machine learning & knowledge extraction

MDPI

Systematic Review

XAIR: A Systematic Metareview of Explainable AI (XAI) Aligned to the Software Development Process

Tobias Clement [1,*,†], Nils Kemmerzell [1,†], Mohamed Abdelaal [2] and Michael Amberg [1]

[1] School of Business and Economics, Friedrich-Alexander University Erlangen-Nuremberg, 90403 Nuremberg, Germany
[2] Software AG, 64297 Darmstadt, Germany
* Correspondence: tobias.clement@fau.de
† These authors contributed equally to this work.

Abstract: Currently, explainability represents a major barrier that Artificial Intelligence (AI) is facing in regard to its practical implementation in various application domains. To combat the lack of understanding of AI-based systems, Explainable AI (XAI) aims to make black-box AI models more transparent and comprehensible for humans. Fortunately, plenty of XAI methods have been introduced to tackle the explainability problem from different perspectives. However, due to the vast search space, it is challenging for ML practitioners and data scientists to start with the development of XAI software and to optimally select the most suitable XAI methods. To tackle this challenge, we introduce XAIR, a novel systematic metareview of the most promising XAI methods and tools. XAIR differentiates itself from existing reviews by aligning its results to the five steps of the software development process, including requirement analysis, design, implementation, evaluation, and deployment. Through this mapping, we aim to create a better understanding of the individual steps of developing XAI software and to foster the creation of real-world AI applications that incorporate explainability. Finally, we conclude with highlighting new directions for future research.

Keywords: explainable AI; software development process; systematic review; feature importance

Citation: Clement, T.; Kemmerzell, N.; Abdelaal, M.; Amberg, M. XAIR: A Systematic Metareview of Explainable AI (XAI) Aligned to the Software Development Process. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 78–108. https://doi.org/10.3390/make5010006

Academic Editor: Luca Longo

Received: 7 December 2022
Revised: 5 January 2023
Accepted: 6 January 2023
Published: 11 January 2023

1. Introduction

In the recent decades, artificial intelligence (AI) models have become a key technology that enabled breakthrough innovations in various application domains, e.g., natural language processing [1], computer vision [2–4], autonomous driving [5], agriculture [6], and healthcare [7]. However, these innovations have been realized at the cost of poor model interpretability, which makes it challenging for humans to trace their decision process. Figure 1 illustrates a classification of most popular AI models according to their complexity, potential performance in AI applications, and level of interpretability. As the figure depicts, traditional machine learning algorithms, tend to be more readily explainable, while being relatively less powerful in terms of predictive performance. Other advanced algorithms, such as deep learning models, remain much harder to explain while being more powerful in complex systems.

This classification leads to a natural question of why explainability is a crucial measure in AI applications. Generally speaking, providing explanations of AI models and their predictions can be necessary for commercial benefits, ethical concerns, or for regulatory considerations. For instance, European Union regulation 679 [8] guarantees data owners the right to receive explanations of the decision reached using their data and to challenge the decision if it was generated by AI models. For decision-critical domains where automated decisions need to be well-understood, explainability is not only indispensable but also increases the acceptance of these AI-powered applications. This acceptance is broadly necessary in situations where AI models have a supporting role (e.g., medical diagnosis) as well as in situations where they practically make the decisions (e.g., autonomous driving).

Aside from justifying the predictions and decisions, explainability also helps in controlling the behavior of AI models by providing greater visibility over the inner components of AI models.

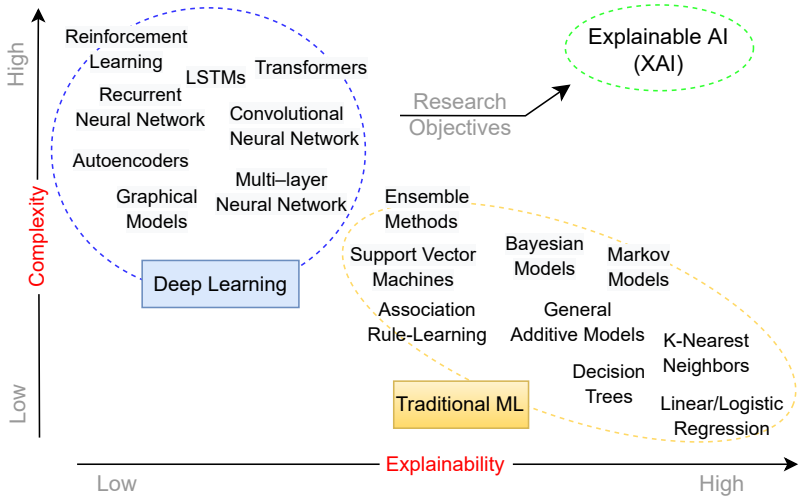


Figure 1. Classification of AI models according to their level of complexity, explainability, and their potential in modern AI applications.

Fortunately, plenty of methods and tools for explaining AI models have been introduced in the scientific literature [9–11]. As XAI has grown exponentially in recent years, it is challenging for individuals to get started with XAI and select appropriate XAI methods and tools for their applications. To keep track of new developments and trends in XAI research, there is an increasing number of reviews and overview articles on specific facets of XAI. A particularly popular trend is to create taxonomies of the proposed XAI methods [12–17], leading to several competing approaches to construct them [18]. However, the XAI landscape is simply too broad and complex to be compressed into a single pragmatically useful taxonomy. Due to this vast amount of taxonomies that lack uniformity between each other, practitioners may struggle to apply those explainability methods. To overcome this problem, few articles and reviews aim to provide assistance for practitioners. For instance, Saeed and Omlin [19] conducted a systematic metareview by adopting the machine learning lifecycle to structure the different challenges and opportunities of XAI. Moreover, Chazette et al. [20] introduce six core activities and related practices for XAI system development based on the results of a literature review and an interview study. Other recent reviews focus on a general overview of XAI [16,18,21–23] or individual aspects of XAI, such as regulatory frameworks [24], barriers and challenges [25], or explanation theory [17,26]. Moreover, some articles review explainability methods for specific application domains [27–33]. Such reviews mostly lack a clear description of all necessary aspects to develop XAI software. Accordingly, converting scientific innovations by academic research into usable real-world systems is still a major challenge, which hinders the applicability of XAI methods in industrial environments.

In this paper, we conduct a systematic metareview, called XAIR (XAI Review) and align our findings along the steps of the software development process [34] to create a better understanding of the individual steps of developing XAI software. Therefore, the research of this paper is structured around five research questions, which are illustrated in Figure 2. By these means, we aim to highlight the important steps during the software development process and to foster the creation of real-world AI applications that incorporate explainability. In response to our research questions, we provide the following three contributions: (1) We thoroughly analyze the literature on XAI to introduce a systematic metareview of the most prominent explainability methods and tools. (2) We align the results of our qualitative and quantitative analysis to the five steps of the software development process in order to

help ML practitioners during the development of XAI software. (3) We discuss the study implications and conclude the findings with future research directions for each step of the XAI software development process.

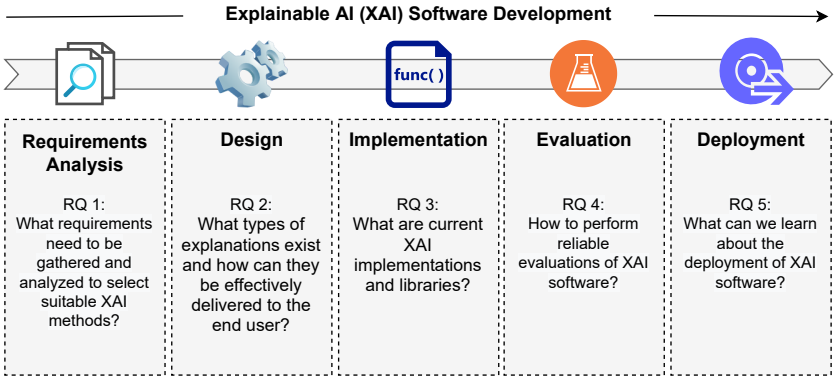


Figure 2. Addressed research questions (RQ) aligned to the XAI software development process.

The remainder of this paper is organized as follows. Section 2 provides an overview of the main terminologies and concepts in the field of XAI. In Section 3, we introduce our research methodology followed to analyze existing XAI literature with the goal of extracting useful insights on the development of XAI software. Sections 4–8 provide detailed answers to the aforementioned research questions which address the requirement analysis, XAI design, implementation, evaluation, and deployment. In Section 4, we list requirements that need to be collected and analyzed before developing or selecting a suitable XAI method. Section 5 elaborates on the implications related to the design phase of XAI software development. This phase includes the explanation design as well as the explainable user interface design. Section 6 highlights the most prominent XAI libraries, before Section 7 introduces different categorizations for the evaluation of XAI. In Section 8, we shed light on the various identified challenges of deploying XAI software. In Section 9, we first discuss different ways to select a suitable XAI method. Then, we present not only general research possibilities but also specific research directions toward the XAI software development process. Finally, Section 10 draws a conclusion of the obtained results.

2. Background

In this section, we discuss the main XAI concepts and terminologies, which are helpful to fully understand the technical contributions in this article. First, we define the term explainability before describing the individual components of an XAI. We then discuss the different types of explanations as well as the existing categorizations of XAI methods.

2.1. Important Concepts

In 2004, the term *XAI* was first coined by Van Lent et al. [35] when describing their system's ability to explain the behavior of AI-controlled entities in simulation games. The progress toward explainable AI models slowed down, however, as AI reached a tipping point with the spectacular advances in ML. In the wake of these advances, the focus of AI research has shifted to improving the predictive power of these models, while the ability to explain the underlying decision processes has taken a back seat [9]. In general, there is no agreed definition of XAI in the scientific literature. A widely accepted definition is introduced by Adadi and Berrada [9], where XAI is defined as the movement, initiatives, and efforts being made in response to concerns about transparency and trust in AI systems. These concerns are mainly connected to the decisions, generated using AI models, which ultimately affect people's lives (such as in healthcare or law) since there is an increasing need to understand how such decisions are made by AI models [36]. Therefore, the goal of enabling explainability "is to ensure that algorithmic decisions as well as any data driving those decisions can be explained to end-users and other stakeholders in nontechnical

terms" [37]. Across different research communities, different terms are used around the key concepts shaping this explainability landscape.

Fundamentally, explainability is closely related to the concepts of *interpretability* and *fidelity*. To this end, interpretable systems are explainable if their operations can be understood by humans [9]. In other words, the interpretability of an explanation indicates how understandable an explanation is for humans. In the technical literature, explainability and interpretability are often used synonymously [10,11]. Accordingly, interpretability can be considered as a property related to an explanation, where explainability is a broader concept related to all actions to be explained. Aside from interpretability, the fidelity of an explanation captures how accurately an explanation describes the behavior of a model, i.e., how faithfully an explanation corresponds to the task model that generates predictions. In this sense, it is argued that an explanation should be understandable to humans and at the same time correctly describing the model behavior in the entire feature space. We follow this notion of explainability, as we believe fidelity is an essential term that should be taken into account in the development of XAI software.

2.2. XAI Components

As outlined in the introduction, this research explores the XAI software development process. Figure 3 illustrates the main components of an explainable AI and its possible stakeholders. As the figure depicts, the explainable AI typically consists of two components, namely a machine learning model and a XAI method. The model calculates predictions based on the training data, while the XAI method is responsible for generating explanations for the inner workings and predictions of the ML model. Accordingly, an explainable AI incorporates two outputs, predictions and explanations. In order to effectively deliver these outputs to the end user, typically a (graphical) user interface is implemented. Various stakeholders are able to engage with both the predictions and the explanations generated by the machine learning model. Data scientists and ML developers may utilize the explanations to gain a deeper understanding of the model's inner workings and optimize its performance. Domain experts may evaluate whether the model's behavior aligns with real-world logic. Managers and business owners may utilize the explanations to make informed decisions based on the model's outputs. To help with developing such an explainable AI, XAIR considers all relevant steps adopted from the building blocks for software development [34], starting from requirement analysis and finishing at the deployment of XAI applications.

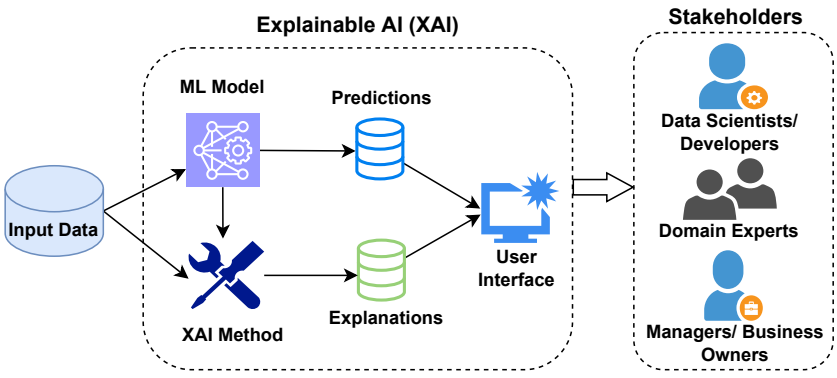


Figure 3. Explainable AI (XAI) and possible stakeholders.

2.3. Black- and White-Box Models

In this section, we highlight a selection of relevant XAI notions that we consider helpful to contextually define the field under study. One important classification of ML models is according to their inherent model interpretability. Hereby, we differentiate between *white-box* and *black-box* models. In general, white-box models typically create a description, interpretable for humans, of the input–output relationship. For example, the coefficients in a linear regression report how an additional input unit changes the output in a linear

relation. Other examples of white-box models comprise other types of traditional machine learning models, e.g., decision trees, rule-based learners and k-nearest-neighbor models. On the contrary, black-box models are de facto not humanly comprehensible due to the high non-linearity and the model size. These properties enable such models to capture complex relationships within the data, which the white-box models are not able to learn. Therefore, interpretability usually comes at the cost of model accuracy. Examples of black-box models are neural networks, ensemble methods and support vector machines. The XAI methods which aim to explain the model predictions after the training and inference processes are typically referred to as *post-hoc* methods in the literature [9,38].

2.4. Global and Local Interpretability

The main objective of *global interpretability* is to create general comprehensibility of a model's behavior [9,39–43]. The derived explanations are valid for all data instances. These types of XAI methods are useful if the goal is to uncover the general mechanisms behind the model features and the outputs. In contrast, *local interpretability* refers to explanations for a single data point [9,39–43]. The aim of local XAI methods is to make individual decisions of a black-box model comprehensible. However, these approaches are not capable of finding a general relationship between the input features and the model outputs. Although, it can be argued that local interpretability is sufficient for some end-users because they might be less interested in a global explanation, but more interested in knowing what caused a certain model prediction in their individual case. In the next section, we elaborate on the research methodology adopted while reviewing the most prominent works in the realm of AI explainability.

