Jeon, B. Kim, T. Jeon, Noise-adding methods of saliency map as series of higher order partial derivative, *arXiv*, 2018, arXiv:1806.03000v1.
11. D. Castelvecchi, Can we open the black box of AI?, *Nature News*, **538**(7623): 20, 2016.
12. H. Lakkaraju, R. Caruana, E. Kamar, J. Leskovec, Interpretable & explorable approximations of black box models, *arXiv*, 2017, arXiv:1707.01154v1.

13. D.W. Apley, J. Zhu, Visualizing the effects of predictor variables in black box supervised learning models, *arXiv*, 2019, arXiv:1612.08468v2.

14. V. Burhrmester, D. Münch, M. Arens, Analysis of explainers of black box deep neural networks for computer vision: A survey, *arXiv*, 2019, arXiv:1911.12116v1.

15. C. Rudin, Stop explaining black box machine learning models for high stake decisions and use interpretable models instead, *arXiv*, 2019, arXiv:1811.10154v3.

16. D. Pedreshi, F. Giannotti, R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, Meaningful explanations of black box AI decision systems, [in:] *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

17. A. Rai, Explainable AI: From black box to glass box, *Journal of the Academy of Marketing Science*, **48**, 137–141, 2020, doi: 10.1007/s11747-019-00710-5.

18. W. Samek, G. Montavon, A. Vedaldi, L.K. Hansen, K.-R. Müller [Eds.], *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, Springer, 2019.

19. W. Landecker, M.D. Thomure, L.M.A. Bettencourt, M. Mitchell, G.T. Kenyon, S.P. Brumby, Interpreting individual classifications of hierarchical networks, [in:] *IEEE Symposium on Computational Intelligence*, pp. 32–38, 2013.

20. P. Chen, W. Dong, J. Wang, X. Lu, U. Kaymak, Z. Huang, Interpretable clinical prediction via attention-based neural network, *BMC Medical Informatics and Decision Making*, **20**(Suppl 3): 131, 2020, doi: 10.1186/s12911-020-1110-7.

21. Y. Shen *et al.*, To explain or not to explain: A study on the necessity of explanations for autonomous vehicles, *arXiv*, 2020, arXiv:2006.11684v1.

22. D. Slack, S. Hilgard, E. Jia, S. Singh, H. Lakkaraju, Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods, *arXiv*, 2020, arXiv:1911.02508v2.

23. A. Holzinger, Interactive machine learning for health informatics: When do we need the human-in-the-loop?, *Brain Informatics*, **3**: 119–131, 2016, doi: 10.1007/s40708-016-0042-6.

24. K.R. Varshney, Trustworthy machine learning and artificial intelligence, *ACM XRDS Magazine*, **25**(3): 26–29, 2019.

25. F. Doshi-Velez, B. Kim, Towards a rigorous science of interpretable machine learning, *arXiv*, 2017, arXiv:1702.08608v2.

26. W. Yuan, P. Liu, G. Neubig, Can we automate scientific reviewing?, *arXiv*, 2021, arXiv:2102.00176v1.

27. V. Arya *et al.*, One explanation does not fit all: A toolkit and taxonomy of AI explainability techniques, *arXiv*, 2019, arXiv:1909.03012v2.

28. A.B. Arrieta *et al.*, Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities, and challenges towards AI, *Information Fusion*, **58**: 82–115, 2020.

29. S. Hooker, D. Erhan, P.-J. Kindermans, B. Kim, A benchmark for interpretability methods in deep neural networks, [in:] H. Wallach *et al.* [Eds.], *Advances in Neural Information Processing Systems*, Vol. 32, 2019.

30. U. Bhatt, A. McKane, A. Weller, A. Xiang, Machine learning explainability for external stakeholders, [in:] *IJCAI – PRICAI Workshop on Explainable Artificial Intelligence (XAI)*, January 8, 2021.

31. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, [in:] *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

32. J. Wang, M. Ren, I. Bogunovic, Y. Xiong, R. Urtasun, Cost-efficient online hyperparameter optimization, *arXiv*, 2021, arXiv:2101.06590v1.

33. Y. LeCun, L. Bottou, L. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, **86**(11): 2278–2324, 1998.

34. B.B. Traore, B. Kamsu-Foguem, F. Tangara, Deep convolution neural network for image recognition, *Ecological Informatics*, **48**: 257–268, 2018.

35. A. Esteva *et al.*, Dermatologist-level classification of skin cancer with deep neural networks, *Nature*, **542**: 115–118, 2017.

36. M. Schwarz, A. Milan, A.S. Periyasamy, S. Behnke, RGB-D object detection and semantic segmentation for autonomous manipulation in clutter, *The International Journal of Robotics Research*, **37**(4–5): 437-451, 2018, doi: 10.1177/0278364917713117.

37. J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, [in:] *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

38. Z. Yang, Y. Yuan, Y. Wu, R. Salakhutdionov, W.W. Cohen, Review networks for caption generation, *arXiv*, 2016, arXiv:1605.07912v2.

