Data Mining and Knowledge Discovery (2023) 37:1719–1778
https://doi.org/10.1007/s10618-023-00933-9

Benchmarking and survey of explanation methods for black box models

Francesco Bodria[1] · Fosca Giannotti[1] · Riccardo Guidotti[3] · Francesca Naretto[1] · Dino Pedreschi[3] · Salvatore Rinzivillo[2]

Received: 24 November 2021 / Accepted: 10 March 2023 / Published online: 3 June 2023

Abstract

The rise of sophisticated black-box machine learning models in Artificial Intelligence systems has prompted the need for explanation methods that reveal how these models work in an understandable way to users and decision makers. Unsurprisingly, the state-of-the-art exhibits currently a plethora of explainers providing many different types of explanations. With the aim of providing a compass for researchers and practitioners, this paper proposes a categorization of explanation methods from the perspective of the type of explanation they return, also considering the different input data formats. The paper accounts for the most representative explainers to date, also discussing similarities and discrepancies of returned explanations through their visual appearance. A companion website to the paper is provided as a continuous update to new explainers as they appear. Moreover, a subset of the most robust and widely adopted explainers, are benchmarked with respect to a repertoire of quantitative metrics.

Responsible editor: Johannes Fürnkranz.

✉ Francesco Bodria
francesco.bodria@sns.it

✉ Francesca Naretto
francesca.naretto@sns.it

Fosca Giannotti
fosca.giannotti@isti.cnr.it

Riccardo Guidotti
riccardo.guidotti@unipi.it

Dino Pedreschi
dino.pedreschi@di.unipi.it

Salvatore Rinzivillo
salvatore.rinzivillo@isti.cnr.it

1 Scuola Normale Superiore, P.zz dei Cavalieri, 7, Pisa, Italy

2 ISTI-CNR, Via Moruzzi, 1, Pisa, Italy

3 University of Pisa, Largo Bruno Pontecorvo, 3, Pisa, Italy

Keywords Explainable artificial intelligence · Interpretable machine learning · Transparent models · Benchmarking

1 Introduction

Artificial Intelligence (AI) has become one of the most important scientific and technological areas, with a tremendous socio-economic impact in many aspects of modern society. The impressive performance of AI systems is often reached by adopting sophisticated Machine Learning (ML) models, whose complexity hides the logic of their internal processes (Miller 2019). For this reason, such models are referred to as "black-box models" (Freitas 2013; Pasquale 2015). Well-known examples of black-box models used within current AI systems include deep learning models and ensemble models such as bagging and boosting (Guidotti et al. 2019c). The high performance of such models in terms of accuracy comes together with a high degree of opaqueness that might cause developers and users to overlook issues that the models may inherit by training data due to different forms of hidden biases (Kurenkov 2020).

There is an inherent risk that relying on opaque models may lead to adopting decisions that we do not fully understand or, even worse, violate ethical principles or legal norms. These risks are particularly relevant in high-stakes decision-making scenarios, such as medicine, justice, finance, recruitment, access to public benefits, and so on (Rudin 2019); the lack of transparency and trust may explain the relatively low adoption rate of current AI-based decision support systems in the mentioned areas. Moreover, companies that embed black-box ML models in their AI products and applications risk incurring a potential loss of safety and trust (Chouldechova 2017).

In 2018, the European Parliament introduced in the GDPR[1] a set of clauses for automated decision-making in terms of *a right of explanation* for all individuals to obtain "meaningful explanations of the logic involved" when automated decision making takes place. Also, in 2019, the High-Level Expert Group on AI presented the ethics guidelines for trustworthy AI,[2] where explainability is indicated as one of the fundamental requirements for trustworthiness. Today, the same principle has become a cornerstone of the AI Act, the proposed new EU regulation establishing standardized rules on artificial intelligence.[3] Indeed, there is a widespread and increasing consensus on the urgency of implementing appropriate explanation tools, although it represents a scientific challenge that is still largely open. Meaningful explanations are fundamental for debugging models, unveiling possible biases, tackling ethical issues, and fostering trust and collaboration between humans and their AI assistants.

Unsurprisingly, as a reaction to the mentioned concerns, we have witnessed in the last years the rise of a plethora of explanation methods for black-box models (Guidotti et al. 2019c; Adadi and Berrada 2018; Arrieta et al. 2020; Theissler et al. 2022; Guidotti 2022; Yuan et al. 2020b) both from academia and industries. eXplainable

[1] https://ec.europa.eu/justice/smedataprotect/.

[2] https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai.

[3] https://digital-strategy.ec.europa.eu/en/library/proposal-regulation-laying-down-harmonised-rules-artificial-intelligence.

Artificial Intelligence (XAI) (Miller 2019) emerged as a broad and very active research area, investigating methods that provide many different types of explanations, each capturing diverse aspects of a black-box model.