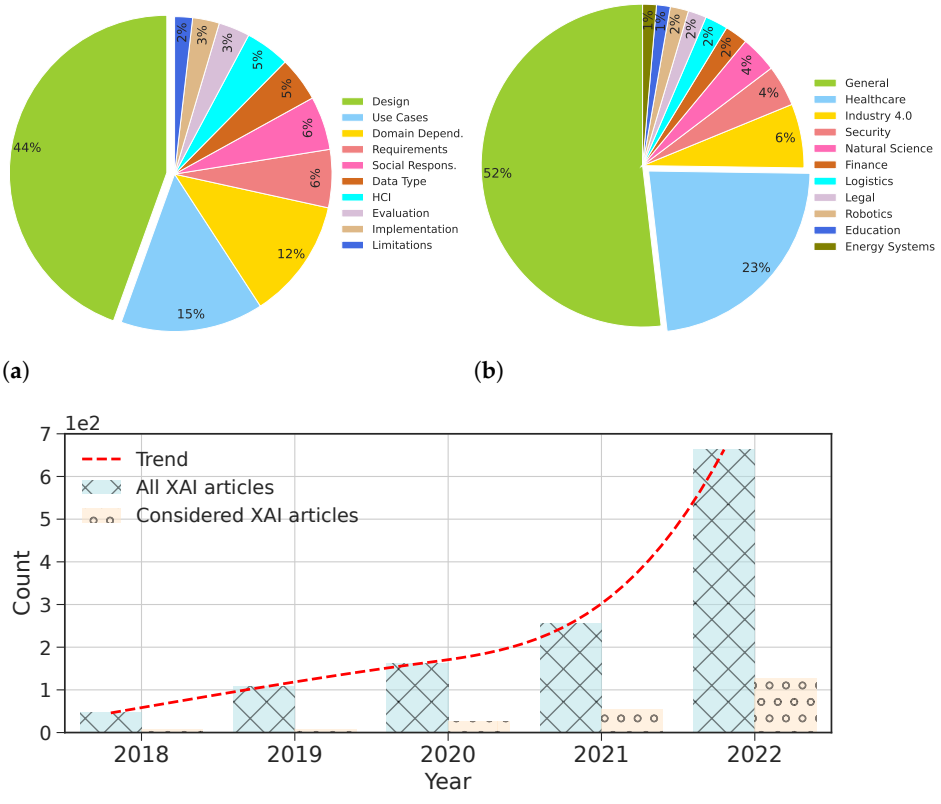
3. Research Methodology

Aiming to gain insights into the different steps of the XAI software development process, XAIR leverages the results of our systematic metareview of the most prominent XAI methods and tools. Accordingly, XAIR can broadly assist data scientists and software developers through leveraging the best practices of developing XAI software. To provide a comprehensive overview of the topic, XAIR analyses several scientific studies, not only from a variety of application domains (e.g., healthcare, Industry 4.0, security, finance, logistics), but also from different XAI research foci (e.g., XAI design, use cases, domain-dependent applications, requirements, social responsibility, HCI, implementation). In fact, organizing and classifying the XAI literature in a precise and indisputable manner is typically a challenging task. The difficulty of such a task emerges due to the multidisciplinary nature of this field of research, which ranges from Computer Science to Mathematics, from Psychology to Human Factors, from Philosophy to Ethics.

To overcome this challenge, XAIR leans on a set of guidelines, defined in [44]. Such guidelines involve four major steps, namely *planning*, *selection*, *extraction*, and *execution*. The planning step deals with clearly identifying the purpose of the intended review. Previous reviews usually lack the perspective on XAI-powered application development as they mainly focus on the broad range of XAI methods. In XAIR, we aim to fill this gap by deriving insights for the development of XAI software from the scientific literature. Accordingly, XAIR can help to translate innovations created by researchers into real-world applications. The selection step involves searching the literature for XAI articles and carrying out an initial screening. The adopted search strings are derived from various XAI concepts (cf. Section 2.1), including "explainable artificial intelligence", "transparent artificial intelligence", "responsible artificial intelligence" and "interpretable artificial intelligence" in combination with "review" or "survey". We opted to cover these four concepts within the literature review to achieve a broad perspective over the entire research field. The search process has been conducted on titles, abstracts, and keywords. As data sources, we selected several databases, such as Scopus, IEEE Xplore and ACM Digital Library, to cover different research communities. We considered the time frame between 2018 and 2022 to capture the most recent XAI literature. Searching within such a time frame resulted in a total of

1327 hits. In an initial screening process, we considered the abstracts, keywords and titles to exclude the articles that do not mainly cover the topic of XAI or do not exhibit a review or survey character. Furthermore, duplicates and non-English articles have been excluded. Such an initial screening resulted in 346 articles. Afterward, a thorough quality inspection of the remaining research articles has been conducted. Accordingly, the articles which did not meet the scientific rigour or were poorly written have been dismissed. As a result, we consider 227 articles in the remainder of this review.

In the extraction step, we carried out a *quantitative analysis* in which the characteristics of each article has been estimated. Figure 4 illustrates the quantitative results obtained in the extraction step. For instance, Figure 4a shows the distribution of different research foci within the analyzed XAI literature. As the figure depicts, 44% of the analyzed articles focus on the design of explainability methods. Such a category encompasses the articles with provide a general overview of available XAI methods and articles covering the explanation theory [11,45–48]. The second fraction, circa 15% of the examined articles, deals with the application of XAI methods in specific use cases. The domain dependent XAI methods, dedicated to specific domain such as natural science or finance, cover around 12% of the examined articles. Other research perspectives: the requirements analysis for XAI applications [49–54], the social responsibility [55–58], the processed data types [59–61], and human–computer interaction (HCI) [62–66] are represented by circa 6%, 6%, 5%, and 5% of the examined articles, respectively.



(**a**) (**b**)

(**c**)

Figure 4. Results of the quantitative analysis. (**a**) Research focus. (**b**) Application domains. (**c**) Time distribution.

As illustrated in Figure 4b, we also quantitatively analyzed the distribution of the XAI application fields. The results show that AI explainability plays a crucial role in several application domains, such as health care, Industry 4.0, and security. More than half of the analyzed reviews are not be linked to any application field. Rather, they take a general perspective on XAI. The most frequently occurring application field is health care (i.e.,

circa 23% of the examined articles). This result is not surprising because explanations in this field are essential to enhance the trust in the AI-powered health care systems and to reason the consequent actions based on the model's predictions. After health care comes Industry 4.0 and security in the third (circa 6%) and the fourth (circa 4%) places, respectively. The Industry 4.0 class covers the articles which adopt XAI methods in industrial settings, while the security class contains the articles which adopt XAI for fraud detection. In natural sciences (circa 4%), XAI has been used to generate and explain new scientific findings, for example in chemistry [67] and high-energy physics [68]. The articles with a legal background (circa 2%) cover algorithmic transparency in the European General Data Protection Regulation (GDPR) [24], the applications of XAI on legal text [69], the explanation techniques in law and their applications for machine learning models [70]. In the field of robotics (circa 2%), the considered articles address the subject of explainable reinforcement learning [71,72] as well as categorization of explanatory capabilities and requirements [51]. Examples of other application fields include autonomous driving [50], communication systems and networking [49,73], education [74,75], and social sciences [11].

As illustrated in Figure 4c, the number of XAI articles is broadly increasing since 2018, confirming the rising interest in the topic of XAI. At the beginning of the observed period, all the analyzed reviews did not address a specific field of application. Since then, the number of publications for specific applications has been increasing, demonstrating the usefulness of XAI in different fields. XAI methods have always been the most important focus of the articles. However, in recent years, the trend has shifted from generally-written articles toward articles targeting specific methods or application domains. This shift indicates that the need for generic reviews has become saturated, whereas their applications are still highly relevant. Hence, it is not only important how to generate explanations from a technical perspective, but also how theory from psychology and linguistics can help to improve explanation quality and how HCI concepts can make them more human-friendly.

This quantitative analysis of the literature shows that XAIR covers a great variety of the academic field. In addition, we used forward and backwards search to further deepen this foundation in order to retrieve all the necessary information [76]. Finally, in the execution step, we thoroughly analyzed the literature with regards to our research questions and aligned our findings along the software development process. The process of software development has been a subject of research for a long time [77,78]. Its basic steps usually include requirement analysis, design, implementation, evaluation, and deployment [34]. Along these lines, XAIR leans on the processes which are well-established in the context of AI and software development in order to identify the most important aspects for the development of XAI software. Since the "explainable" component places additional requirements on the development, there is hardly any development process for this type of XAI software so far [79]. Hereby, we focus on how the basic steps of the software development process need to be adapted with respect to this additional component. In the next sections, we align our results of the proposed literature meta-review along the aforementioned software development steps in order to provide answers to our research questions (cf. Figure 2).

4. Requirement Analysis

In the first step of the development process, the requirements of the XAI software need to be clearly specified. In addition to the functional requirements regarding the software itself, it is also necessary to formulate the requirements addressing the explainable-component. Hereby, we extracted several starting points from the literature. Initially, it is necessary to specify *what* needs to be explained and *to whom* [80]. In fact, a precise definition of the relevant stakeholders and the target users represents a major aspect for deriving the requirements [80–85]. To formulate such a definition, several user characteristics, e.g., AI knowledge, domain knowledge, attitude toward AI, responsibilities and cognitive abilities, can be exploited [20,80,83]. This definition is de facto important because different users require different kinds of explanations. For example, a machine learning engineer who has

a strong knowledge of the inner workings of ML models can make use of more complex explanations than a novice user without technical background. In fact, it is not only important to define what to explain and to whom but also *how to explain*. Hence, the requirements also deal with the type of explanations [82] or certain explanation characteristics [81]. For example, a manager would like to know the variables that need to be changed in order to obtain a different result, whereas a machine learning engineer would prefer to identify the most important variables for a better model understanding. This type of requirement is often linked to the type of user. The most frequently-used *data type* of an application can also act as a requirement when selecting the appropriate explainability method [83]. For instance, some explainability methods have been primarily developed for images [86] or tabular data [87], whereas others are not dependent on the data type [88].

Another set of requirements can be derived based on the *underlying ML model* [20]. Specific tasks may require certain model architectures, e.g., CNNs and RNNs. For example, in natural language processing, large transformer-based models [89] are the current state of the art. However, the size and computational cost of these networks may hinder the adoption of certain XAI methods. In this context, model-specific XAI methods could efficiently provide better explanations by considering the unique model characteristics. Along a similar line, a set of requirements can also be derived based on the *existence* of an AI system for which the explanations are needed. If an AI model is already in place, time and cost have already been invested. Thus, techniques which focus on post-hoc explanations are preferred, as they do not require any changes to the trained model. In contrast, if there is no trained model, choosing a white-box model design is more feasible. Aside from the ML models, the requirements can also be defined based on a set of *decision characteristics*, such as the outcome criticality, the time sensitivity and the decision complexity if the explanation is used to support decision making [83]. Another important aspect is the *context* of the explanation which can also be used to derive requirements, because it can imply constraints on the explanations [80]. For example, in the case of an automotive assistant system, visual explanations might be less feasible since they might distract the driver from the traffic.

In summary, there exist several requirements which need to be collected and thoroughly analyzed before designing the XAI software. In this regard, Chazette et al. [20] propose a trade-off analysis as part of the requirement analysis. Hereby, they argue that trade-off between explainability and other quality aspects like ease of use, user interface design or information load should already be part of the requirement analysis. When setting up the requirements, it is not only important to define the ideal system behavior, but also what happens if explainability is not possible and how much uncertainty is tolerable [82]. Following this approach usually reduces the unexpected consequences caused by unspecified system behavior.

5. Design Phase

In this section, we present our findings for the design phase of XAI software. The goal is to identify suitable methods for generating and presenting explanations for the model output. In comparison to the traditional design phase of a an AI system without an explainability component, XAI software needs two additional design steps, namely *explanation design* and *explainable user interface design* [79], which is illustrated in Figure 5.

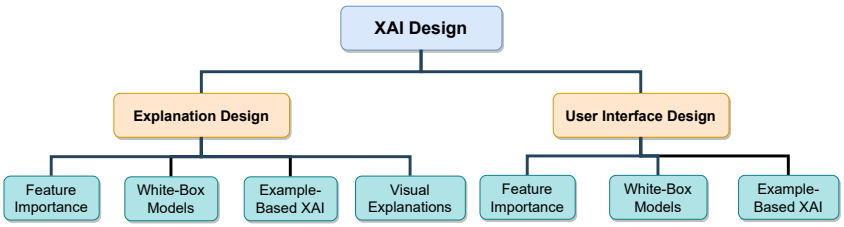


Figure 5. Classification of the reported XAI design methods.

To this end, there exist a variety of methods for both parts of the XAI design phase. On the one hand, the *explanation design* (Section 5.1) covers the process of selecting one or more suitable methods to generate appropriate explanations, depending on the requirements and the application which is a major step within the design phase. On the other hand, it is necessary to adequately *design the user interface* (Section 5.2) that precisely delivers the explanations to the application users. In the following, we use this classification to report on relevant XAI design methods.

5.1. Explanation Design

Here, the aim is not to provide a comprehensive overview on all existing XAI methods, but rather to propose a starting point for developers and practitioners to select suitable XAI methods. In this regard, XAIR differentiates itself from existing taxonomies, which merely focus on the categorization of XAI methods on a purely functional level [9,13,38]. Instead, XAIR introduces a simple categorization based on the type of explanation that developers and practitioners may want to implement to make their model more interpretable in combination with common types of ML models which they might want to use. As Figure 5 illustrates, we consider four major kinds of explanations for explanation design used in the literature, namely feature selection, white-box models, example-based XAI, and visual explanations. The most common explanation methods generate importance values for the input features. These features might be image pixels, word tokens, or numeric features from structured data. A second group of methods aims to create a white-box model that mimics the original black-box model and is inherently interpretable. Furthermore, example-based XAI methods use instances from the training data to make decisions of the black-box model more comprehensible. The last group relies on purely visual explanations. Below, we discuss the most prominent XAI methods in each of these groups. Table 1 summarizes the reported XAI methods with highlighting their scope and functionality.

5.1.1. Feature Importance

In the following, we discuss the most prominent feature importance methods, which calculate importance values for the input features. These features can be features can be in the form of image pixels, word tokens, or numeric features from structured data.

LIME. Several XAI methods that provide feature importance values can be employed with almost any type of black-box ML models. Among these XAI methods are the surrogate models, which can be used in lieu of the original models to improve explainability. In the context of local explainability, the surrogate models create a white-box model around a single data instance. Such surrogate models are inherently explainable, but are only valid for a single data instance. In particular, a surrogate model fits a data set sampled around a focal data point and is subsequently used to retrieve the importance vector for the input features. The first XAI method to exploit this approach is *LIME* [88]. As a model-agnostic XAI method (Model-agnostic methods can be used with any machine learning models, while model-specific methods aim to understand the black model machine learning models by analyzing their internal components and how they interact), LIME generates explanations through perturbing the input data samples to understand how the predictions change. Specifically, *LIME* creates a linear surrogate model which is trained on small perturbations of an original data instance. Accordingly, the output of LIME is simply a vector of explanations, reflecting the contribution of each feature to the prediction of a data instance. However, LIME suffers in some scenarios since (1) simple perturbations may not be sufficient to generate accurate predictions of the surrogate model and (2) it relies only on linear surrogate models which may not be powerful enough if the region, generated by perturbation around the sampled data instance, is relatively large.

Table 1. Overview of the reported XAI methods.