39. J. Johnson, A. Karpathy, L. Fei-Fei, DenseCap: Fully convolutional localization networks for dense captioning, [in:] *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

40. H. Gao *et al.*, Are you talking to a machine? Dataset and methods for multilingual image question answering, [in:] *NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems*, Vol. 2, pp. 2296–2304, 2015.

41. M. Ren, R. Kiros, R. Zemel, Exploring models and data for image question answering, [in:] *Advances in Neural Information Processing Systems 28 (NIPS)*, pp. 1–9, 2015.

42. S. Antol *et al.*, VQA: Visual question answering, [in:] *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

43. M. Malinowski, M. Rohrbach, M. Fritz, Ask your neurons: A neural-based approach to answering questions about images, [in:] *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

44. D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, A. Farhadi, IQA: Visual question answering in interactive environments, *arXiv*, 2017, arXiv:1712.03316.

45. A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, D. Batra, Embodied question answering, [in:] *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

46. H. de Vries, F. Strub, S. Chandar, O. Pietquin, H. Larochelle, A.C. Courville, Guess what?! Visual object discovery through multi-modal dialogue, [in:] *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

47. A. Das *et al.*, Visual dialog, [in:] *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

48. S.A. Bargal, A. Zunino, D. Kim, J. Zhang, V. Murion, S. Sclaroff, Excitation backprop for RNNs, *arXiv*, 2018, arXiv:1711.06778v3.

49. S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Computation*, **9**(8): 1735–1780, 1997.

50. B. DuSell, D. Chiang, Learning context-free languages with nondeterministic stack RNNs, *arXiv*, 2020, arXiv:2010.04674v1.

51. M. Venzke, D. Klish, P. Kubik, A. Ali, J.D. Missier, Artificial neural networks for sensor data classification on small embedded systems, *arXiv*, 2020, arXiv:2012.08403v1.

52. L. Arras, G. Montavon, K.-R. Müller, W. Samek, Explaining recurrent neural network predictions in sentiment analysis, [in:] *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 159–168, 2017, doi: 10.18653/v1/W17-5221.

53. S. Tao, Deep neural network ensambles, *arXiv*, 2019, arXiv:1904.05488v2.

54. W. Samek, T. Wiegand, K.-R. Müller, Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models, *arXiv*, 2017, arXiv:1708.08296v1.

55. P. Angelov, E. Soares, Towards explainable deep neural networks (xDNN), *Neural Networks*, **130**: 185–194, 2020, doi: 10.1016/j.neunet.2020.07.010.

56. S.J. Oh, M. Augustin, B. Schiele, M. Fritz, Towards reverse-engineering black-box neural networks, [in:] *9th International Conference on Learning Representations*, Vancouver, Canada, 30 April – 3 May 2018.

57. Z.C. Lipton, The mythos of model interpretability, *arXiv*, 2017, arXiv:1606.03490v3.

58. A. Holzinger, M. Plass, K. Holzinger, G.C. Crişan, C.-M. Pintea, V. Palade, A glass-box interactive machine learning approach for solving NP-hard problems with the human-in-the-loop, *arXiv*, 2017, arXiv:1708.01104.

59. P.W. Koh, P. Liang, Understanding black-box predictions via influence functions, [in:] *Proceedings of the 34th International Conference on Machine Learning (ICML)*, Sydney, Australia, Vol. 70, pp. 1885–1894, August 6–11, 2017.

60. D. Mascharka, P. Tran, R. Soklaski, A. Majumdar, Transparency by design: Closing the gap between performance and interpretability in visual reasoning, *arXiv*, 2018, arXiv:1803.05268v2.

61. V. Beaudouin *et al.*, Flexible and context-specific AI explainability: A multidisciplinary approach, *arXiv*, 2020, arXiv:2003.07703v1.

62. K. Sokol, P. Flach, Explainability fact sheets: A framework for systematic assessment of explainable approaches, [in:] *FAT* '20: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 56–67, 2020, doi: 10.1145/3351095.3372870.

63. F. Xu, H. Uszkoreit, Y. Du, W. Fan, D. Zhao, J. Zhu, Explainable AI: A brief survey on history, research areas, approaches and challenges, [in:] J. Tang *et al.* [Eds.], *Natural Language Processing and Chinese Computing (NLPCC)*, Lecture Notes in Computer Science, Springer, Cham, Vol. 11839, 2019, doi: 10.1007/978-3-030-32236-6_51.

64. N.C. Thompson, K. Greenwald, K. Lee, G.F. Manso, The computational limits of deep learning, *arXiv*, 2020, arXiv:2007.05558v1.

65. S. Liu, X. Wang, M. Liu, J. Zhu, Towards better analysis of machine learning models: A visual analytics perspective, *Visual Informatics*, **1**(1): 48–56, 2017, doi: 10.1016/j.visinf.2017.01.006.

66. F.F.J. Kameni, N. Tsopze, Simplifying explanation of deep neural networks with sufficient and necessary feature-sets: Case of text classification, *arXiv*, 2020, arXiv:2010.03724v2.

67. D. Erhan, A. Courville, Y. Bengio, *Understanding representations learned in deep architectures*, Department d'Informatique et Recherche Operationnelle, University of Montreal, QC, Canada, 2010.