The objective of this work is to provide a fresh account of the ideas and tools of the current generation of explanation methods, or explainers, from the perspective of the different types of explanations, that such methods produce as an output.[4] To this purpose, we provide a comprehensive ontology of the explanations returned, taking into account the most popular data formats: tabular data, images, and text, with an account also of time series and graphs. We mainly focus on the most recent approaches of post-hoc explainers but also consider some promising intrinsic explainers ("explainable-by-design" models).

While presenting the plethora of existing XAI methods, we provide more details on solid, pioneering approaches but try to discuss also less known recent methods. In any case, further information can be found by the reader in the companion website of this paper, the *XAI Live Survey*[5] that we are maintaining to keep pace with newly emergent methods. We report extensive examples of the various explanations for each data type, highlighting similarities and discrepancies of returned explanations through their visual appearance. Finally, a subset of methods, selected among the most robust and widely adopted, are benchmarked with respect to a collection of quantitative metrics, assessing faithfulness, stability, robustness, and execution time.

The rest of the paper is organized as follows. Section 2 summarizes existing surveys on XAI and highlights the differences between this work and previous ones. Section 3 presents the proposed explanation-based taxonomy of explanation methods, while Sect. 4 illustrates and formalizes existing evaluation measures typically used to benchmark explanation methods. Sections 5, 6 and 7 present the details of the most recent and widely adopted explanation methods, together with examples and quantitative comparison. Finally, Sect. 10 summarizes the crucial aspects that emerged from the analysis of state of the art, and briefly discusses future research directions.

2 Related works

The widespread need for XAI in the last years caused an explosion of interest in the design of explanation methods (Goebel et al. 2018). The books (Molnar 2022; Samek et al. 2019) present in detail a restricted set of the best-known methodologies to make general ML models interpretable (Molnar 2022) and to explain the outcomes of deep neural networks (Samek et al. 2019).

In (Guidotti et al. 2019c), the classification is based on four categories of problems, and the explanation methods are classified according to the problem they are able to solve. The first distinction is between *explanation by design* [(also named *intrinsic* interpretability), and *black-box* explanation (also named *post-hoc* interpretability (Adadi and Berrada 2018; Murdoch et al. 2019; Carvalho et al. 2019)]. The

[4] This work extends and completes "A Survey Of Methods For Explaining Black-Box Models" appeared in ACM Computing Surveys (CSUR), 51(5), 1–42 (Guidotti et al. 2019c).

[5] https://kdd-lab.github.io/XAISurvey/.

second distinction in (Guidotti et al. 2019c) further classifies the black-box explanation problem into model explanation, outcome explanation, and black-box inspection. Model explanation, achieved by *global* explainers (Craven and Shavlik 1995), aims at explaining the whole logic of a model. Outcome explanation, achieved by *local* explainers (Ribeiro et al. 2016; Lundberg and Lee 2017), understands the reasons for a specific prediction. Finally, the aim of the black-box inspection is to retrieve a visual representation for understanding how the black-box works. Another crucial distinction highlighted in (Martens et al. 2007; Guidotti et al. 2019c; Adadi and Berrada 2018; Došilović et al. 2018; Carvalho et al. 2019) is between *model-specific* and *model-agnostic* explanation methods. This classification depends on whether the technique adopted to explain can work only on a specific black-box model or can be adopted on any black-box.

In (Gilpin et al. 2018), the focus is to propose a unified taxonomy to classify the existing literature. The following key terms are defined: *explanation*, *interpretability*, and *explainability*. Interpretability consists of describing the internals of a system in a way that is understandable to humans. A system is called interpretable if it produces descriptions that are simple enough for a person to understand using a vocabulary that is meaningful to the user. An alternative classification of definitions is presented in (Arrieta et al. 2020), with a specific taxonomy for explainers of deep learning models. The leading concept of the classification is Responsible Artificial Intelligence, i.e., a methodology for the large-scale implementation of AI methods in real organizations with fairness, model explainability, and accountability at its core. Similarly to (Guidotti et al. 2019c), in (Arrieta et al. 2020), the term interpretability (or transparency) is used to refer to a passive characteristic of a model that makes sense for a human observer. On the other hand, explainability is an active characteristic of a model, denoting any action taken with the intent of clarifying or detailing its internal functions. Further taxonomies and definitions are presented in (Murdoch et al. 2019; Carvalho et al. 2019). Another branch of the literature review focuses on the quantitative and qualitative evaluation of explanation methods (Samek et al. 2019; Carvalho et al. 2019). Also, we are recently witnessing to XAI surveys devoted to specific topics such as XAI methods for time series classification (Theissler et al. 2022), or counterfactual explanations (Guidotti 2022; Karimi et al. 2020b). Finally, we highlight that the literature reviews related to explainability are focused not just on ML and AI but also on social studies (Miller 2019; Byrne 2019), recommendation systems (Zhang and Chen 2020), model-agents(Anjomshoae et al. 2019), and domain-specific applications such as health and medicine (Tjoa and Guan 2019).