Explanation Type	Black-Box Model	Method	Scope	Functionality	Source
Feature Importance	Any	LIME	Local	Surrogate Model	[88]
		LORE	Local	Surrogate Model	[90]
		Anchors	Local	Surrogate Model	[91]
		Occlusion	Local	Input Perturbation	[86]
		Permutation Feature Importance	Global	Input Perturbation	[92]
		Shapley Feature Importance	Global	Game-Theory	[93]
		SHAP	Both	Game-Theory	[87]
	Neural Network	Guided Backpropagation	Local	Backpropagation	[94]
		Integrated Gradients	Local	Backpropagation	[95]
		Layerwise Relevance Propagation	Local	Backpropagation	[96]
		DeepLift	Local	Backpropagation	[97]
		Testing with Concept Activation Vectors	Global	Human Concepts	[98]
		Activation Maximization	Global	Forwardpropagation	[99]
	CNN	Deconvolution	Local	Backpropagation	[86]
		Class Activation Map	Local	Backpropagation	[100]
		Grad-CAM	Local	Backpropagation	[101]
	Transformer	Attention Flow/Attention Rollout	Local	Network Graph	[102]
		Transformer Relevance Propagation	Local	Backpropagation	[103]
White-Box Model	Any	Rule Extraction	Global	Simplification	[104]
		Tree Extraction	Global	Simplification	[105]
		Model Distillation	Global	Simplification	[106]
	CNN	Attention Network	Global	Model Adaption	[107]
	RNN	Attention Network	Global	Model Adaption	[108]
Example-Based	Any	Prototypes	Global	Example (Train Data)	[109]
		Critisisms	Global	Example (Train Data)	[110]
		Counterfactuals	Global	Fictional data point	[111]
Visual Explanations	Any	Partial Dependence Plot	Global	Marginalization	[112]
		Individual Conditional Expectation	Global	Marginalization	[113]
		Accumulated Local Effects	Global	Accumulation	[114]

Anchors & LORE. To overcome these limitations, the same authors introduce another model-agnostic XAI method based on IF-THEN rules, referred to as *Anchors* [91]. Instead of identifying the important features as *LIME* does, *Anchors* creates a region, i.e., numerical ranges, in the feature space. Such a region can be defined within a set of decision rules to precisely interpret the outputs of the black-box model. These rules can be interpreted as sufficient conditions for a particular prediction. For instance, Anchors can generate the following explanation: *Rose survived Titanic since she was a woman aged between 15 and 25, who had a cabin on the upper deck*. To find the decision rules, *Anchors* implements a greedy beam search algorithm. In particular, *Anchors* exploits the beam search algorithm to find a solution within a set of candidate rules which have been derived from perturbations of the input data instance. Due to the need for input data permutation, the search process becomes computationally intensive, especially if the number of features is relatively large. Another model-agnostic XAI method that is based on local surrogate models is *LORE* [90]. Similar to *Anchors*, *LORE* generates explanations in the form of decision rules. However, *LORE* generates the decision rules in a different way. Specifically, *LORE* creates synthetic neighborhood data samples, using a genetic algorithm, which then are used to fit a decision tree. Afterward, a set of decision rules is extracted from the decision tree classifier.

Permutation Feature Importance (PFI). Aside from surrogate models, permutation feature importance (PFI) is another powerful model-agnostic tool that aim to detect which features have predictive power. To this end, PFI methods tend to randomly shuffle the feature values. To measure the importance of an input feature, its data instances are shuffled, while leaving all other features constant. Afterward, the impact of such perturbation on the model performance is measured. In this context, a feature is considered important if the performance of the ML model drops significantly due to the perturbation process, which deliberately removes the dependency, learned by the model, between the input data and

the labels. For instance, Zeiler and Fergus [86] iteratively occlude different parts of the input images to find important regions. Although this concept was originally introduced for convolutional neural networks (CNN), it can be applied to any black-box model by defining an appropriate policy to occlude parts of the input data. Despite being simple, PFI methods suffer from several drawbacks. For instance, they are computationally intensive due to iterating through each of the input features. Moreover, they usually offer poor performance if the input features are correlated. In this case, the model may still perform well after shuffling one of these correlated features.

Shapley. Another group of model-agnostic XAI methods is based on the Shapley value [115], which is originally a concept from cooperative game theory. It has been applied to the field of XAI with the goal of attributing a model prediction to individual input features. Specifically, the Shapley value method treats each feature as a "player" in a game and the prediction as the "payout". In this setting, the Shapley value method finds how to fairly distribute the payout among the features. To this end, the Shapley method calculates the average marginal contribution of each feature across all possible coalitions (i.e., distinct combinations of this feature with other features). Due to the huge space of possible coalitions, several approximations, implemented in different ways, are necessary to efficiently execute the Shapley method. In this realm, *Shapley Feature Importance* (SFI) [93] is a global explanation method which relies on permutation to approximate the computations. The SFI method can be seen as an extension of the PFI method because instead of fully attributing importance gained through feature interaction to all features, it fairly distributes the importance scores across the features based on the Shapley values [93]. In contrast, *SHAP* (Shapley Additive Explanations) [87] generates local explanations. The approximations are made by adding mathematical assumptions which simplify the computations of the Shapley value. The authors provide both a model-agnostic version and model-specific versions for linear models and neural networks, which exploit the individual model characteristics.

Model-specific XAI. Because neural networks are one of the most commonly-used type of black-box models, there are plenty of XAI methods which focus on their explainability. They leverage the fact that the gradient of a neural network is already calculated during the training process. Thus, many approaches use the gradient to assert importance values to the input features. Due to the layered structure of neural networks, it is also possible to apply these XAI methods not only from input to output but also between different layers. This strategy enables a more detailed view on the hidden layers. This detailed view can be broadly valuable for AI engineers who aim to better understand the inner workings of their neural networks. However, the naïve solution of exploiting the gradient of the utility function with respect to a certain input has two major shortcomings. First, *saturation* [116] can cause the gradient to be almost zero. Thus, a change in the input will have no impact on the output, even when it might be important. Second, *discontinuities* in the gradient can cause sudden jumps in the importance scores over infinitesimal changes in the input [97]. Therefore, the gradient by itself is not feasible as a function of feature importance.

In order to overcome the problem of saturation within gradients, the *Integrated Gradients* [95] method interpolates between a baseline and the normal input. The baseline might be an all black image or a vector consisting of only zeros for tabular data. The gradients at each step of the interpolation are accumulated. This sum describes the importance of the input feature. Another approach to address the gradient saturation problem is *DeepLift* [97]. The basic idea behind *DeepLift* is to calculate the difference in the output from a reference output with respect to the difference in input from a reference input. This way, the gradient can be zero if saturation has been reached, but the difference from the reference is nonzero. The reference is chosen based on the problem at hand and can be compared to choosing the baseline for the *Integrated Gradients* method. The feature importance can be interpreted as the change in the output from the reference that is attributed by the change in the input feature from the reference point.

Backpropagation-based XAI. To understand what intermediate layers of convolutional neural networks (CNNs) are responding to, *Guided Backpropagation* [94] has been introduced. During the backward pass through a CNN network, all negative gradients are deliberately set to zero. Therefore, only positive weights are considered. This action leads to a better picture of the effect of the input on a particular output when compared to the case of including negative weights. Even though this method was originally introduced for CNNs, it can be applied to all other neural networks as long as they are differentiable. Another model-specific XAI method is *Layerwise Relevance Propagation* [96], where its idea stems from Kirchhoff's Law for electric circuits. It dictates that the sum of currents flowing into a node is equal to the sum of currents flowing out. Adapted to neural networks, this law means that the relevance score of a neuron must be equal to the sum of relevance scores of all connected neurons in the lower layer. Therefore, the relevance is redistributed through all layers from the output to the input of the network.

Forward Propagation-based XAI. Erhan et al. [99] propose an approach, referred to as *Activation Maximization*, based on the concept of forward propagation. The main objective of *Activation Maximization* is to find an input which produces the highest activation in the output layer. The higher the generated activation, the more important is the input feature. In contrast to the backpropagation-based methods, the *Activation Maximization* method results in global explanations. However, similar to the permutation-based methods, the *Activation Maximization* method is computationally inefficient compared to the backward-propagation-based methods because the forward propagation has to be calculated numerous times, whereas the gradient during the backward propagation is only calculated once.

Human Concepts. Aside from forward and backpropagation-based XAI methods, *Testing with Concept Activation Vectors (TCAV)* [98] is a global explanation method based on human understandable concepts. The idea behind TCAV is to test whether a neural network has learned a specific concept, for example, the concept of stripes in zebra images, by identifying what activations it causes within a specific layer of the neural network. Through TCAV, we can estimate the *relative importance* between a small set of concepts, rather than ranking the importance of all possible features/concepts. The main advantage of TCAV is that is requires no retraining or modifying the network. Specifically, users can express their concepts of interest using examples, i.e., a set of data instances exemplifying the concept. For instance, if the concept of interest is the gender, users can provide several images of women. In this case, a linear classifier can learn the differences, within the activations, between a data set containing the concept of interest and a random control group which does not. In addition, the authors show that visual concepts, like stripes or dots, appear more toward earlier network layers, whereas higher level concepts arise more at later layers. The disadvantage of this method is that additional data for each concept is needed to train classifiers for each individual concept. Hence, Ghorbani et al. [117] propose a method to automatically derive concepts for each class of a trained classifier based on a set of images of that particular class. Specifically, segments of different resolutions are automatically extracted and clustered into concepts representing textures, objects, and object parts. Subsequently, *TCAV* can be applied to the extracted concepts to investigate what concepts the classifier has learned.

CNNs. The above described approaches are applicable for any type of neural network architecture. However, some methods are focused on a specific type of architectures. A common type of neural networks typically used for image data are CNNs. The idea behind *Deconvolution* [118] is to inverse the layers of a CNN network. The authors propose network layers which link the hidden feature maps back to the input pixels and show which pixels caused an activation in the feature map. The larger the activation, the more important is the input feature. Another piece of work in this direction is *Class Activation Map (CAM)* [100] which shows regions within an input image which are important for the prediction of a specific output class. Hereby, the prediction score is mapped back from the output layer to the previous convolutional layer based on the linear weights. However, this approach is only viable for CNNs without any fully connected layers.

Therefore, Selvaraju et al. [101] propose *Grad-CAM* which is capable of creating CAM for a wider range of CNN model architectures. By multiplying the CAM with the output from *Guided Backpropagation*, it is possible to create both high-resolution and concept-specific explanations. This step is broadly necessary because the CAM method has the same resolution of the last convolutional feature map, e.g., for VGG [119], it is only 14×14 pixels.

Transformers & Attention Models. Recently, the transformer architecture [89] has been the foundation of models like BERT [120] or GPT-3 [1] which have led to state-of-the-art performances in the field of natural language processing. Subsequently, the architecture has also been applied to the field of computer vision [4]. At its heart lies the attention mechanism [121], which attributes pairwise importance scores between tokens within an input sequence. Although, attention weights are easily interpretable, their usefulness for explainability is a debated issue. On the one hand, Vashishth [122] argues that attention scores can be used to explain the model behavior. On the other hand, Pruthi et al. [123] show that they can be manipulated while still being relied on. This could be exploited to deceive end-users with wrong explanations. Furthermore, attention weights are frequently uncorrelated to feature importance values generated by gradient-based methods [124]. As a result, Abnar and Zuidema [102] propose a method which goes beyond individual attention scores. It computes importance values for input tokens at a specific layer by utilizing both the raw attention scores of the focal layer and those from previous layers. The calculation is based on a directed graph which resembles the structure of the transformer. Hidden embeddings and inputs are modeled as nodes, whereas the attention values weight the edges in-between. The authors provide two different ways of propagating the importance scores through the graph to the inputs, namely *Attention rollout* and *Attention Flow*. Instead of using a graph which linearly combines the attention weights, Chefer et al. [103] apply the gradient to propagate attention scores through the network using the principles of *Layerwise Relevance Propagation* and specifically adapting to the transformer architecture.

5.1.2. White-Box Models

Instead of interpreting the model predictions, another group of methods aims to convert the original black-box model into an inherently explainable white-box model. For instance, Craven et al. [105] propose a method to extract decision trees from neural networks. Similarly, Zilke et al. [104] present how to extract decision rules from a deep neural network. Along a similar line, Liu et al. [106] apply the concept of knowledge distillation to transfer knowledge from a deep neural network to a decision tree. The neural network acts as a teacher, and the white-box model tries to mimic its decisions. This gain in interpretability comes at the cost of model accuracy, as the simpler model is not able to capture high-level dependencies in the input data. Instead of transferring knowledge to a simpler model, it is also possible to adjust the neural network architecture to make its predictions more comprehensible. For example, Zhang et al. [125] modify the top convolutional layer of a CNN to align the filters with different object parts existing in the input image. By adding an attention layers before the convolutional layers, Seo et al. [107] reveal which inputs are important for the CNN's prediction. Similarly, Choi et al. [108] apply the attention mechanism to the recurrent neural networks. In contrast to feature importance methods, changing the model's architecture requires a retraining of the network. Therefore, this approach can lead to comparatively high computational effort, and thus is more appropriate if there is no existing black-box model.

It is commonly believed that fuzzy rules are relatively easy to interpret [126]. However, some research suggests that the large number of fuzzy rules required to adequately describe a given context may make it difficult for humans to understand the model [127]. As a result, several approaches have been proposed to improve the interpretability of fuzzy systems. One strategy is to reduce the number of rules used, though this may negatively impact the model's performance. Alternatively, other methods involve using explanations in natural language to enhance the interpretability of the fuzzy rules [128–130]. In addition, there exist more advanced research works on nonmonotonic fuzzy reasoning and defeasible

reasoning for explainability, which are getting increasing importance in the field of XAI [131–133]. The traditional fuzzy logic and fuzzy rules are related to defeasible reasoning in that they both allow for uncertainty and imprecision in the reasoning process. Fuzzy logic is a form of many-valued logic that allows for degrees of truth rather than binary true or false values, and fuzzy rules are used to encode fuzzy logic systems. It is often used in situations where the available information is incomplete or imprecise, and it allows for the possibility of multiple conclusions being reached based on different interpretations of the information. Defeasible reasoning, however, is a type of logical reasoning that allows for the possibility of revising or overruling conclusions based on new evidence. It is also known as nonmonotonic fuzzy reasoning, and differs from fuzzy logic and fuzzy rules, which do not allow for such revision. This type of reasoning is often used in legal and decision-making contexts, where it is important to consider multiple pieces of evidence and be open to revising conclusions based on new information. This is in contrast to classical logic, in which conclusions are considered to be definitively true once they have been reached based on the available evidence. Because defeasible reasoning allows for the possibility of revising conclusions based on new information, it can be better suited to changing circumstances or situations where new information is constantly emerging. Another advantage of defeasible reasoning is that it can be more transparent and accountable than other types of reasoning. Because it is based on the consideration of multiple pieces of evidence and allows for the possibility of revising conclusions, it can be more easily understood and evaluated by others. One way to represent and understand the relationships between rules in a defeasible reasoning system is through the use of a graph, which can show the interactions and influences between different rules. This can make it easier to understand and trace the reasoning process and to identify any potential conflicts or inconsistencies in the system [131–133].

5.1.3. Example-Based XAI

The concept behind example-based methods is to create explanations based on the training data. The so-called *Prototypes* [109] are single data instances which are representative of all instances from a particular output class. They are identified by solving the optimization problem of finding the point which has the lowest distance to all other points in the data set [134–136]. The prototype can make a classification model more interpretable by demonstrating the differences between the representative data points for every class. In contrast, *Criticisms* [110] are examples where the machine learning model fails to fit the data. In other words, a criticism is a data instance that is not well represented by the set of prototypes. The authors argue that the combination with *Prototypes* can further enhance the model understanding by illustrating the limitations of the model. However, the main challenge of such methods is how to select the optimal configurations, such as the number of prototypes and criticisms.