68. M. Du, N. Liu, X. Hu, Techniques for interpretable machine learning, *arXiv*, 2019, arXiv:1808.00033v3.

69. S. Watcher, B. Mittelstadt, L. Floridi, Why a right to explanation of automated decision-making does not exist in the General Data Protection Regulation, *International Data Privacy Law*, **7**(2): 76–99, 2017.

70. D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, G.-Z. Yang, XAI – Explainable artificial intelligence, *Science Robotics*, **4**(37): eaay7120, 2019, doi: 10.1126/scirobotics.aay7120.

71. A. Holzinger, G. Langs, H. Denk, K. Zatlouk, H. Müller, Causability and explainability of artificial intelligence in medicine, *WIREs Data Mining Knowledge Discovery*, **9**(4): e1312, 2019, doi: 10.1002/widm.1312.

72. G. Ras, M. van Gerven, P. Haselager, Explanation methods in deep learning: Users, values, concerns and challenges, *arXiv*, 2018, arXiv:1803.07517v2.

73. T. Kulesza, M. Burnett, W. Wong, S. Stumpf, Principles of explanatory debugging to personalize interactive machine learning, [in:] *Proceedings of the 20th International Conference on Intelligent User Interfaces (ACM)*, pp. 126–137, 2015.

74. D. Wang, Q. Yang, A. Abdul, B.Y. Lim, Designing theory-driven user-centric explainable AI, [in:] *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (ACM)*, Paper no. 601, pp. 1–15, 2019.

75. C.C.S. Liem *et al.*, Psychology meets machine learning: Interdisciplinary perspectives on algorithmic job candidate screening, [in:] *Explainable and Interpretable Models in Computing Vision and Machine Learning*, H.J. Escalante *et al.* [Eds.], Springer, Cham, pp. 197–253, 2018.

76. J. Hestness *et al.*, Deep learning scaling is predictable, empirically, *arXiv*, 2017, arXiv:1712.00409v1.

77. B. Ginsburg *et al.*, Training deep networks with stochastic gradient normalized by layer-wise adaptive second moments, [in:] OpenReview.net, 2019.

78. G. Montavon, W. Samek, K.-R. Müller, Methods for interpreting and understanding deep neural networks, *Digital Signal Processing*, **73**: 1–15, 2018, doi: 10.1016/j.dsp.2017.10.011.

79. T. Miller, Explanation in artificial intelligence: Insight from the social sciences, *Artificial Intelligence*, **267**: 1–38, 2019, doi: 10.1016/j.artint.2018.07.007.

80. R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, [in:] *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, June 23–28, 2014.

81. M. Ancona, E. Ceolini, C. Özitreli, M. Gross, Towards better understanding of gradient-based attribution methods for deep neural networks, *arXiv*, 2018, arXiv:1711.06104v4.

82. D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, [in:] D.E. Rumelhart, J.L. McClelland [Eds.], *Parallel Distributing Processing*, MIT Press, pp. 318–362, 1986.

83. P.-J. Kindermans *et al.*, The (un)realibility of saliency methods, [in:] W. Samek *et al.* [Eds.], *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, LNCS, Springer, Cham, Vol. 11700, pp. 267–280, 2019.

84. R. Roscher, B. Bohn, M.F. Duarte, J. Garcke, Explainable machine learning for scientific insights and discoveries, *arXiv*, 2020, arXiv:1905.08883v3.

85. M.T. Ribeiro, S. Singh, C. Guestrin, Why should I trust you?: Explaining the predictions of any classifier, [in:] *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, California, USA, pp. 1135–1144, August 13–17, 2016.

86. M. Alber, Software and application patterns for explanation methods, *arXiv*, 2019, arXiv:1904.04734v1.

87. K.R. Varshey, H. Alemzadeh, On the safety of machine learning: Cyber-physical systems, decision science, and data products, *arXiv*, 2017, arXiv:1610.01256v2.

88. X. Zhao *et al.*, A safety framework for critical systems utilising deep neural networks, *arXiv*, 2020, arXiv:2003.05311v3.

89. A. Weller, Transparency: Motivations and challenges, [in:] W. Samek *et al.* [Eds.], *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, Springer, pp. 23–40, 2019.

90. M. Raghu, E. Schmidt, A survey of deep learning for scientific discovery, *arXiv*, 2020, arXiv:2003.11755v1.

91. L.A. Hendricks, A. Rohrbach, B. Schiele, T. Darrell, Generating visual explanations with natural language, *Applied AI Letters*, **2**(4): e55, pp. 1–16, 2021.

92. J.M. Oramas, K. Wang, T. Tuytelaars, Visual explanation by interpretation: Improving visual feedback capabilities of deep neural networks, *arXiv*, 2019, arXiv:1712.06302v3.

93. J. Kaplan *et al.*, Scaling laws for natural language models, *arXiv*, 2020, arXiv:2001.08361v1.

94. G.G. Towell, J.W. Shavlik, Extracting refined rules from knowledge-based neural networks, *Machine Learning*, **13**: 71–101, 1993.

95. C. Molnar, Interpretable machine learning: A guide for making black box models explainable, 1st ed., Leanpub, https://christophm.github.io/interpretable-ml-book/, accessed on 08.01.2021.