In this survey, we propose an alternative view w.r.t. the taxonomy introduced in (Guidotti et al. 2019c) but starting from the data perspective. In light of the works mentioned above, we believe that an updated systematic categorization of explanation methods based on the type of explanation returned and their comparison is still missing in the literature and might be beneficial for end users. Indeed, users requiring an explanation first know which data type they are dealing with, then the type of explanation they can have for that data type, and finally, they can select the best XAI method that can be used to obtain such explanation among the available ones comparing the properties offered by the method as well as a first general evaluation.

3 Explanation-based taxonomy

The goal of this survey is to provide the reader with a guide to map a problem setting with a certain data type and black-box model to a set of compatible explanation methods. Thus, in this section, we present a novel taxonomy of XAI methods based on the type of explanation returned. To collect the papers presented in the following sections, we opted for a semi-systematic literature review (Snyder 2019): (i) we conducted a systematic search on Scopus using a set of XAI search terms like "explain*" or "interpretab*" in title or abstracts; (ii) we conducted a dynamic search to uncover additional papers in the different subfields (i.e., by collecting papers citing the once we found with the first search); (iii) the collected papers were analyzed by the authors based on the number of citations, venue, and availability of an open-source library.[6]

In particular, we focus our survey on explanations and explanation methods acting on the three principal data types recognized in the literature: *tabular* data, *images* and *text* (Guidotti et al. 2019c; Adadi and Berrada 2018; Miller 2019). For each data type, we have distinguished different types of explanations illustrated in Fig. 1. A reader should use Fig. 1 as follows. First, she should identify through the column header the data type of her problem setting. After that, each row offers an alternative type of explanation with an example. For instance, if we are interested in images, we should look to the second column. Here we can find saliency maps and concept attribution as image-specific explanation types. The last rows reports visualizations and examples of prototypes and counterfactuals, i.e., instance-based explanations, which are available independently from the data type analyzed. Finally, once the reader has selected the desired/most-suitable explanation type on Fig. 1, she can find in the corresponding section an overview of the most well-known and used explanation methods able to return that kind of explanation. For the sake of completeness, other types of data, increasingly present in literature, such as graphs (Yuan et al. 2020b) and time series (Theissler et al. 2022), have been included in a separate section. A Table appearing at the beginning of each subsequent section summarizes the explanation methods by grouping them accordingly to the classification illustrated in Fig. 1. Besides, in every section, we present the meaning of each type of explanation. The acronyms reported in capital letters in Fig. 1, in this section and in the following, are used in the remainder of the work to quickly categorize the various explanations and explanation methods.

In the rest of this section, we recall the existing taxonomy and classification of XAI methods present in the literature (Guidotti et al. 2019c; Adadi and Berrada 2018;

[6] In particular, we adopted the following criteria. *First, contributions with working implementation* There are several publications in the field of XAI, but the majority of them do not have a working implementation. Since the goal of this survey is also to provide a quantitative comparison among different XAI methods, the availability of an implementation is a key factor. *Second, contributions with a high number of citations*. In this survey, we discuss and also present some works that do not have implementations. These works are still considered due to the fact that they have a great number of citations (more than 150); hence, in our opinion, they are pillars of this field. *Third, XAI pillars*. We also considered some works that do not have working implementations, but they were published several years ago with a great number of citations. For this reason, we considered them pillars of this research field, and we added them to the survey. *Fourth, works considering ethical aspects*. Lastly, we considered works that tackle ethical aspects, such as privacy. This choice is due to the fact that the GDPR and the AI Act are changing the focus of the works in XAI: during the last few years, the concept of trustworthy AI is gaining a lot of interest, leading up to many works that consider not only the aspect of explanation, but also fairness, privacy, and accountability.