Another example-based method is *Counterfactuals* [111] is based on the concept of contrastiveness, and it can be used to explain predictions of individual data instances [11]. Instead of explaining why a data point led to a certain decision of the black-box model, the counterfactual explanation provides the user with suggestions of how a decision, made by the machine learning model, can be altered via carrying out minimal changes to the input features. Such changes have to occur in a feature that can be feasibly changed by individuals [137]. For example, when applying for a bank credit, a counterfactual explanation displaying the necessary amount of additional income, that is needed to change the model's decision, is more useful than a counterfactual suggesting the alteration of the gender. As opposed to *Prototypes*, counterfactuals do not have to be actual instances from the training data, but can be a new combination of feature values.

It is also necessary to differentiate between counterfactuals and adversarial examples [138]. In general, the *adversarial examples* are input instances which lead to wrong model predictions due to limitations of the model [139]. They are usually created from existing instances for which the model can generate correct predictions by minimally changing the

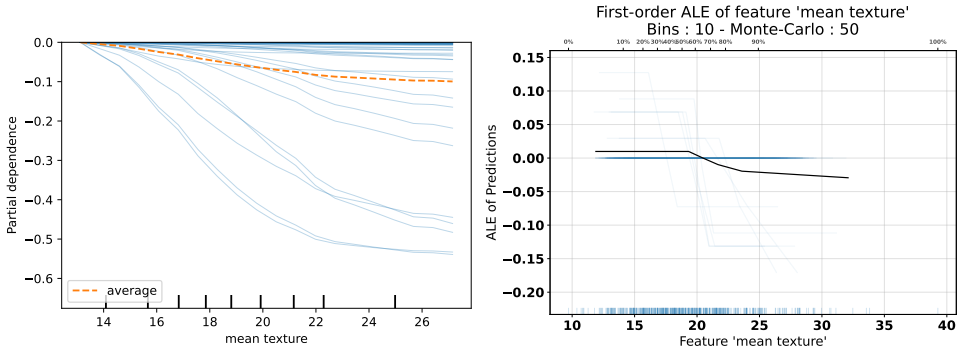
input until the model prediction changes. The major difference between counterfactuals and adversarial examples is that counterfactuals typically have different target values (usually the opposite), while adversarial examples have the same target values as the original instances. Finally, it is important to highlight that there exist both model-agnostic and model-specific counterfactual explanation methods.

5.1.4. Visual XAI

Some explanation techniques are based on purely visual concepts. Most prominently, *Partial Dependence Plot* (PDP) [112] is a global and model-agnostic XAI method which employs uses the partial dependence values to show the marginal effect of an input feature on the outcome of a machine learning model. Specifically, PDPs demonstrates whether the relationship between a feature and the prediction is linear, monotonic, or more complex. For instance, in binary classification tasks where machine learning models generate probabilities, PDPs shows the probability for a certain class given different values of a certain feature. In this case, the feature importance can be deducted from the shape of the curves in PDP plots. In particular, a flat PDP plot indicates that the feature is not important, and the more the PDP varies, the more important the feature is. Despite being simple and easy to interpret, PDP plots suffer in multiple scenarios, since they assume that the features are not correlated. Accordingly, PDPs becomes less suitable if there are interdependencies among the features. Moreover, PDPs usually fail to capture the heterogeneous effects which occur when a certain feature has different impacts on the prediction in different intervals, e.g., a positive association in one interval and a negative association in a subsequent interval. In this case, the PDP plot of such a feature may misleadingly show that the overall marginal effect is zero, since these two counteracting associations may cancel each other out.

An extension of the PDP plot is the *Individual Conditional Explanation* (ICE) plot [113], which also illustrates the relationship between an input feature and the target. To this end, it plots the average predicted outcomes for different values of a feature while holding the values of other features constant. In contrast to PDP, ICE shows the dependence for every sample of a certain feature, whereas PDP only visualizes the average contribution. This fine granularity level can be extremely helpful when heterogeneous effects within the features exist in the data set. Similar to PDP, ICE plots also suffer from the assumption of feature independence, which can result in misleading explanations. Moreover, it may become difficult to digest an ICE plot, if the number of ICE lines in the plot is extremely large. Similar to PDPs, the *Accumulated Local Effects* (ALE) plots [114] visualize the average effect of an input feature on the outcome of a machine learning model. However, ALEs differ from PDPs in computing the differences in predictions instead of showing the average values of the predictions. Specifically, ALE divides each feature into multiple small windows, where it estimates the prediction difference in each window. Afterward, it accumulates all the local windows to gain a full picture of the impact of that feature on the outcome of the machine learning model. The main advantage of ALE plots is that they are valid when the input features are correlated. Nevertheless, ALE plots typically fail to interpret an effect across windows. Moreover, there is a need to define an optimal number of windows for each feature. Figure 6 demonstrates an example of applying the three methods, i.e., PDP, ICE, and ALE, on the Breast Cancer data set (https://archive.ics.uci.edu/ml/datasets/breast+cancer, accessed on 1 December 2022). In this data set, several features, computed from a digitized image of a fine needle aspirate of a breast mass, are employed to differentiate between malignant and benign cancer. To this end, we train a multilayer perceptron (MLP) classifier with two hidden layers. For brevity, we selected only one feature, called the "mean texture", to show the difference among the three XAI methods. For instance, Figure 6a demonstrates the partial dependence values for different values of the mean texture. For the ICE plot, we limit to only 50 ICE curves to not overcrowd the plot. The dashed orange curve, which represents the output of the PDP method, clearly shows that the mean texture has a slight impact on the predictions of the MLP model. However, the ICE curves (light blue curves)

depicts a number of exceptions where increasing the mean texture has a negative influence on the output probabilities. Finally, Figure 6b demonstrates the first-order ALE plot of the mean texture. The figure shows different impacts within each interval, i.e., part of the light blue curves is above zero and another part is under zero. As a result, the average ALE curve, i.e., black curve, shows a flat line resulted from balancing the contradicting impacts in each interval.



(**a**) (**b**)

Figure 6. Examples of the visual-based XAI methods. (**a**) PDP & ICE curves. (**b**) ALE curves.

5.2. Explainable User Interface Design

After selecting a specific XAI method, the generated explanations need to be converted into a concrete presentation that the application's user can interact with. In the literature, this interaction is referred to as *explainable user interface* (XUI) [63,140]. Apart from general characteristics of human-friendly explanations like contrastiveness, truthfulness, and selectiveness [39], Chromik et al. [63] propose four design principles for XUI, including: (1) Combine texts and images to facilitate understanding and communicate effectively; (2) Offer hierarchical or iterative functionality that efficiently allows follow-ups on initial explanations; (3) Multiple explanation methods and modalities can help to triangulate insights; and (4) Adjust explanations to the user's mental model and context. Following these principles can elevate the XAI application from pure method outputs to a user-friendly application. Hence, we present our findings regarding the implementation of such principles as well as further methods to enhance the user-friendliness of explanations, generated from the XAI methods discussed in Section 5.1.

Feature Importance. Visualization is an important tool to create human-friendly explanations. Heatmaps can be generated from every XAI method that generates feature importance values for image data. The features can be individual pixels or a group of pixels [88]. SmoothGrad [141] can visually enhance these heatmaps by reducing noise. For text data, feature importance values can be visualized by highlighting individual tokens. For transformer-based architectures, an interactive visualization of the self-attention can be a useful tool [142]. For structured data, tornado diagrams can be applied to present the importance values in a visually appealing way. However, visualization is not the only way to deliver feature importance values to the end-user. For instance, in the realm of explainable recommender systems, the researchers propose to generate textual explanations from feature importance vectors. Specifically, the *template-based approaches* use presets of sentences that are employed following simple rules dependent on the feature importance values [143]. This allows a simple implementation and offers the XAI designer a complete control of the explanation that the end-user will receive. The disadvantage of this approach is that it might generate identical explanations when the decision rule between the templates is not fine-granular. This can lead to mistrust into the system, as unique explanations for unique data points might be expected by the end-user. Recently, Kim et al. [144] propose a method to generate textual explanations with the help of a neural network. This enables user-specific explanations by adapting the network's linguistic style individually and therefore adapting to the user's mental model (design principle 4). In contrast to template-based explanations, the designer has no way to control the output of the neural

network. In the worst case, this can lead to wrong explanations or explanations that might be linguistically offensive. Furthermore, training data is needed to train a generator network, which leads to additional labeling effort. To combine the advantages from both approaches, Li et al. [145] train a generator network whose explanations are restricted by a surrounding template.

White-Box Models. The fundamental attribute of white-box models is their intrinsic explainability. Regression models are typically used for structured data applications and produce linear weights for input features. Therefore, the presented approaches for feature importance methods can be applied for regression models as well. The basic structure of decision trees can be depicted efficiently as long as the tree size is moderate. More refined approaches [146] show the data distributions at each node, which makes the decision process more transparent and also enables model debugging. Decision rules are commonly presented as decision tables or rule sets in textual form. Ming et al. [147] propose an interactive framework to depict decision rules for a better understanding. Hereby, the authors use a combination of sankey diagrams and visualizations of data distributions. Additionally, decision rules can be converted into decision trees [148] which enables the application of the above-mentioned visualization techniques.

Example-based XAI. Example-based methods are inherently comprehensible for humans. However, prototypes and critisisms are likely to be insufficient on their own, as they do not provide a local explanation. Therefore, it is recommended to combine these methods with a local approach (design principle 3). Counterfactuals are easy to understand due to their contrastive nature [39]. However, finding a good counterfactual can be challenging because the counterfactual example should slightly differ from the original data point. Keane et al. [149] argue that counterfactuals from the original data set are more expressive than randomly perturbated examples. However, natural counterfactuals are sparse; therefore, they provide a method to artificially generate counterfactuals based on the characteristics of natural counterfactuals in the data set.

6. Implementation Phase

Following the design phase, the next step in the development process is to implement such a design. In fact, numerous open-source implementations of different XAI methods are available. In this section, we provide an overview of popular XAI libraries written in Python (cf. Table 2). This list makes no claim for completeness, but it is rather a starting point for the implementation of the most prominent XAI methods. Most libraries provide implementations of different types of explanations, although there is a general emphasis on feature importance methods. Popular XAI methods, e.g., *LIME* or *SHAP*, are often included in each package. Some libraries focus on a set of explainability methods or black-box model architectures. For instance, *Captum* provides feature importance methods for neural networks implemented in *Pytorch*. Another example is *DiCE*, which focuses on the generation of counterfactuals. In *PAIR Saliency*, various methods are implemented to generate feature importance maps for images. Similarly, *Quantus* is a package that focuses on the quantitative evaluation of explanations, primarily for the task of image classification. The authors provide more than 25 evaluation metrics that cover six different explanation characteristics, for example, robustness, faithfulness, or complexity. Even though deep neural models have become the state-of-the-art in many application fields, they are partially supported in XAI libraries. Additionally, most libraries are framework-dependent in the sense that they only focus on the implementation of XAI methods using either PyTorch or TensorFlow frameworks. For the explanation of traditional machine learning algorithms, most libraries rely on the model implementations from Scikit-learn [150].

Table 2. Overview of popular XAI libraries.

Name	Focus	Feature Importance	White-Box Models	Example-Based XAI	Visual XAI	Framework
AIX 360 [151]	General	LIME, SHAP	Decision Rules, Model Distillation	Prototypes, Contrastive Explanations	—	—
Alibi [152]	General	Anchors, Integrated Gradients, SHAP,	—	Contrastive Explanations, Counterfactuals	ALE	TensorFlow
Captum [153]	Neural Networks	DeepLift, Deconvolution, Integrated Gradients, SHAP, Guided Backpropagation, GradCam, Occlusion, PFI	—	—	—	PyTorch
DALEX [154]	General	LIME, SHAP, PFI	—	—	ALE, PDP	—
DiCE [155]	Counterfactuals	—	—	Counterfactuals	—	—
InterpretML [156]	General	LIME, SHAP, Morris Sensitivity Analysis	Explainable Boosting, Decision Tree, Decision Rules, Regression	—	PDP	—
PAIR Saliency [157]	Saliency Maps	Integrated Gradients, GradCam, Occlusion, Guided Backpropagation, Ranked Area Integrals, SmoothGrad	—	—	—	PyTorch, TensorFlow
Skater [158]	General	Layerwise Relevance Propagation, LIME, Integrated Gradients, Occlusion, PFI	Bayesian Rule List, Decision Tree	—	PDP	TensorFlow
Quantus [159]	Quantitative Evaluation	—	—	—	—	TensorFlow, PyTorch
ExplainerDashboard [160]	General	SHAP, PFI	Decision Tree	—	PDP	Scikit-learn
Ecco [161]	NLP	Integrated Gradients, Saliency, DeepLift, Guided Backprop	—	—	—	PyTorch
XAITK [162]	General	Saliency Maps	Decision Tree	Explanation by Example	—	—

7. Evaluation Phase

In this section, we present our results of the literature review regarding the evaluation of XAI methods. In general, the evaluation of XAI methods is a heavily-researched topic without a general consent toward a common solution. Here, the main challenge is that explanations are designed for humans, which makes the evaluation of such explanations partially subjective. The explanation recipients can greatly vary with respect to expectations, machine learning expertise, and domain knowledge. Therefore, the same explanation can be satisfying for one user but totally overwhelming or incomprehensible for another.

With respect to our literature analysis, we recognize a broad variety of different categorizations of evaluation levels and quality aspects [11,39,85,159,163–171]. However, to provide a workable overview of the relevant evaluation options, we follow a common way of classifying evaluation levels according to whether user involvement is required (human-based) or not (computational) [171]. As a guideline, the choice of the evaluation method might change depending on the advancement of the development process. At the beginning, computational evaluation can help to select the appropriate XAI methods without incurring high costs. With progress of the development process, the selected XAI methods can be examined for general applicability by human lay users. At the end of the development process, the XAI software has to be tested in its target setting, requiring expert knowledge for evaluation. Below, we discuss a variety of different quality aspects and evaluations metrics for both computational evaluation and human-based evaluation.

7.1. Computational Evaluation

This type of evaluation is performed automatically without human intervention and mainly relies on a set of metrics to evaluate XAI methods. In fact, such level of automation in the evaluation process, significantly reduces the costs in comparison to human-based evaluation. However, automatic quality evaluation is still challenging due to the inherent nature of explanations, being designed for humans. Moreover, ground truth explanations are often not existent [42]. In this context, the automatic metrics usually focus on individual properties of the explanations. Various explanation properties have been discussed in the literature [11,166] as well as their association to quantitative metrics [39].

Robustness. An important characteristic of an explanation is its *robustness* toward small changes in the input data. A good explanation is expected to be stable even when the input is slightly perturbed [159] because a user would expect similar input data to result in similar model behavior that can be explained in the same way. There are several implementations of such a concept, primarily for feature importance methods. The robustness of the explanation can be measured among others by similarity scores [117], sensitivity

analysis [172] and the variation between an explanation obtained from the original input and its perturbed version [173].