96. S. Kim, M. Jeong, B.C. Ko, Interpretation and simplification of deep forest, *arXiv*, 2020, arXiv:2001.04721v1.

97. W.-J. Nam, S. Gur, J. Choi, L. Wolf, S.-W. Lee, Relative attribution propagation: Interpreting the comparative contribution of individual units in deep neural networks, *arXiv*, 2019, arXiv:1904.00605v4.

98. L.K. Hansen, L. Rieger, Interpretability in intelligent systems – A new concept?, [in:] W. Samek *et al.* [Eds.], *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, LNCS, Springer, Cham, Vol. 11700, pp. 41–49, 2019.

99. R. Fong, A. Vedaldi, Explanations for attributing deep neural network prediction, [in:] W. Samek *et al.* [Eds.], *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, LNCS, Springer, Cham, pp. 149–168, 2019.

100. Q.V. Liao, D. Gruen, S. Miller, Questioning the AI: Information design practice for explainable AI user experiences, *arXiv*, 2020, arXiv:2001.02478v2.

101. W.J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asi, B. Yu, Interpretable machine learning: Definitions, methods, and applications, *arXiv*, 2019, arXiv:1901.04592v1.

102. Z. Yang, A. Zhang, A. Sudjianto, Enhancing explainability of neural networks through architecture constraints, *arXiv*, 2019, arXiv:1901.03838v2.

103. J. Vaughan, A. Sudjianto, E. Brahimi, J. Chen, V.N. Nair, Explainable neural networks based on additive index models, *arXiv*, 2018, arXiv:1806.01933v1.

104. S. Chauhan, L. Vig, M. De Filippo De Grazia, M. Corbetta, S. Ahmad, M. Zorzi, A comparison of shallow and deep learning methods for predicting cognitive performance of stroke patients from MRI lesion images, *Frontiers in Neuroinformatics*, **13**: 53, 2019, doi: 10.3389/fninf.2019.00053.

105. N. Tintarev, *Explaining recommendations*, PhD Dissertation, University of Aberdeen, 2009.

106. G. Chrysostomou, N. Alertas, Improving the faithfulness of attention-based explanations with task-specific information for text classification, *arXiv*, 2021, arXiv:2105. 02657v2.

107. G. Vilone, L. Longo, Explainable artificial intelligence: A systematic review, *arXiv*, 2020, arXiv:2006.00093v3.

108. A. Papenmeier, G. Englebienne, C. Seifert, How model accuracy and explanation fidelity influence user trust, *arXiv*, 2019, arXiv:1907.12652v1.

109. H. Harutyunyan *et al.*, Estimating informativeness of samples with smooth unique information, *arXiv*, 2021, arXiv:2101.06640v1.

110. S. Rivera, J. Klipfel, D. Weeks, Flexible deep transfer learning by separate feature embeddings and manifold alignment, *arXiv*, 2020, arXiv:2012.12302v1.

111. R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, D. Pedreschi, F. Giannotti, A survey of methods for explaining black box models, *arXiv*, 2018, arXiv:1802.01933v3.

112. U. Kamath, J. Liu, *Explainable artificial intelligence: An introduction to interpretable machine learning*, Springer, 2021.

113. F. Bodria, F. Giannotti, R. Guidotti, F. Naretto, D. Pedreschi, S. Rinzivillo, Benchmarking and survey of explanation methods for black box models, *arXiv*, 2021, arXiv:2102.13076v1.

114. P. Linardatos, V. Papastefanopoulos, S. Kostianstis, Explainable AI: A review of machine learning interpretability methods, *Entropy*, **23**(1): 18, 2021, doi: 10.3390/e23010018.

115. J. Zhou, A.H. Gandomi, F. Chen, A. Holzinger, Evaluating the quality of machine learning explanations: A survey on methods and metrics, *Electronics*, **10**(5): 593, 2021, doi: 10.3390/electronics10050593.

116. V. Belle, I. Papantonis, Principles and practice of explainable machine learning, *arXiv*, 2020, arXiv:2009.11698v1.

117. O. Benchekroun, A. Rahimi, Q. Zhang, T. Kodliuk, The need for standardized explainability, *arXiv*, 2020, arXiv:2010.11273v2.

118. S. Chari, O. Seneviratne, D.M. Gruen, M.A. Foreman, A.K. Das, D.L. McGuinness, Explanation ontology: A model of explanations for user-centered AI, *arXiv*, 2020, arXiv:2010.01479v1.

119. A. Das, P. Rad, Opportunities and challenges in explainable artificial intelligence (XAI): A survey, *arXiv*, 2020, arXiv:2006.11371v2.

120. P. Hase, M. Bansal, Evaluating explainable AI: Which algorithmic explanations help users predict model behavior?, *arXiv*, 2020, arXiv:2005.01831v1.

121. N. Xie, G. Ras, M. van Gerven, D. Doran, Explainable deep learning: A field guide for the uninitiated, *arXiv*, 2020, arXiv:2004.14545v1.

122. D.V. Carvalho, E.M. Pereira, J.S. Cardoso, Machine learning interpretability: A survey on methods and metrics, *Electronics*, **8**(8): 832, 2019, doi: 10.3390/electronics8080832.

123. K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, *arXiv*, 2013, arXiv:1312.6034.

124. L.M. Zintgraf, T.S. Cohen, S. Adel, M. Welling, Visualizing deep neural network decisions: Prediction difference analysis, *arXiv*, 2017, arXiv:1702.04595v1.

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

126. D. Bau, J.-Y. Zhu, H. Strobelt, A. Lapedriza, B. Zhou, A. Torralba, Understanding the role of individuals units in a deep neural network, *arXiv*, 2020, arXiv:2009.05041v2.

127. S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, K.-R. Müller, Unmasking Clever Hans predictors and assessing what machines really learn, *Nature Communications*, **10**: 1096, 2019.

128. D. Bau, B. Zhou, A. Khosla, A. Oliva, A. Torralba, Network dissection: Quantifying interpretability of deep visual representations, *arXiv*, 2017, arXiv:1704.05796v1.

129. A. Lucieri, M.N. Bajwa, S.A. Braun, M.I. Malik, A. Dengel, S. Ahmed, On interpretability of deep learning based skin lesion classifiers using concept activation vectors, *arXiv*, 2020, arXiv:2005.02000v1.

130. D. Smilkov, N. Thorat, B. Kim, F. Viégas, M. Wattenberg, SmoothGrad: Removing noise by adding noise, *arXiv*, 2017, arXiv:1706.03825v1.

131. R.R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, *arXiv*, 2016, arXiv:1610.02391.

132. M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, *arXiv*, 2017, arXiv:1703.01365v2.

133. M. Munir, S.A. Siddiqui, F. Kusters, D. Mercier, A. Dengel, S. Ahmed, TSXplain: Demystification of DNN decisions for time-series using natural language and statistical features, *arXiv*, 2019, arXiv:1905.06175.

134. Z. Zhang, Y. Xie, F. Xing, M. McGough, L. Yang, MDNet: A semantically and visually interpretable medical image diagnosis network, [in:] *Proceedings of the IEEE Conference*

on Computer Vision and Pattern Recognition (CVPR), Honolulu, USA, pp. 6428–6436, July 21–26, 2017.

135. B. Kim *et al.*, Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV), *arXiv*, 2018, arXiv:1711.11279v5.

136. A. Nguyen, M.R. Martínez, On quantitative aspects of model interpretability, *arXiv*, 2020, arXix:2007.07584v1.

137. I. Lage *et al.*, An evaluation of the human-interpretability of explanation, *arXiv*, 2019, arXiv:1902.00006v2.

138. P. Cortez, M.J. Embrechts, Opening black box data mining models using sensitivity analysis, [in:] *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, Paris, France, April 11–15, 2011.

139. M. Sundararajan, A. Taly, Q. Yan, Gradients of counterfactuals, *arXiv*, 2016, arXiv: 1611.02639v2.

140. R.C. Fong, A. Vedaldi, Interpretable explanation of black boxes by meaningful perturbation, [in:] *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, pp. 3449–3457, October 22–29, 2017.

141. R. Sun, Optimization for deep learning: Theory and algorithms, *arXiv*, 2019, arXiv: 1912.08957v1.

142. M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, *arXiv*, 2013, arXiv:1311.29013v3.

143. R. Fong, M. Parick, A. Vedaldi, Understanding deep networks via extremal perturbations and smooth masks, *arXiv*, 2019, arXiv:1910.08485v1.

144. V. Petsiuk, A. Das, K. Saenlo, RISE: Randomized input sampling for explanation of black-box models, *arXiv*, 2018, arXiv:1806.07421v3.

145. S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, Universal adversarial perturbations, *arXiv*, 2017, arXiv:1610.08401v3.

146. J. Li, W. Monroe, D. Jurafsky, Understanding neural networks through representation erasure, *arXiv*, 2017, arXiv:1612.08220v3.

147. D. Baehrens, D. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, K.-R. Müller, How to explain individual classification decisions, *Journal of Machine Learning Research*, **11**: 1803–1831, 2010.

148. A. Shrikumar, P. Greenside, A. Shcherbina, A. Kundaje, Not just a black box: Learning important features through propagation activation differences, *arXiv*, 2016, arXiv:1605.01713v2.

149. C. Szegedy *et al.*, Intriguing properties of neural networks, *arXiv*, 2014, arXiv:1312. 6199v4.

150. K. Dhamdhere, M. Sundararajan, Q. Yan, How important is a neuron?, *arXiv*, 2018, arXiv:1805.12233v1.

151. J.T. Springenberg, A. Dosovitsky, T. Brox, M. Riedmiller, Striving for simplicity: The all convolutional net, *arXiv*, 2015, arXiv:1412.6806v3.

152. K. Leino, S. Sen, A. Datta, M. Fredrikson, L. Li, Influence-directed networks explanations for deep convolutional, *arXiv*, 2018, arXiv:1802.03788v2.

153. A. Nguyen, J. Yosinski, J. Clune, Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks, *arXiv*, 2016, arXiv:1602.03616v2.