Faithfulness. The term *faithfulness* describes whether the detected importance of features is equivalent to their importance in the real world [42]. By removing important variables, the predictive accuracy should decrease. The faster it decreases, the more faithful the explanation method. Different strategies to implement this metric have been proposed [172,174–176]. Most of them rely on adding noise to the focal variable as a mean to remove the information it contains. The advantage of this approach is that the model does not need to be retrained, which would be required if the feature is entirely removed. Therefore, it truncates evaluation time, thanks to entirely avoiding the computationally-expensive retraining process.

Complexity. The *complexity* metric of an explanation is often determined by the amount of information that it contains, which is connected to its comprehensiveness [166]. Silva et al. [177] argue that explanations should be succinct. However, the optimal complexity might also differ between different types of end-users. Expert users might prefer more complex explanations than lay users. In the context of feature importance explanations, the term *complexity* relates to the number of important features on which the model relies on for its prediction. Hereby, different implementations exist in the literature. At their core, they rely on the importance distribution onto the entire set of features. For example, Chalasani et al. [178] use the Gini-index of the feature attribution vector as a measurement of its sparseness. The sparser the vector, the lower is the complexity of the explanation, since fewer features are required to explain the model behavior. Bhatt et al. [174] compute the entropy for the fractional attribution of each feature in relation to the total attribution. Nguyen et al. [175] count the number of features which exceed a specific attribution threshold.

Task-specific evaluation. Some evaluation metrics are only applicable for certain tasks. For example, in a multiclass classification, the *class sensitivity* method describes whether the explanation changes for different decisions [179]. A good explanation should be unique for each concept it explains. Therefore, the explanation should vary for different classes. The sensitivity can be measured by the similarity between the individual explanations for each class. In the case of image classification, the *localization* metric measures the ability of the XAI method to locate objects within an image, assuming that objects are usually important concepts for an explanation [179]. It can be calculated as the intersection of the concepts found by the XAI algorithm and objects found by an externally trained object detector [180]. If the goal of an XAI method is to increase the performance of the underlying AI model through its better understanding, the quality of the explanation can be measured as the difference in performance before and after making adequate changes based on the gained insights [85].

Method-specific evaluation. These quantitative evaluation metrics are oftentimes solely suited for a single type of explanations. In the case of feature importance methods, several metrics have been introduced and categorized by Hedström et al. [159]. For the evaluation of white-box models, which are extracted from an underlying black-box model, one metric is of particular interest, referred to as the *fidelity* metric. Specifically, the *fidelity* of a white-box model describes how well it matches the black-box model [41]. Hereby, common similarity measures can be applied. The greater the similarity between the two models, the more useful is the extracted model to explain the behavior of the original model. However, higher *fidelity* is usually provided by more complex models, which in return reduces their inherent explainability. Keane and Smyth [181] propose several evaluation metrics for counterfactual explanations. For instance, the *proximity* metric describes the similarity of the test instance to the generated counterfactual. It is assumed that a higher similarity is related to a better explanation. If the generated counterfactual is closer to the original data point, it is easier to comprehend the changes which would be necessary to change the model outcome. In order to quantitatively evaluate it, common distance metrics like the L1- or L2-norm can be used. Similarly, the *sparsity* metric describes the number

of changed features in the counterfactual example. The authors argue that a range of one to five changed features is reasonable. This measure is related to the *complexity* metric of feature importance explanations. The *relative distance* metric compares the mean distance of the generated counterfactuals to the mean distance of counterfactual instances, which naturally occur in the data set, on the assumption that a lower distance results in better explanations [181].

Sanity Checks. Adebayo et al. [182] provide two different approaches to sanity check an explanation method, namely the *model parameter randomization* test and the *data randomization* test. By measuring how strongly the explanation reacts to an increasing randomization of the model parameters or the training data, it is possible to evaluate the applicability of the XAI method on certain tasks. For example, if the explanations are independent of the data or the model, then they are not suitable to be used in tasks that depend on the model or the relationship between the inputs and the outputs.

7.2. Human-based Evaluation

Human-based evaluation within the XAI development process can be grouped into three types, including *goodness*, *user satisfaction*, and *mental model* [85]. The *goodness* of an explanation relies on a set of attributes that have been used by consensus to describe a useful explanation. Thus, it does not account for the individual situation of the recipient. Wanner et al. [165] suggest rating the goodness of an explanation by its intuitiveness, complexity, trustworthiness, understandability, and sufficiency. Hoffmann et al. [85] additionally examine—similar to Löfström et al. [169]—whether the explanation helps users to apply the AI model and to estimate its reliability. In contrast to the context-averse evaluation of goodness, *user satisfaction* is a contextualized measurement considering whether an explanation is adequate in the user's situation. It measures the "degree to which users feel that they understand the AI model or process being explained to them" [85]. For example, different users bring unique levels of background knowledge that require different kinds of explanations. A highly-detailed explanation that might be demanded by someone who has strong background in the AI field can be totally overwhelming for a layman. The user's *mental model* refers to the level of understanding of the underlying AI model [85]. A good explanation should strengthen the user's comprehension of the model behavior. This can be evaluated by letting the user predict which circumstances might lead to a good or bad model outputs given the explanations he has seen. This level of evaluations can usually be carried out by lay users, i.e., no domain experts are required. In this case, the evaluation can be performed through questionnaires, self-reports or interviews [41].

Depending on the maturity of the development process, the human-based evaluation of XAI software can also be conducted on an application basis [171]. This evaluation level involves conducting human experiments within a real application. In particular, the quality of an XAI method is evaluated in the context of its end-task. In other words, the application-grounded evaluation typically takes place during the final application of the XAI method. Thus, testers need domain knowledge to reliably evaluate the explanations within the application context. Requiring domain experts raises the evaluation cost, but it is still a strong indicator for the success of the XAI application. In this context, the evaluation criteria are use case specific, and they can be determined already within the requirements phase.

8. Deployment Phase

During the deployment, the AI software moves from the experimental development phase to the production phase. In this context, the deployment phase deals with enabling interoperability between ML models and other pieces of software, especially software that uses business logic. In fact, the deployment phase of XAI software has been rarely addressed within the scope of the analyzed literature. One possible reason is that application deployments are less common within research projects than in industrial settings. To explore how organizations deploy explainability methods, Bhatt et al. [174] conducted fifty

interviews with different stakeholders, e.g., data scientists, managers, and domain experts. They found that the majority of deployments are not for end users affected by the model but rather for data scientists, who use explainability to debug the model itself. Accordingly, there is a large gap between explainability in practice and the goal of transparency, since explanations primarily serve internal stakeholders rather than external ones. Moreover, the survey show that data scientists mostly need XAI methods to carry out model debugging (i.e., feature engineering to improve the model performance), model monitoring (i.e., early detection of data drift in the deployment phase), model transparency (i.e., communicate predictions to external stakeholders), and model audit (i.e., comply with regulations such as GDPR and AI Act).

In general, there exist several reasons for the limited deployment of XAI software and their use primarily as sanity checks for data scientists. First, several organizations prefer to rely on the judgments of domain experts rather than on explanations generated by the deployed models. Second, it becomes challenging to show explanations to the end users in real time due to technical limitations, e.g., the latency incurred by the computational complexity of the deployed XAI software or the difficulty of finding plausible counterfactual datapoints. Furthermore, providing certain explanations can raise privacy concerns owing to the risk of model inversion. Other reasons include the lack of *causal* explanations and the risk of spurious correlations which can be reflected in the generated model explanations. In [183], the authors conducted a study to understand the benefits of deploying XAI methods in cybersecurity operations. They faced several challenges in deploying the XAI software. For instance, there was a relatively low level of engagement from the intended end users, i.e., security analysts in this scenario, due to the "location" of the XAI tool, where it was embedded in an accordion menu in a supporting system. Moreover, the authors report about the excessive time needed to generate the explanations. Aside from the challenges of XAI deployment, the Python package *explainerdashboard* [160] enables a quick deployment of a Web-based interpretability dashboard that explains the workings of traditional machine learning models. Specifically, the interpretability dashboards are either exported to a static HTML file directly from a running dashboard, or programmatically, as an artifact, as a part of an automated CI/CD deployment process.

9. Discussion and Research Opportunities

By the means of a systematic literature review, we derived implications for the steps necessary to develop XAI software. However, due to the early maturity of XAI software, various research opportunities exist, and future work can address a variety of different topics to improve the development of future applications. In this section, we shed light on these research directions.

9.1. XAI Method Selection

As clearly observed in Section 5, in the literature exist plenty of XAI methods. These methods differ in their approaches and may even provide different explanations that complement each other in some scenarios. Accordingly, it is a challenge to select a well-suited XAI method for a particular use case. However, it makes sense to first raise a question of whether explanations of the model predictions are needed or simply documentations of the data, the model, or the system. Some organizations tend to make their ML systems transparent and accountable through documentation. Examples of such documentations are model cards [184], data cards [185], data statements [186], datasheets for datasets [187], data nutrition labels [188], system cards [189], and FactSheets [190]. All these documentation methods strive to organize information to make the ML systems more transparent to different stakeholders. Inspired by these documentation efforts, PAI [191] introduce an XAI Toolsheet, a one-page summary format for comparing between different XAI tools. For such comparisons, XAI Toolsheets adopt 22 dimensions falling under three major categories, namely metadata, utility, and usability. Aside from documentations, Belaid et al. [192] introduce Compare-xAI, a unified benchmark, with multiple use-cases, indexing

+16 post-hoc xAI algorithms, +22 tests, and +40 research paper. Through Compare-xAI, practitioners and data scientists can gain insights on which XAI methods are relevant to their problems.

In fact, documentations and benchmarks are effective tools for comparing and deciding upon the right XAI tools to adopt. Nevertheless, data scientists and practitioners should be aware of a set of hierarchically-structured selection criteria to systematically filter out irrelevant XAI tools. The main objective of such a set of criteria is to reduce the search space while selecting the right XAI tool. In addition to the criteria identified in the requirement analysis phase (cf. Section 4), data scientists and practitioners may also consider the scope of explanations, i.e., global or local. It is worthwhile mentioning that some XAI methods, such as SHAP, support both global and local explanations. In addition to the data type as a selection criterion, the data size also plays an important role in the selection process, especially for global XAI methods. As discussed in Section 5, several XAI methods, such as permutation-based methods, suffer from scalability issues due to their high computational costs. If the size of the data is large together with possessing limited computational resources, then it makes no sense to adopt such XAI methods.

Another criterion is related to how complex patterns in the data can be explained. Specifically, some XAI methods, e.g., SHAP, generate explanations, based on the individual contribution of a given feature, as well as the interactions between features. While other simple XAI methods, e.g., permutation methods, generate explanations for each feature separately, which in turn leads to a limited understanding of nonlinear patterns that the model has learned. Similarly, correlated features may have a significant impact on the quality of the model explanation. Therefore, it is necessary to identify correlated features in the data before generating explanations. Finally, the cardinality of the input features is another criterion for selecting an XAI method. In particular, the impurity-based feature importance for trees are strongly biased, where they typically favor high cardinality features (which have many possible distinct values, i.e., numerical features) over low cardinality features such as binary features or categorical variables with a small number of possible classes. Below, we provide a set of directions for future research to address the gaps and shortcomings identified throughout this study.

9.2. Future Directions

In this section, we express our thoughts on future research direction, formulated in general and development phase-dependent aspects. A further overview of current challenges and future research directions can be found in [19].

General. To date, not all phases of the development process are covered equally in the literature. For example, there has been a strong focus on the design process. In contrast, the deployment phase has not been a subject of research. Therefore, future work may aim to resolve this imbalance. Previous works have focused on individual aspects of the XAI development process. However, XAI software development has yet to be addressed in the literature from a point of view to cover all steps of the development process. Therefore, by applying this holistic view, future research might reveal potentials for synergies and conflicts within the development process. Currently, experiences derived from real-world implementations of XAI applications like from [193–195] are rare in the technical literature. This makes the deduction of design guidelines unfeasible. Therefore, applied research, for example, in the form of design science research [196], could help to close this gap.

Requirement Analysis Phase. Even though several starting points for generating requirements for XAI software have been proposed in the literature, a unified requirements catalog that addresses all major criteria is still missing. Hence, future research could develop such a tool to perform a comprehensive requirement analysis and thereby enhance the development of future XAI software.

Design Phase. Research towards XUI has been separated from XAI methods. However, from a practical perspective, it makes sense for XAI method designers to keep the principles of UI design in mind in order to assure that the generated explanations are

human-friendly. This can lead to better comprehensibility of XAI methods and higher quality of the XAI applications overall. Future research could further explore this intersection between XAI methods and interfaces. It has not been studied how different kinds of XAI techniques can be combined successfully to create a comprehensive explanation. Hence, future research could address this issue in user studies which analyze the fit between different types of explanations in order to increase the overall explanation quality.

Implementation Phase. So far, no implementation has been established as the go-to solution. Instead, there are a lot of redundant open-source packages. A general package that covers a great variety of explanations is yet not existent. This complicates the design of multiple different types of explanations. What further complicates the development of XAI software is that most packages only support one model backbone like *PyTorch*, textitTensorFlow, or *Scikit-learn*. Even though most developers may stick to one backbone, the usability of the packages is restricted in general. Therefore, future implementations could aim to unify existing approaches to support multiple explanation types and model architectures. This could enable a great variety of XAI software. Current open-source packages primarily focus on the implementation of XAI methods. In contrast, less address the evaluation of explanations and the user interface. Hence, future implementations could focus on these aspects to create a greater benefit for other developers and industry practitioners.

Evaluation Phase. Automated evaluation of explanation is inherently difficult; however, it has the potential to significantly decrease the human effort during the development process. The development of automated evaluation metrics has primarily focused on feature importance methods. In contrast, other types of explanations have been less explored. Additionally, researchers have criticized that user-studies were neglected during the development of these metrics [9,181]. It is therefore not conclusively clear whether the automatic metrics and the human perception of a good explanation match. This can be achieved by conducting user studies. Hence, the field of automated explanation evaluation provides numerous research opportunities. Furthermore, there is no evaluation methodology for XAI applications which has been commonly identified as a standard method. Hence, future work could address this issue by determining which evaluation approaches synergize well together and can create a holistic assessment of an XAI method or software. Hereby, all three evaluation levels proposed by Doshi-Velez and Kim [171] can be incorporated.

Deployment Phase. So far, researchers have not addressed the deployment of XAI software. However, this could be expected. Usually, the code which results from research projects is not deployed into real-world applications. Therefore, the researcher's development of XAI software usually stops after the evaluation process. In order to close this gap, future work could accompany the deployment of real-world XAI software and describe insights, challenges or best practices.