154. B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, *arXiv*, 2015, arXiv:1512.04150v1.

155. A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, *arXiv*, 2019, arXiv:1704.02685v2.

156. J. Zhang, Z. Lin, J. Brandt, X. Shen, S. Sclaroff, Top-down neural attention by excitation backprop, *arXiv*, 2016, arXiv:1608.00507v1.

157. J.K. Tsotsos, S.M. Culhane, W.Y.K. Wai, Y. Lai, N. Davis, F. Nuflo, Modeling visual attention via selective tuning, *Artificial Intelligence*, **78**: 507–545, 1994.

158. G. Liu, D. Gifford, Visualizing feature maps in deep neural networks using DeepResolve. A genomics case study, [in:] *International Conference on Machine Learning 2017 – Workshop on Visualization for Deep Learning (ICML)*, Sydney, Australia, pp. 32–41, 2017.

159. A. Chattopadhyay, A. Sarkar, P. Howlader, V. Balasubramanian, Grad-CAM++: Improved visual explanations for deep convolutional networks, *arXiv*, 2018, arXiv:1710. 11063v3.

160. S. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, [in:] *31st Conference of Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, December 4–9, 2017.

161. G. Montavon, S. Bach, A. Binder, W. Samek, K.-R. Müller, Explaining nonlinear classification decisions with deep Taylor decomposition, *arXiv*, 2015, arXiv:1512.02479v1.

162. L. Kirsch, J. Kunze, D. Barber, Modular networks: Learning to decompose computation, [in:] *32nd Conference on Neural Information Processing (NeurIPS)*, Montréal, Canada, December 2–8, 2018.

163. P. Manisha, C.V. Jawahar, S. Gujar, Learning optimal redistribution mechanisms through neural networks, *arXiv*, 2018, arXiv:1801.08808v1.

164. H. Tsukimoto, Extracting rules from trained neural networks, *IEEE Trans Neural Network*, **11**(2): 377–389, 2000.

165. B. Zhou, D. Bau, A. Oliva, A. Torralba, Interpreting deep visual, representations via network dissection, *arXiv*, 2018, arXiv:1711.05611v2.

166. H. Li, J.G. Ellis, L. Zhang, S.-F. Chang, PatternNet: Visual pattern mining with deep neural network, *arXiv*, 2018, arXiv:1703.06339v2.

167. B. Zhou, D. Bau, A. Oliva, A. Torralba, Comparing the interpretability of deep networks via network dissection, [in:] W. Samek *et al.* [Eds.], *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, LNCS, Springer, Cham, Vol. 11700, pp. 243–252, 2019.

168. W. Samek, G. Montavon, S. Lapushkin, C.J. Anders, K.-R. Müller, Towards interpretable machine learning: Transparent deep neural networks and beyond, *arXiv*, 2020, arXiv:2003.07631v1.

169. M. Graziani, V. Andrearczyk, H. Müller, Regression concept vectors for bidirectional explanations in histopathology, [in:] D. Stoyanow *et al.* [Eds.], *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*, Springer, pp. 124–132, 2018.

170. A. Ghorbani, J. Wexler, J. Zou, B. Kim, Towards automatic concept-based explanations, *arXiv*, 2019, arXiv:1902.03129v3.

171. D. Bau *et al.*, GAN dissection: Visualizing and understanding generative adversarial networks, *arXiv*, 2018, arXiv:1811.10597v2.

172. J. Kauffmann, M. Esders, G. Montavon, W. Samek, K.-R. Müller, From clustering explanations via neural networks, *arXiv*, 2019, arXiv:1906.07633v1.

173. G. Jeon, H. Jeon, J. Choi, An efficient explorative sampling considering the generative boundaries of deep generative neural networks, *arXiv*, 2019, arXiv:1912.05827v1.

174. M.D. Zeiler, G.W. Taylor, R. Fergus, Adaptive deconvolutional networks for mid and high level feature learning, [in:] *13th International Conference on Computer Vision (ICCV)*, Barcelon, Spain, November 6–13, 2011.

175. P.-J. Kindermans, K.T. Schütt, M. Alber, K.-R. Müller, D. Erhan, B. Kim, Learning how to explain neural networks: PatternNet and PatternAttribution, *arXiv*, 2017, arXiv:1705.05598v2.

176. B.N. Oreshkin, D. Carpov, N. Chapados, Y. Bengio, N-BETAS: Neural basis expansion analysis for interpretable time series forecasting, *arXiv*, 2020, arXiv:1905.10437v4.

177. U. Schlegel, D. Oelke, D.A. Keim, M. El-Assady, An empirical study of explainable AI techniques on deep learning models for time series tasks, *arXiv*, 2020, arXiv:2012.04344v1.

178. G. Bologna, Y. Hayashi, Characterization of symbolic rules embedded in deep DIMLP networks: A challenge to transparency of deep learning, *Journal of Artificial Intelligence and Soft Computing Research (JAISCR)*, **7**(4): 265–286, 2017, doi: 10.1515/jaiscr-2017-0019.

179. D.R. Kuhn, R.N. Kacker, Y. Lei, D. Simos, Combinatorial methods for explainable AI, [in:] *9th International Workshop on Combinatorial Testing (IWCT)*, Porto, Portugal, March 23–27, 2020.