10. Conclusions

Explainability is an emerging interdisciplinary research field in the AI ecosystem. There are several research initiatives toward solving the ethical and trust-building issues surrounding the use of AI in its current form of real-world applications. Blind faith in the results of powerful predictive models is not advisable by today's standards due to the significant impact of data bias, trustworthiness, and adversarial examples in AI. In the spirit of holism, in this paper, we have first provided a comprehensive background on the topic of XAI. In the interest of mapping the broad landscape around XAI research, this paper has thoroughly reviewed a portfolio of explainability approaches by means of a systematic meta-review. In the center of this paper, we leveraged the findings of our literature analysis to derive insights for the development of XAI software and organized them along the the five phases of the software development process. The findings show that XAI has found its way into various application domains. However, we have seen evidence throughout this work that there is a lack of practical experience derived from real-world implementations of XAI software in the scientific literature. Moreover, we recognized that not all phases of the software development process are covered equally

by the literature. In essence, it has been noted that the current focus of XAI research is put on developing new theoretical explainability methods, whereas the application of those methods to real-world scenarios is sparse. It has then been concluded that considerable future effort will be required to tackle the practical challenges with the development of XAI software. In this paper, we aim to assist practitioners in incorporating XAI into real-world applications by compiling relevant information from the scientific literature for each step of the software development process. These resources may serve as starting points for practitioners seeking to incorporate XAI into their projects. Along these lines, we plan to extract further insights on the development of XAI software by applying the described XAI methods to real-world application scenarios.

Funding: This work was supported (in part) by the Federal Ministry of Education and Research through grants 02L19C155, 01IS21021A (ITEA project number 20219), and grant 01IS17045 (Software Campus project).

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
2. Smith, S.; Patwary, M.; Norick, B.; LeGresley, P.; Rajbhandari, S.; Casper, J.; Liu, Z.; Prabhumoye, S.; Zerveas, G.; Korthikanti, V.; et al. Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model. *arXiv* **2022**, arXiv:2201.11990.
3. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 12 June 2015.
4. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale. *arXiv* **2010**, arXiv:2010.11929 .
5. Muhammad, K.; Ullah, A.; Lloret, J.; Del Ser, J.; de Albuquerque, V.H.C. Deep Learning for Safe Autonomous Driving: Current Challenges and Future Directions. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 4316–4336. [CrossRef]
6. Fountas, S.; Espejo-Garcia, B.; Kasimati, A.; Mylonas, N.; Darra, N. The Future of Digital Agriculture: Technologies and Opportunities. *IT Prof.* **2020**, *22*, 24–28. [CrossRef]
7. Esteva, A.; Kuprel, B.; Novoa, R.A.; Ko, J.; Swetter, S.M.; Blau, H.M.; Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **2017**, *542*, 115–118. [CrossRef] [PubMed]
8. *Regulation (EU) 2016/679 of the European Parliament and of the Council*; Council of the European Union: Luxembourg, 2016.
9. Adadi, A.; Berrada, M. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access* **2018**, *6*, 52138–52160. [CrossRef]
10. Cabitza, F.; Campagner, A.; Ciucci, D. New frontiers in explainable AI: Understanding the GI to interpret the GO. In Proceedings of the International Cross-Domain Conference for Machine Learning and Knowledge Extraction, Dublin, Ireland, 25–28 August 2019; Springer: Berlin/Heidelberg, Germany, 2019, pp. 27–47.
11. Miller, T. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.* **2019**, *267*, 1–38. [CrossRef]
12. Angelov, P.P.; Soares, E.A.; Jiang, R.; Arnold, N.I.; Atkinson, P.M. Explainable artificial intelligence: An analytical review. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2021**, *11*, e1424. [CrossRef]
13. Belle, V.; Papantonis, I. Principles and practice of explainable machine learning. *Front. Big Data* **2021**, *4*, 39.. [CrossRef]
14. Langer, M.; Oster, D.; Speith, T.; Hermanns, H.; Kästner, L.; Schmidt, E.; Sesing, A.; Baum, K. What do we want from Explainable Artificial Intelligence (XAI)?–A stakeholder perspective on XAI and a conceptual model guiding interdisciplinary XAI research. *Artif. Intell.* **2021**, *296*, 103473. [CrossRef]
15. McDermid, J.A.; Jia, Y.; Porter, Z.; Habli, I. Artificial intelligence explainability: The technical and ethical dimensions. *Philos. Trans. R. Soc. A* **2021**, *379*, 20200363. [CrossRef]
16. Minh, D.; Wang, H.X.; Li, Y.F.; Nguyen, T.N. Explainable artificial intelligence: A comprehensive review. *Artif. Intell. Rev.* **2021**, *55*, 3503–3568. [CrossRef]
17. Vilone, G.; Longo, L. Classification of explainable artificial intelligence methods through their output formats. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 615–661. [CrossRef]
18. Speith, T. A review of taxonomies of explainable artificial intelligence (XAI) methods. In Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, 21–24 June 2022; pp. 2239–2250.

19. Saeed, W.; Omlin, C. Explainable AI (XAI): A Systematic Meta-Survey of Current Challenges and Future Opportunities. *arXiv* **2021**, arXiv:2111.06420.
20. Chazette, L.; Klünder, J.; Balci, M.; Schneider, K. How Can We Develop Explainable Systems? Insights from a Literature Review and an Interview Study. In Proceedings of the International Conference on Software and System Processes and International Conference on Global Software Engineering, Pittsburgh, PA, USA, 20–22 May 2022; pp. 1–12.
21. Mueller, S.T.; Hoffman, R.R.; Clancey, W.; Emrey, A.; Klein, G. Explanation in Human-AI Systems: A Literature Meta-Review, Synopsis of Key Ideas and Publications, and Bibliography for Explainable AI. *arXiv* **2019**, arXiv:1902.01876.
22. Li, X.; Xiong, H.; Li, X.; Wu, X.; Zhang, X.; Liu, J.; Bian, J.; Dou, D. Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond. *arXiv* **2022**, arXiv:2103.10689.
23. Dwivedi, R.; Dave, D.; Naik, H.; Singhal, S.; Rana, O.; Patel, P.; Qian, B.; Wen, Z.; Shah, T.; Morgan, G.; et al. Explainable AI (XAI): Core ideas, techniques, and solutions. *Acm Comput. Surv.* **2022**. [CrossRef]
24. Wulf, A.J.; Seizov, O. Artificial Intelligence and Transparency: A Blueprint for Improving the Regulation of AI Applications in the EU. *Eur. Bus. Law Rev.* **2020**, *31*, 4. [CrossRef]
25. Merhi, M.I. An Assessment of the Barriers Impacting Responsible Artificial Intelligence. *Inf. Syst. Front.* **2022**, 1–14. [CrossRef]
26. Srinivasan, R.; Chander, A. Explanation perspectives from the cognitive sciences—A survey. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, Yokohama, Japan, 7–15 January 2021; pp. 4812–4818.
27. Islam, M.R.; Ahmed, M.U.; Barua, S.; Begum, S. A systematic review of explainable artificial intelligence in terms of different application domains and tasks. *Appl. Sci.* **2022**, *12*, 1353. [CrossRef]
28. Degas, A.; Islam, M.R.; Hurter, C.; Barua, S.; Rahman, H.; Poudel, M.; Ruscio, D.; Ahmed, M.U.; Begum, S.; Rahman, M.A.; et al. A survey on artificial intelligence (AI) and explainable AI in air traffic management: Current trends and development with future research trajectory. *Appl. Sci.* **2022**, *12*, 1295. [CrossRef]
29. Ersöz, B.; Sağıroğlu, Ş.; Bülbül, H.İ. A Short Review on Explainable Artificial Intelligence in Renewable Energy and Resources. In Proceedings of the 2022 11th International Conference on Renewable Energy Research and Application (ICRERA), Istanbul, Turkey, 18–21 September 2022 ; IEEE: New York, NY, USA, 2022, pp. 247–252.
30. Başağaoğlu, H.; Chakraborty, D.; Lago, C.D.; Gutierrez, L.; Şahinli, M.A.; Giacomoni, M.; Furl, C.; Mirchi, A.; Moriasi, D.; Şengör, S.S. A Review on Interpretable and Explainable Artificial Intelligence in Hydroclimatic Applications. *Water* **2022**, *14*, 1230. [CrossRef]
31. Katarya, R.; Sharma, P.; Soni, N.; Rath, P. A Review of Interpretable Deep Learning for Neurological Disease Classification. In Proceedings of the 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 25–26 March 2022 ; IEEE: New York, NY, USA, 2022; Volume 1, pp. 900–906.
32. Fuhrman, J.D.; Gorre, N.; Hu, Q.; Li, H.; El Naqa, I.; Giger, M.L. A review of explainable and interpretable AI with applications in COVID-19 imaging. *Med Phys.* **2022**, *49*, 1–14. [CrossRef] [PubMed]
33. Anagnostou, M.; Karvounidou, O.; Katritzidaki, C.; Kechagia, C.; Melidou, K.; Mpeza, E.; Konstantinidis, I.; Kapantai, E.; Berberidis, C.; Magnisalis, I.; et al. Characteristics and challenges in the industries towards responsible AI: A systematic literature review. *Ethics Inf. Technol.* **2022**, *24*, 1–18. [CrossRef]
34. Royce, W.W. Managing the development of large software systems: Concepts and techniques. In Proceedings of the 9th International Conference on Software Engineering, Monterey, CA, USA, 30 March–2 April 1987; pp. 328–338.
35. Van Lent, M.; Fisher, W.; Mancuso, M. An explainable artificial intelligence system for small-unit tactical behavior. In Proceedings of the National Conference on Artificial Intelligence, Orlando, FL, USA, 18–22 July 1999; AAAI Press: Menlo Park, CA, USA; MIT Press: Cambridge, MA, USA, 2004; pp. 900–907.
36. Goodman, B.; Flaxman, S. European Union regulations on algorithmic decision-making and a "right to explanation". *AI Mag.* **2017**, *38*, 50–57. [CrossRef]
37. Barocas, S.; Friedler, S.; Hardt, M.; Kroll, J.; Venka-Tasubramanian, S.; Wallach, H. In Proceedings of the FAT-ML Workshop Series on Fairness, Accountability, and Transparency in Machine Learning, Stockholm, Sweden, 13–15 July 2018; p. 7.
38. Barredo Arrieta, A.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; Garcia, S.; Gil-Lopez, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. [CrossRef]
39. Carvalho, D.V.; Pereira, E.M.; Cardoso, J.S. Machine Learning Interpretability: A Survey on Methods and Metrics. *Electronics* **2019**, *8*, 832. [CrossRef]
40. Linardatos, P.; Papastefanopoulos, V.; Kotsiantis, S. Explainable AI: A Review of Machine Learning Interpretability Methods. *Entropy* **2020**, *23*, 18. [CrossRef]
41. Mohseni, S.; Zarei, N.; Ragan, E.D. A Multidisciplinary Survey and Framework for Design and Evaluation of Explainable AI Systems. *arXiv* **2018**, arXiv:1811.11839.
42. Samek, W.; Montavon, G.; Lapuschkin, S.; Anders, C.J.; Muller, K.R. Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications. *Proc. IEEE* **2021**, *109*, 247–278. [CrossRef]
43. Vilone, G.; Longo, L. Notions of explainability and evaluation approaches for explainable artificial intelligence. *Inf. Fusion* **2021**, *76*, 89–106. [CrossRef]
44. Okoli, C. A Guide to Conducting a Standalone Systematic Literature Review. *Commun. Assoc. Inf. Syst.* **2015**, *37*, *43* . [CrossRef]

45. Roscher, R.; Bohn, B.; Duarte, M.F.; Garcke, J. Explainable machine learning for scientific insights and discoveries. *IEEE Access* **2020**, *8*, 42200–42216. [CrossRef]
46. Chazette, L.; Brunotte, W.; Speith, T. Exploring explainability: A definition, a model, and a knowledge catalogue. In Proceedings of the 2021 IEEE 29th International Requirements Engineering Conference (RE), Notre Dame, IN, USA, 20–24 September 2021; pp. 197–208.
47. Vassiliades, A.; Bassiliades, N.; Patkos, T. Argumentation and explainable artificial intelligence: A survey. *Knowl. Eng. Rev.* **2021**, *36*, e5. [CrossRef]
48. Israelsen, B.W.; Ahmed, N.R. "Dave...I Can Assure You ...That It's Going to Be All Right ..." A Definition, Case for, and Survey of Algorithmic Assurances in Human-Autonomy Trust Relationships. *ACM Comput. Surv.* **2019**, *51*, 1–37 . [CrossRef]
49. Zhang, T.; Qiu, H.; Mellia, M.; Li, Y.; Li, H.; Xu, K. Interpreting AI for Networking: Where We Are and Where We Are Going. *IEEE Commun. Mag.* **2022**, *60*, 25–31. [CrossRef]
50. Omeiza, D.; Webb, H.; Jirotka, M.; Kunze, L. Explanations in autonomous driving: A survey. *arXiv* **2021**, arXiv:2103.05154.
51. Sheh, R. Explainable Artificial Intelligence Requirements for Safe, Intelligent Robots. In Proceedings of the 2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR), Tokoname, Japan, 4–6 March 2021; pp. 382–387.
52. Adams, J.; Hagras, H. A type-2 fuzzy logic approach to explainable AI for regulatory compliance, fair customer outcomes and market stability in the global financial sector. In Proceedings of the 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Glasgow, UK, 19–24 July 2020 ; pp. 1–8.
53. Gerlings, J.; Shollo, A.; Constantiou, I. Reviewing the Need for Explainable Artificial Intelligence (xAI). In Proceedings of the 54th Hawaii International Conference on System Sciences, Maui, HI, USA, 5 January 2021; p. 1284.
54. Sokol, K.; Flach, P. Explainability fact sheets. In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, 27–30 January 2020; Hildebrandt, M., Ed.; Association for Computing Machinery: New York, NY, USA; ACM Digital Library: New York, NY, USA, 2020; pp. 56–67. [CrossRef]
55. Zhang, T.; Qin, Y.; Li, Q. Trusted Artificial Intelligence: Technique Requirements and Best Practices. In Proceedings of the 2021 International Conference on Cyberworlds (CW), Caen, France, 28–30 September 2021; pp. 303–306.
56. Cheng, L.; Varshney, K.R.; Liu, H. Socially Responsible AI Algorithms: Issues, Purposes, and Challenges. *J. Artif. Intell. Res.* **2021**, *71*, 1137–1181. [CrossRef]
57. Trocin, C.; Mikalef, P.; Papamitsiou, Z.; Conboy, K. Responsible AI for Digital Health: A Synthesis and a Research Agenda. *Inf. Syst. Front.* **2021**, 1–19. . [CrossRef]
58. Ntoutsi, E.; Fafalios, P.; Gadiraju, U.; Iosifidis, V.; Nejdl, W.; Vidal, M.E.; Ruggieri, S.; Turini, F.; Papadopoulos, S.; Krasanakis, E.; et al. Bias in data–driven artificial intelligence systems—An introductory survey. *WIREs Data Min. Knowl. Discov.* **2020**, *10*, e1356. . [CrossRef]
59. Yepmo, V.; Smits, G.; Pivert, O. Anomaly explanation: A review. *Data Knowl. Eng.* **2022**, *137*, 101946. [CrossRef]
60. Sahakyan, M.; Aung, Z.; Rahwan, T. Explainable Artificial Intelligence for Tabular Data: A Survey. *IEEE Access* **2021**, *9*, 135392–135422. [CrossRef]
61. Zucco, C.; Liang, H.; Di Fatta, G.; Cannataro, M. Explainable Sentiment Analysis with Applications in Medicine. In Proceedings of the 2018 IEEE International Conference on Bioinformatics and Biomedicine, Madrid, Spain, 3–6 December 2018; Zheng, H., Ed.; IEEE: Piscataway, NJ, USA, 2018; pp. 1740–1747. [CrossRef]
62. Nazar, M.; Alam, M.M.; Yafi, E.; Mazliham, M.S. A Systematic Review of Human-Computer Interaction and Explainable Artificial Intelligence in Healthcare with Artificial Intelligence Techniques. *IEEE Access* **2021**, *9*, 153316–153348.. [CrossRef]
63. Chromik, M.; Butz, A. Human-XAI Interaction: A Review and Design Principles for Explanation User Interfaces. In *Human-Computer Interaction—INTERACT 2021*; Ardito, C., Lanzilotti, R., Malizia, A., Petrie, H., Piccinno, A., Desolda, G., Inkpen, K., Eds.; Information Systems and Applications, incl. Internet/Web, and HCI; Springer: Cham, Switzerland, 2021; pp. 619–640.
64. Dazeley, R.; Vamplew, P.; Foale, C.; Young, C.; Aryal, S.; Cruz, F. Levels of explainable artificial intelligence for human-aligned conversational explanations. *Artif. Intell.* **2021**, *299*, 103525. [CrossRef]
65. Naiseh, M.; Jiang, N.; Ma, J.; Ali, R. Explainable Recommendations in Intelligent Systems: Delivery Methods, Modalities and Risks. In *Research Challenges in Information Science*; Dalpiaz, F., Zdravkovic, J., Loucopoulos, P., Eds.; Lecture Notes in Business Information Processing; Springer: Cham, Switzerland, 2020; Volume 385, pp. 212–228. .{_}13. [CrossRef]
66. Wickramasinghe, C.S.; Marino, D.L.; Grandio, J.; Manic, M. Trustworthy AI Development Guidelines for Human System Interaction. In Proceedings of the 2020 13th International Conference on Human System Interaction (HSI), Tokyo, Japan, 6–8 June 2020; Muramatsu, S., Ed.; IEEE: Piscataway, NJ, USA, 2020; pp. 130–136. [CrossRef]
67. Dybowski, R. Interpretable machine learning as a tool for scientific discovery in chemistry. *New J. Chem.* **2020**, *44*, 20914–20920. [CrossRef]
68. Turvill, D.; Barnby, L.; Yuan, B.; Zahir, A. A Survey of Interpretability of Machine Learning in Accelerator-based High Energy Physics. In Proceedings of the 2020 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT), Leicester, UK, 7–10 December 2020; pp. 77–86.
69. Górski, Ł.; Ramakrishna, S. Explainable artificial intelligence, lawyer's perspective. In Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, Sao Paulo, Brazil, 21–25 June 2021 ; Maranhão, J., Wyner, A.Z., Eds.; ACM: New York, NY, USA, 2021; pp. 60–68. [CrossRef]