180. W. Samek, K.-R. Müller, Towards explainable artificial intelligence, *arXiv*, 2019, arXiv:1919.12072v1.

181. M.T. Ribeiro, S. Singh, C. Guestrin, Model-agnostic interpretability of machine learning, [in:] *Proceedings of Human Interpretability in Machine Learning Workshop (WHI)*, New York, USA, 2016.

182. S. Liu *et al.*, Actionable attribution maps for scientific machine learning, *arXiv*, 2020, arXiv:2006.16533v1.

183. M.T. Ribeiro, S. Singh, C. Guestrin, Anchors: High-precision model-agnostic explanations, [in:] *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, New Orleans, USA, pp. 1527–1535, February 2–7, 2018.

184. L. Bottou *et al.*, Counterfactual reasoning and learning systems, *arXiv*, 2013, arXiv: 1209.2355.

185. K. Sokol, P. Flach, Counterfactual explanations of machine learning predictions: Opportunities and challenges for AI safety, [in:] *Proceedings of the AAAI Workshop on Artificial Intelligence Safety*, Vol. 2301, 2019.

186. P. Hall, On the art and science of explainable machine learning: Techniques, recommendations, and responsibilities, *arXiv*, 2020, arXiv:1810.02909v4.

187. E.R. Elenberg, A.G. Dimakis, M. Feldman, A. Karbasi, Streaming weak submodularity: Interpreting neural networks on the fly, *arXiv*, 2017, arXiv:1703.02647v3.

188. G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, *arXiv*, 2015, arXiv:1503.02531v1.

189. Q. Zhang, R. Cao, Y.N. Wu, S.-C. Zhu, Growing interpretable part graphs on ConvNets via multi-shot learning, *arXiv*, 2017, arXiv:1611.04246v2.

190. Q. Zhang, X. Wang, R. Cao, Y.N. Wu, F. Shi, S.-C. Zhu, Extracting an explanatory graph to interpret a CNN, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **43**(11): 3863–3877, 2020, doi: 10.1109/TPAMI.2020.2992207.

191. R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, F. Giannotti, Local rule-based explanations of black box decision systems, *arXiv*, 2018, arXiv:1805.10820.

192. J.R. Zilke, E.L. Mencía, F. Janssen, DeepRED – Rule extraction from deep neural networks, [in:] *Discovery Science 19th International Conference Proceedings (LNAI)*, Vol. 9956, pp. 457–473, 2016.

193. M.G. Augasta, T. Kathirvalavakumar, Reverse engineering the neural networks for rule extraction in classification problems, *Neural Processing Letters*, **35**(2): 131–150, 2012.

194. G. Su, D. Wei, K.R. Varshney, D.M. Malioutov, Interpretable two-level Boolean rule learning for classification, *arXiv*, 2016, arXiv:1511.07361v1.

195. W.J. Murdoch, A. Szlam, Automatic rule extraction from long short term memory networks, [in:] *International Conference on Learning Representations*, Toulon, France, April 23–26, 2017.

196. Y. Ming, H. Qu, E. Bertini, RuleMatrix: Visualizing and understanding classifiers with rules, *arXiv*, 2018, arXiv:1807.06228v1.

197. M. Sato, H. Tsukimoto, Rule extraction from neural networks via decision tree induction, [in:] *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, Cat. No. 01CH37222, Vol. 3, pp. 1870–1875, 2001, doi: 10.1109/IJCNN.2001.938448.

198. N. Frosst, G. Hinton, Distilling a neural network into a soft decision tree, *arXiv*, 2017, arXiv:1711.09784v1.

199. Q. Cao, X. Liang, K. Wang, L. Lin, Linguistic driven graph capsule network for visual question reasoning, *arXiv*, 2020, arXiv:2003.10065v1.

200. O. Bastani, C. Kim, H. Bastani, Interpretability via model extraction, *arXiv*, 2018, arXiv: 1706.09773v4.

201. S. Tan, R. Caruana, G. Hooker, P. Koch, A. Gordo, Learning global explanations for neural nets model distillation, *arXiv*, 2018, arXiv:1801.08640v2.

202. H. Bride, J. Dong, J.S. Dong, Z. Hóu, Towards dependable and explainable machine learning using automated reasoning, [in:] *20th International Conference on Formal Engineering Methods (ICFEM)*, Gold Coast, QLD, Australia, 2018.

203. S. Krishnan, E. Wu, PALM: Machine learning explanations for iterative debugging, [in:] *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*, 2017, doi: 10.1145/3077257.3077271.

204. C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, C. Zhong, Interpretable machine learning: Fundamental principles and 10 grand challenges, *arXiv*, 2021, arXiv: 2103.11251v2.

205. J. Andreas, M. Rohrbach, T. Darell, D. Klein, Neural module networks, *arXiv*, 2017, arXiv:1511.02799v4.

206. R. Hu, M. Rohrbach, J. Andreas, T. Darrell, K. Saenko, Modeling relationship in referential expressions with compositional modular networks, *arXiv*, 2016, arXiv:1611.09978.

207. S. Sabour, N. Frost, G.E. Hinton, Dynamic routing between capsules, [in:] *31st Conference of Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, 2017.

208. H. Xue, W. Chu, Z. Zhao, D. Cai, A better way to attend: Attention with tress for video question answering, *arXiv*, 2019, arXiv:1909.02218v1.

209. A. Vaswani *et al.*, Attention is all you need, *arXiv*, 2017, arXiv:1706.03762v5.

210. X. Liu, K. Duh, L. Liu, J. Gao, Very deep transformers for neural machine translation, *arXiv*, 2020, arXiv:2008.07772v2.

211. H. Zhang, I. Goodfellow, D. Metaxas, A. Odena, Self-attention generative adversarial networks, *arXiv*, 2019, arXiv:1805.08318v2.

212. N. Mishra, M. Rohaninejad, X. Chen, P. Abbeel, A simple neural attentive meta-learner, *arXiv*, 2018, arXiv:1707.03141v3.

213. B. Hoover, H. Strobelt, S. Gehrmann, exBERT: A visual analysis tool to explore learned representations in transformers models, *arXiv*, 2019, arXiv:1910.05276.

214. R. He, W.S. Lee, H.T. Ng, D. Dahlmeier, Effective attention modeling for aspect-level sentiment classification, [in:] *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA, August 20–26, 2018.

215. J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, *arXiv*, 2019, arXiv:1810.04805v2.

216. G. Letarte, F. Paradis, P. Giguère, F. Laviolette, Importance of self-attention for sentiment analysis, [in:] *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 267–275, 2018.

217. E. Choi, M.T. Bahadori, J.A. Kulas, A. Schuetz, W.F. Stewart, J. Sun, RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism, *arXiv*, 2017, arXiv:1608.05745v4.

218. K. Xu *et al.*, Show, attend and tell: Neural image caption generation with visual attention, *arXiv*, 2016, arXiv:1502.03044v3.

219. O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: A neural image caption generator, *arXiv*, 2015, arXiv:1411.4555v2.

220. N. Xie, F. Lai, D. Doran, A. Kadav, Visual entailment: A novel task for fine-grained image understanding, *arXiv*, 2019, arXiv:1901.06706.

221. D.H. Park, L.A. Hendricks, Z. Akata, B. Schiele, T. Darrell, M. Rohrbach, Attentive explanations: Justifying decisions and pointing to the evidence, *arXiv*, 2016, arXiv:1612.04757.

222. D. Masharka, P. Tran, R. Soklaski, A. Majumdar, Transparency by design: Closing the gap between performance and interpretability in visual reasoning, *arXiv*, 2018, arXiv:1803.05268v2.

223. P. Anderson *et al.*, Bottom-up and top-down attention for image captioning and visual question answering, *arXiv*, 2018, arXiv:1707.07998v3.

224. D. Teney, P. Anderson, X. He, A. van der Hengel, Tips and tricks for visual question answering: Learnings from the 2017 challenge, *arXiv*, 2017, arXiv:1708.02711v1.

225. L.A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, T. Darrell, Generating visual explanations, *arXiv*, 2016, arXiv:1603.08507.

226. J. Kim, A. Rohrbach, T. Darrell, J. Canny, Z. Akata, Textual explanations for self-driving vehicles, *arXiv*, 2018, arXiv:1807.11546v1.

227. H. Liu, Q. Yin, W.Y. Wang, Towards explainable NLP: A generative explanation framework for text classification, *arXiv*, 2019, arXiv:1811.00196v2.

228. R. Zellers, Y. Bisk, A. Farhadi, Y. Choi, From recognition to cognition: Visual commonsense reasoning, *arXiv*, 2019, arXiv:1811.10830v2.

229. O. Li, H. Liu, C. Chen, C. Rudin, Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions, *arXiv*, 2017, arXiv:1710.048062v2.

230. C. Chen, O. Li, C. Tao, A.J. Barnett, J. Su, C. Rudin, This looks like that: Deep learning for interpretable image recognition, *arXiv*, 2019, arXiv:1806.10574v5.

231. P. Hase, C. Chen, O. Li, C. Rudin, Interpretable image recognition with hierarchical prototypes, *arXiv*, 2019, arXiv:1906.10651v1.

232. T. Lei, R. Barzilay, T. Jaakkola, Rationalizing neural predictions, *arXiv*, 2016, arXiv: 1606.04155.

233. D. Alvarez-Melis, T.S. Jaakkola, Towards robust interpretability with self-explaining neural networks, *arXiv*, 2018, arXiv:1806.07538v2.

234. Y. Dong, H. Su, J. Zhu, B. Zhang, Improving interpretability of deep neural networks with semantic information, *arXiv*, 2017, arXiv:1703.04096v2.

235. M. Alber *et al.*, iNNvestigate neural networks!, *arXiv*, 2018, arXiv:1808.04260v1.

236. H. Nori, S. Jenkins, P. Koch, R. Caruana, InterpretML: A unified framework for machine learning interpretability, *arXiv*, 2019, arXiv:1909.09223v1.

237. T. Spinner, U. Schlegel, H. Schäfer, M. El-Assady, explAIner: A visual analytics framework for interactive and explainable machine learning, *arXiv*, 2019, arXiv:1908.00087v2.

Received February 17, 2022; accepted May 5, 2022.