70. Atkinson, K.; Bench-Capon, T.; Bollegala, D. Explanation in AI and law: Past, present and future. *Artif. Intell.* **2020**, *289*, 103387. [CrossRef]
71. Anjomshoae, S.; Omeiza, D.; Jiang, L. Context-based image explanations for deep neural networks. *Image Vis. Comput.* **2021**, *116*, 104310. [CrossRef]
72. Puiutta, E.; Veith, E. Explainable reinforcement learning: A survey. In Proceedings of the International Cross-Domain Conference for Machine Learning and Knowledge Extraction, Dublin, Ireland, 25–28 August 2020; pp. 77–95.
73. Guo, W. Explainable artificial intelligence for 6G: Improving trust between human and machine. *IEEE Commun. Mag.* **2020**, *58*, 39–45. [CrossRef]
74. Alamri, R.; Alharbi, B. Explainable student performance prediction models: A systematic review. *IEEE Access* **2021**, *9*, 33132–33143. [CrossRef]
75. Fiok, K.; Farahani, F.V.; Karwowski, W.; Ahram, T. Explainable artificial intelligence for education and training. *J. Def. Model. Simulation Appl. Methodol. Technol.* **2021**, *19*, 154851292110286. [CrossRef]
76. Webster, J.; Watson, R.T. Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Q.* **2002**, *26*, xiii–xxiii.
77. Balsamo, S.; Di Marco, A.; Inverardi, P.; Simeoni, M. Model-based performance prediction in software development: A survey. *IEEE Trans. Softw. Eng.* **2004**, *30*, 295–310. [CrossRef]
78. Abrahamsson, P.; Salo, O.; Ronkainen, J.; Warsta, J. Agile Software Development Methods: Review and Analysis. *arXiv* **2017**, arXiv:1709.08439. https://doi.org/10.48550/ARXIV.1709.08439.
79. Liao, Q.V.; Pribić, M.; Han, J.; Miller, S.; Sow, D. Question-Driven Design Process for Explainable AI User Experiences. *arXiv* **2021**, arXiv:2104.03483.
80. Köhl, M.A.; Baum, K.; Langer, M.; Oster, D.; Speith, T.; Bohlender, D. Explainability as a non-functional requirement. In Proceedings of the 2019 IEEE 27th International Requirements Engineering Conference (RE), Jeju, Republic of Korea, 23–27 September 2019; pp. 363–368.
81. Hall, M.; Harborne, D.; Tomsett, R.; Galetic, V.; Quintana-Amate, S.; Nottle, A.; Preece, A. A systematic method to understand requirements for explainable AI (XAI) systems. In Proceedings of the IJCAI Workshop on eXplainable Artificial Intelligence (XAI 2019), Macau, China, 11 August 2019; Volume 11.
82. e-Habiba, U.; Bogner, J.; Wagner, S. Can Requirements Engineering Support Explainable Artificial Intelligence? Towards a User-Centric Approach for Explainability Requirements. *arXiv* **2022**, arXiv:2206.01507
83. Liao, Q.V.; Gruen, D.; Miller, S. Questioning the AI: Informing Design Practices for Explainable AI User Experiences. In *CHI'20*; Bernhaupt, R., Mueller, F.F., Verweij, D., Andres, J., McGrenere, J., Cockburn, A., Avellino, I., Goguey, A., Bjørn, P., Zhao, S., et al., Eds.; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1–15. [CrossRef]
84. Sheh, R.; Monteath, I. Defining Explainable AI for Requirements Analysis. *KI-Künstliche Intell.* **2018**, *32*, 261–266. [CrossRef]
85. Hoffman, R.R.; Mueller, S.T.; Klein, G.; Litman, J. Metrics for explainable AI: Challenges and prospects. *arXiv* **2018**, arXiv:1812.04608.
86. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. *arXiv* **2013**, arXiv:1311.2901.
87. Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 4765–4774.
88. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why Should I Trust You?". In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; Krishnapuram, B., Ed.; ACM Digital Library; ACM: New York, NY, USA, 2016; pp. 1135–1144. [CrossRef]
89. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, b.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
90. Guidotti, R.; Monreale, A.; Ruggieri, S.; Pedreschi, D.; Turini, F.; Giannotti, F. Local Rule-Based Explanations of Black Box Decision Systems. *arXiv* **2018**, arXiv:1805.10820.
91. Ribeiro, M.T.; Singh, S.; Guestrin, C. Anchors: High-Precision Model-Agnostic Explanations. *Proc. AAAI Conf. Artif. Intell.* **2018**, *32*, 1527–1535.
92. Fisher, A.; Rudin, C.; Dominici, F. All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *J. Mach. Learn. Res.* **2019**, *20*, 1–81.
93. Casalicchio, G.; Molnar, C.; Bischl, B. Visualizing the Feature Importance for Black Box Models. In *Machine Learning and Knowledge Discovery in Databases*; Berlingerio, M., Bonchi, F., Gärtner, T., Hurley, N., Ifrim, G., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2019; pp. 655–670.
94. Springenberg, J.T.; Dosovitskiy, A.; Brox, T.; Riedmiller, M. Striving for Simplicity: The All Convolutional Net. *arXiv* **2014**, arXiv:1412.6806.
95. Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic Attribution for Deep Networks. *arXiv* **2017**, arXiv:1703.01365.
96. Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K.R.; Samek, W. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLoS ONE* **2015**, *10*, e0130140. [CrossRef]
97. Shrikumar, A.; Greenside, P.; Kundaje, A. Learning Important Features through Propagating Activation Differences. *arXiv* **2017**, arXiv:1704.02685.

98. Kim, B.; Wattenberg, M.; Gilmer, J.; Cai, C.; Wexler, J.; Viegas, F.; Sayres, R. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). *arXiv* **2018**, arXiv:1711.11279.
99. Erhan, D.; Bengio, Y.; Courville, A.; Vincent, P. Visualizing higher-layer features of a deep network. *Univ. Montr.* **2009**, *1341*, 1.
100. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning deep features for discriminative localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26–30 June 2016; pp. 2921–2929.
101. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *arXiv* **2016**, arXiv:1610.02391. https://doi.org/10.1007/s11263-019-01228-7.
102. Abnar, S.; Zuidema, W. Quantifying Attention Flow in Transformers. *arXiv* **2020**, arXiv:2005.00928.
103. Chefer, H.; Gur, S.; Wolf, L. Transformer interpretability beyond attention visualization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021 ; pp. 782–791.
104. Zilke, J.R.; Loza Mencía, E.; Janssen, F. DeepRED—Rule Extraction from Deep Neural Networks. In *Discovery Science*; Calders, T., Ceci, M., Malerba, D., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland 2016; pp. 457–473.
105. Craven, M.; Shavlik, J. Extracting Tree-Structured Representations of Trained Networks. *Adv. Neural Inf. Process. Syst.* **1995**, *8*, *24–30* .
106. Liu, X.; Wang, X.; Matwin, S. Improving the Interpretability of Deep Neural Networks with Knowledge Distillation. In Proceedings of the 18th IEEE International Conference on Data Mining Workshops, Orleans, LA, USA, 18–21 November 2018; Tong, H., Ed.; IEEE: Piscataway, NJ, USA, 2018; pp. 905–912. [CrossRef]
107. Seo, S.; Huang, J.; Yang, H.; Liu, Y. Interpretable Convolutional Neural Networks with Dual Local and Global Attention for Review Rating Prediction. In Proceedings of the Eleventh ACM Conference on Recommender Systems, Como, Italy, 28 August 2017; Cremonesi, P., Ed.; ACM Digital Library; ACM: New York, NY, USA, 2017; pp. 297–305. [CrossRef]
108. Choi, E.; Bahadori, M.T.; Sun, J.; Kulas, J.; Schuetz, A.; Stewart, W. RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism. *arXiv* **2016**, arXiv:1608.05745.
109. Bien, J.; Tibshirani, R. Prototype selection for interpretable classification. *Ann. Appl. Stat.* **2011**, *5*, 2403–2424. [CrossRef]
110. Kim, B.; Khanna, R.; Koyejo, O.O. Examples are not enough, learn to criticize! Criticism for Interpretability. *Adv. Neural Inf. Process. Syst.* **2016**, *29, 2280–2288.*
111. Wachter, S.; Mittelstadt, B.; Russell, C. Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR. *Ssrn Electron. J.* **2017**, *31*, 2018. [CrossRef]
112. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
113. Goldstein, A.; Kapelner, A.; Bleich, J.; Pitkin, E. Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation. *J. Comput. Graph. Stat.* **2015**, *24*, 44–65. [CrossRef]
114. Apley, D.W.; Zhu, J. Visualizing the effects of predictor variables in black box supervised learning models. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2020**, *82*, 1059–1086. [CrossRef]
115. Shapley, L.S. A Value for n-Person Games. In *Contributions to the Theory of Games*; Kuhn, H.W.; Tucker, A.W., Eds.; Annals of Mathematics Studies; Princeton Univ. Press: Princeton, NJ, USA, 1953; pp. 307–318. [CrossRef]
116. Rakitianskaia, A.; Engelbrecht, A. Measuring saturation in neural networks. In Proceedings of the 2015 IEEE Symposium Series on Computational Intelligence, Cape Town, South Africa, 7–10 December 2015; pp. 1423–1430.
117. Ghorbani, A.; Wexler, J.; Zou, J.Y.; Kim, B. Towards Automatic Concept-based Explanations. *Adv. Neural Inf. Process. Syst.* **2019**, *32, 9273–9282* .
118. Zeiler, M.D.; Taylor, G.W.; Fergus, R. Adaptive deconvolutional networks for mid and high level feature learning. In Proceedings of the 2011 International Conference on Computer Vision (ICCV 2011), Barcelona, Spain, 6–13 November 2011 ; IEEE: Piscataway, NJ, USA, 2011; pp. 2018–2025. [CrossRef]
119. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556
120. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
121. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2018**, arXiv:1409.0473.
122. Vashishth, S.; Upadhyay, S.; Tomar, G.S.; Faruqui, M. Attention Interpretability Across NLP Tasks. *arXiv* **2019**, arXiv:1909.11218.
123. Pruthi, D.; Gupta, M.; Dhingra, B.; Neubig, G.; Lipton, Z.C. Learning to Deceive with Attention-Based Explanations. *arXiv* **2019**, arXiv:1909.07913.
124. Jain, S.; Wallace, B.C. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 3543–3556. [CrossRef]
125. Zhang, Q.; Wu, Y.N.; Zhu, S.C. Interpretable convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8827–8836.
126. Guillaume, S. Designing fuzzy inference systems from data: An interpretability-oriented review. *IEEE Trans. Fuzzy Syst.* **2001**, *9*, 426–443. [CrossRef]
127. Hüllermeier, E. Does machine learning need fuzzy logic? *Fuzzy Sets Syst.* **2015**, *281*, 292–299. . fss.2015.09.001. [CrossRef]
128. Mencar, C.; Alonso, J.M. Paving the Way to Explainable Artificial Intelligence with Fuzzy Modeling. In *Fuzzy Logic and Applications*; Fullér, R., Giove, S., Masulli, F., Eds.; Springer: Cham, Switzerland, 2019; pp. 215–227.

129. Bouchon-Meunier, B.; Laurent, A.; Lesot, M.J. XAI: A Natural Application Domain for Fuzzy Set Theory. In *Women in Computational Intelligence: Key Advances and Perspectives on Emerging Topics*; Smith, A.E., Ed.; Springer: Cham, Switzerland, 2022; pp. 23–49. .{_}2. [CrossRef]
130. Trivino, G.; Sugeno, M. Towards linguistic descriptions of phenomena. *Int. J. Approx. Reason.* **2013**, *54*, 22–34. . ijar.2012.07.004. [CrossRef]
131. Rizzo, L.; Longo, L. An empirical evaluation of the inferential capacity of defeasible argumentation, non-monotonic fuzzy reasoning and expert systems. *Expert Syst. Appl.* **2020**, *147*, 113220. [CrossRef]
132. Rizzo, L.; Longo, L. A qualitative investigation of the degree of explainability of defeasible argumentation and non-monotonic fuzzy reasoning. In Proceedings of the Proceedings of the 26th AIAI Irish conference on artificial intelligence and cognitive science, Dublin, Ireland, 6–7 December 2018; pp. 138–149.
133. Rizzo, L.; Longo, L. Inferential Models of Mental Workload with Defeasible Argumentation and Non-monotonic Fuzzy Reasoning: a Comparative Study. In Proceedings of the 2nd Workshop on Advances In Argumentation In Artificial Intelligence, Trento, Italy, 20–23 November 2018; pp. 11–26.
134. Ming, Y.; Xu, P.; Cheng, F.; Qu, H.; Ren, L. ProtoSteer: Steering Deep Sequence Model with Prototypes. *IEEE Trans. Vis. Comput. Graph.* **2020**, *26*, 238–248. [CrossRef]
135. Gurumoorthy, K.S.; Dhurandhar, A.; Cecchi, G.; Aggarwal, C. Efficient Data Representation by Selecting Prototypes with Importance Weights. In Proceedings of the 19th IEEE International Conference on Data Mining, Beijing, China, 8–11 November 2019 ; Wang, J., Shim, K., Wu, X., Eds.; IEEE: Piscataway, NJ, USA, 2019; pp. 260–269. [CrossRef]
136. Li, O.; Liu, H.; Chen, C.; Rudin, C. Deep Learning for Case-Based Reasoning Through Prototypes: A Neural Network That Explains Its Predictions. *Proc. AAAI Conf. Artif. Intell.* **2018**, *32, 3530–3537.* [CrossRef]
137. van Looveren, A.; Klaise, J. Interpretable Counterfactual Explanations Guided by Prototypes. In *Machine Learning and Knowledge Discovery in Databases. Research Track*; Oliver, N., Pérez-Cruz, F., Kramer, S., Read, J., Lozano, J.A., Eds.; Lecture Notes in Artificial Intelligence; Springer: Cham, Switzerland, 2021; pp. 650–665.
138. Kuhl, U.; Artelt, A.; Hammer, B. Keep Your Friends Close and Your Counterfactuals Closer: Improved Learning From Closest Rather Than Plausible Counterfactual Explanations in an Abstract Setting. *arXiv* **2022**, arXiv:2205.05515.
139. Madsen, A.; Reddy, S.; Chandar, S. Post-hoc Interpretability for Neural NLP: A Survey. *arXiv* **2021**, arXiv:2108.04840
140. Gunning, D.; Aha, D. DARPA's Explainable Artificial Intelligence (XAI) Program. *AI Mag.* **2019**, *40*, 44–58. [CrossRef]
141. Smilkov, D.; Thorat, N.; Kim, B.; Viégas, F.; Wattenberg, M. SmoothGrad: Removing noise by adding noise. *arXiv* **2017**, arXiv:1706.03825
142. Vig, J. *A Multiscale Visualization of Attention in the Transformer Model*; Association for Computational Linguistics: Florence, Italy, 2019.
143. Wang, N.; Wang, H.; Jia, Y.; Yin, Y. Explainable Recommendation via Multi-Task Learning in Opinionated Text Data. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018 ; Collins-Thompson, K., Ed.; ACM Conferences; ACM: New York, NY, USA, 2018; pp. 165–174. [CrossRef]
144. Kim, J.; Rohrbach, A.; Darrell, T.; Canny, J.; Akata, Z. Textual Explanations for Self-Driving Vehicles. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, , 4–14 September 2018.
145. Li, X.H.; Cao, C.C.; Shi, Y.; Bai, W.; Gao, H.; Qiu, L.; Wang, C.; Gao, Y.; Zhang, S.; Xue, X.; et al. A survey of data-driven and knowledge-aware explainable ai. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 29–49. [CrossRef]
146. Parr, T.; Grover, P. How to Visualize Decision Trees. 2020. Available online: https://explained.ai/decision-tree-viz/, (accessed on 1 December 2022).
147. Ming, Y.; Qu, H.; Bertini, E. RuleMatrix: Visualizing and Understanding Classifiers with Rules. *J. Mag.* **2018**, *25, 342–352* . [CrossRef] [PubMed]
148. Wang, K.; Zhou, S.; He, Y. Growing decision trees on support-less association rules. In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, USA, 20–23 Augusut 2000; pp. 265–269.
149. Keane, M.T.; Smyth, B. Good Counterfactuals and Where to Find Them: A Case-Based Technique for Generating Counterfactuals for Explainable AI (XAI). In *Case-Based Reasoning Research and Development*; Watson, I., Weber, R., Eds.; Lecture notes in computer science; Springer: Cham, Switzerland, 2020; pp. 163–178.
150. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Müller, A.; Nothman, J.; Louppe, G.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
151. Arya, V.; Bellamy, R.K.E.; Chen, P.-Y.; Dhurandhar, A.; Hind, M.; Hoffman, S.C.; Houde, S.; Liao, Q.V.; Luss, R.; Mojsilović, A.; et al. One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques. *arXiv* **2019**, arXiv:1909.03012.
152. Klaise, J.; van Looveren, A.; Vacanti, G.; Coca, A. Alibi: Algorithms for Monitoring and Explaining Machine Learning Models. 2019. Available online: https://github.com/SeldonIO/alibi (accessed on 1 December 2022).
153. Kokhlikyan, N.; Miglani, V.; Martin, M.; Wang, E.; Alsallakh, B.; Reynolds, J.; Melnikov, A.; Kliushkina, N.; Araya, C.; Yan, S.; et al. Captum: A unified and generic model interpretability library for PyTorch. *arXiv* **2020**, arXiv:2009.07896.
154. Przemyslaw Biecek. DALEX: Explainers for Complex Predictive Models in R. *J. Mach. Learn. Res.* **2018**, *19*, 1–5.
155. Mothilal, R.K.; Sharma, A.; Tan, C. Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations. In Proceedings of 2020 Conference on Fairness, Accountability, and Transparency, 2020, pp. 607–617. Available online: https://github.com/interpretml/DiCE (accessed on 1 December 2022).

156. Nori, H.; Jenkins, S.; Koch, P.; Caruana, R. InterpretML: A Unified Framework for Machine Learning Interpretability. *arXiv* **2019**, arXiv:1909.09223 .
157. PAIR, G. PAIR Saliency : Framework-agnostic implementation for state-of-the-art saliency methods, 2022. Available online: https://github.com/PAIR-code/saliency (accessed on 1 December 2022).
158. Oracle. Skater: Unified Framework for Model Interpretation. 2022. Available online: https://github.com/oracle/Skater (accessed on 1 December 2022).
159. Hedström, A.; Weber, L.; Bareeva, D.; Motzkus, F.; Samek, W.; Lapuschkin, S.; Höhne, M.M.C. Quantus: An explainable AI toolkit for responsible evaluation of neural network explanations. *arXiv* **2022**, arXiv:2202.06861 .
160. Dijk, O. Explainerdashboard: Quickly Deploy a Dashboard Web App for Interpretability of Machine Learning Model. 2022. Available online: https://github.com/oegedijk/explainerdashboard (accessed on 1 December 2022)
161. Alammar, J. Ecco: An Open Source Library for the Explainability of Transformer Language Models. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations. Association for Computational Linguistics, Online, 1–6 August 2021.
162. Hu, B.; Tunison, P.; Vasu, B.; Menon, N.; Collins, R.; Hoogs, A. XAITK: The explainable AI toolkit. *Appl. AI Lett.* **2021**, *2*, e40. [CrossRef]
163. *ISO/IEC TR 24028:2020*; Overview of Trustworthiness in Artificial Intelligence. International Organization for Standardization: Vernier, Switzerland, 2020.
164. Lopes, P.; Silva, E.; Braga, C.; Oliveira, T.; Rosado, L. XAI Systems Evaluation: A Review of Human and Computer-Centred Methods. *Appl. Sci.* **2022**, *12*, 9423. [CrossRef]
165. Wanner, J.; Herm, L.V.; Heinrich, K.; Janiesch, C. A social evaluation of the perceived goodness of explainability in machine learning. *J. Bus. Anal.* **2021**, *5* , 29–50. [CrossRef]
166. Robnik-Šikonja, M.; Bohanec, M. Perturbation-Based Explanations of Prediction Models. In *Human and Machine Learning*; Springer: Cham, Switzerland, 2018; pp. 159–175. [CrossRef]
167. Murdoch, W.J.; Singh, C.; Kumbier, K.; Abbasi-Asl, R.; Yu, B. Definitions, methods, and applications in interpretable machine learning. *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 22071–22080. [CrossRef]
168. Gilpin, L.H.; Bau, D.; Yuan, B.Z.; Bajwa, A.; Specter, M.; Kagal, L. Explaining explanations: An overview of interpretability of machine learning. In Proceedings of the 2018 IEEE 5th International Conference on data science and Advanced Analytics (DSAA), Turin, Italy, 1–3 October 2018 ; IEEE: Piscataway, NJ, USA, 2018, pp. 80–89.
169. Löfström, H.; Hammar, K.; Johansson, U. A Meta Survey of Quality Evaluation Criteria in Explanation Methods. In Proceedings of the International Conference on Advanced Information Systems Engineering, Leuven, Belgium, 28 June–2 July 2022 ; Springer: Berlin/Heidelberg, Germany, 2022, pp. 55–63.
170. Pavlidis, M.; Mouratidis, H.; Islam, S.; Kearney, P. Dealing with trust and control: A meta-model for trustworthy information systems development. In Proceedings of the 2012 Sixth International Conference on Research Challenges in Information Science (RCIS), Valencia, Spain, 16–18 May 2012 ; IEEE: Piscataway, NJ, USA, 2012, pp. 1–9.
171. Doshi-Velez, F.; Kim, B. Towards a Rigorous Science of Interpretable Machine Learning. *arXiv* **2017**, arXiv:1702.08608.
172. Yeh, C.K.; Hsieh, C.Y.; Suggala, A.; Inouye, D.I.; Ravikumar, P.K. On the (in) fidelity and sensitivity of explanations. *Adv. Neural Inf. Process. Syst.* **2019**, *32, 10967–10978* .
173. Montavon, G.; Samek, W.; Müller, K.R. Methods for interpreting and understanding deep neural networks. *Digit. Signal Process.* **2018**, *73*, 1–15. [CrossRef]
174. Bhatt, U.; Xiang, A.; Sharma, S.; Weller, A.; Taly, A.; Jia, Y.; Ghosh, J.; Puri, R.; Moura, J.M.F.; Eckersley, P. Explainable machine learning in deployment. In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, 27–30 January 2020; Hildebrandt, M., Ed.; ACM Digital Library; Association for Computing Machinery: New York, NY, USA, 2020; pp. 648–657. [CrossRef]
175. Nguyen, A.p.; Martínez, M.R. On quantitative aspects of model interpretability, 2020. *arXiv* **2020**, arXiv:2007.07584.
176. Rong, Y.; Leemann, T.; Borisov, V.; Kasneci, G.; Kasneci, E. A Consistent and Efficient Evaluation Strategy for Attribution Methods, 2022. *arXiv* **2022**, arXiv:2202.00449.
177. Silva, W.; Fernandes, K.; Cardoso, M.J.; Cardoso, J.S. Towards complementary explanations using deep neural networks. In *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 133–140.
178. Chalasani, P.; Chen, J.; Chowdhury, A.R.; Jha, S.; Wu, X. Concise Explanations of Neural Networks using Adversarial Training, 2018. *arXiv* **2018**, arXiv:1810.06583.
179. Li, L.; Zhang, Y.; Chen, L. Generate Neural Template Explanations for Recommendation. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual, 19–23 October 2020 ; d'Aquin, M., Dietze, S., Eds.; ACM Digital Library; Association for Computing Machinery: New York, NY, USA, 2020; pp. 755–764. [CrossRef]
180. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
181. Keane, M.T.; Kenny, E.M.; Delaney, E.; Smyth, B. If Only We Had Better Counterfactual Explanations: Five Key Deficits to Rectify in the Evaluation of Counterfactual XAI Techniques. *arXiv* **2021**, arXiv.2103.01035. [CrossRef]

182. Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I.; Hardt, M.; Kim, B. Sanity Checks for Saliency Maps. *arXiv* **2018**, arXiv:1810.03292.
183. Nyre-Yu, M.; Morris, E.; Moss, B.C.; Smutz, C.; Smith, M. Explainable AI in Cybersecurity Operations: Lessons Learned from xAI Tool Deployment. In Proceedings of the Usable Security and Privacy (USEC) Symposium, San Diego, CA, USA, 28 April 2022.
184. Mitchell, M.; Wu, S.; Zaldivar, A.; Barnes, P.; Vasserman, L.; Hutchinson, B.; Spitzer, E.; Raji, I.D.; Gebru, T. Model cards for model reporting. In Proceedings of the Conference on Fairness, Accountability, and Transparency, Atlanta, GA, USA, 2–31 January 2019; pp. 220–229.
185. Pushkarna, M.; Zaldivar, A.; Kjartansson, O. Data Cards: Purposeful and Transparent Dataset Documentation for Responsible AI. *arXiv* **2022**, arXiv:2204.01075.
186. Bender, E.M.; Friedman, B. Data Statements for Natural Language Processing: Toward Mitigating System Bias and Enabling Better Science. *Trans. Assoc. Comput. Linguist.* **2018**, *6*, 587–604. [CrossRef]
187. Gebru, T.; Morgenstern, J.; Vecchione, B.; Vaughan, J.W.; Wallach, H.; Iii, H.D.; Crawford, K. Datasheets for datasets. *Commun. ACM* **2021**, *64*, 86–92. [CrossRef]
188. Holland, S.; Hosny, A.; Newman, S.; Joseph, J.; Chmielinski, K. The dataset nutrition label. *Data Prot. Privacy* **2020**, *12*, 1.
189. Alsallakh, B.; Cheema, A.; Procope, C.; Adkins, D.; McReynolds, E.; Wang, E.; Pehl, G.; Green, N.; Zvyagina, P. *System-Level Transparency of Machine Learning*; Technical Report; Meta AI: New York, NY, USA, 2022.
190. Arnold, M.; Bellamy, R.K.; Hind, M.; Houde, S.; Mehta, S.; Mojsilović, A.; Nair, R.; Ramamurthy, K.N.; Olteanu, A.; Piorkowski, D.; et al. FactSheets: Increasing trust in AI services through supplier's declarations of conformity. *IBM J. Res. Dev.* **2019**, *63*, 1–6. [CrossRef]
191. Karunagaran, S. Making It Easier to Compare the Tools for Explainable AI. 2022. Available online: https://partnershiponai.org/making-it-easier-to-compare-the-tools-for-explainable-ai/, (accessed on 1 December 2022)
192. Belaid, M.K.; Hüllermeier, E.; Rabus, M.; Krestel, R. Do We Need Another Explainable AI Method? Toward Unifying Post-hoc XAI Evaluation Methods into an Interactive and Multi-dimensional Benchmark. *arXiv* **2022**, arXiv:2207.14160.
193. Meskauskas, Z.; Kazanavicius, E. About the New Methodology and XAI-Based Software Toolkit for Risk Assessment. *Sustainability* **2022**, *14*, 5496. [CrossRef]
194. Marín Díaz, G.; Galán, J.J.; Carrasco, R.A. XAI for Churn Prediction in B2B Models: A Use Case in an Enterprise Software Company. *Mathematics* **2022**, *10*, 3896. [CrossRef]
195. Maltbie, N.; Niu, N.; van Doren, M.; Johnson, R. XAI Tools in the Public Sector: A Case Study on Predicting Combined Sewer Overflows. In Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Athens, Greece, 23–28 August 2021; Association for Computing Machinery: New York, NY, USA, 2021; ESEC/FSE 2021; pp. 1032–1044. [CrossRef]
196. Hevner, A.R.; March, S.T.; Park, J.; Ram, S. Design science in information systems research. *MIS Q.* **2004**, *28*, 75–105. [CrossRef]