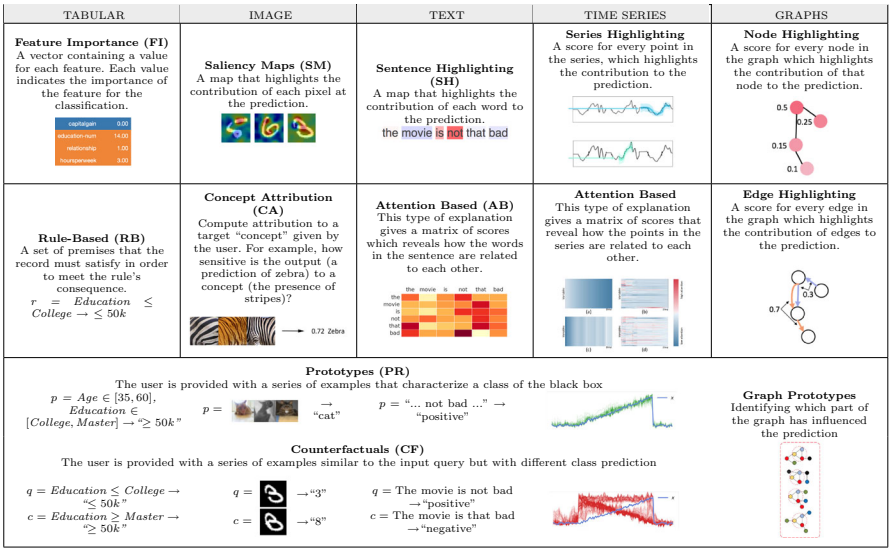

Fig. 1 Explanation-based taxonomy with examples divided for different data types

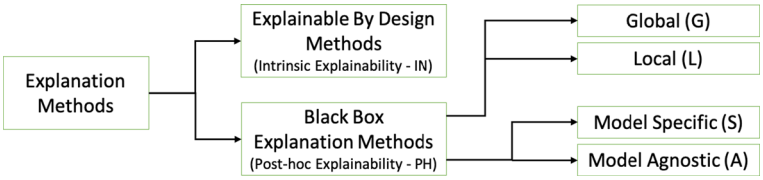

Fig. 2 Existing taxonomy for the classification of explanation methods

Gilpin et al. 2018; Arrieta et al. 2020; Samek et al. 2019; Carvalho et al. 2019) in order to provide the reader a complete overview of the proposed explanation-based categorization of explanation methods. In the following, we summarize the fundamental distinctions adopted to annotate the methods in Fig. 2:

- *INtrinsically (IN)* explainable methods are explainable by design methods that return a decision, and the reasons for the decision are directly accessible because the model is transparent.
- *Post-Hoc (PH)* explanation methods provides explanations for a black-box model.
- *Global (G)* explanation methods aim at explaining the overall logic of a black-box model. Therefore the explanation returned is a global, complete explanation valid for any instance;
- *Local (L)* explainers aim at explaining the reasons for the decision of a black-box model for a specific instance.
- *Model-Agnostic (A)* explanation methods can be used to interpret *any type* of black-box model;
- *Model-Specific (S)* explanation methods that can be used to interpret *only a specific type* of black-box model.

Moreover, to provide the reader with a self-contained review of XAI methods, we complete this section by reporting the definitions of explanation, interpretability, transparency, and complexity present in the literature:

- *Explanation* (Arrieta et al. 2020; Guidotti et al. 2019c) is an *interface* between humans and an AI decision-maker that is both comprehensible to humans and an accurate proxy of AI. Consequently, explainability is the ability to provide a *valid* explanation.
- *Interpretability*(Guidotti et al. 2019c), or comprehensibility (Gleicher 2016), is the ability of stakeholders to understand relevant aspects of the modeling process. Interpretability and comprehensibility are tied to the evaluation of the model complexity.
- *Transparency* (Arrieta et al. 2020), or equivalently understandability or intelligibility, is the capacity of a model to be interpretable itself. Thus, the model allows a human to direct understand its internal mechanism and its decision process.
- *Complexity* (Doshi-Velez and Kim 2017) is the degree of effort required by a user to comprehend an explanation. The complexity can consider the user background or eventual time limitation necessary for the understanding.

Finally, the same taxonomy and categorization adopted in the survey are exploited in a *XAI Live Survey*[7] where the existing methods are further analyzed and continuously updated with emergent approaches.

4 Evaluation measures for explanations

There is a wide debate on how to evaluate the quality of the explanation methods. Often it is formulated as properties of the returned explanations aimed at capturing concepts as goodness and usefulness of explanations (Guidotti et al. 2019c; Adadi and Berrada 2018; Gilpin et al. 2018; Arrieta et al. 2020; Samek et al. 2019; Carvalho et al. 2019). In the following, we describe a selection of established methodologies for the evaluation of explanation methods both from the quantitative and qualitative point of view which are typically used to judge the output of XAI methods.

Quantitative evaluation focuses on the performance of the explainer and on the goodness of the explanations returned. In the following, we present the general idea of each metric used later on for benchmarking. Since every metric may vary in its application depending on data type, further details are provided in the various sections.

- *Fidelity* aims to evaluate how good is f at mimicking b. There are different implementations of fidelity, depending on the type of explanator under analysis (Guidotti et al. 2019a). For example, in methods where there is a creation of a surrogate model g to mimic b, fidelity compares the prediction of b and g on the instances used to train g.
- *Stability* aims at validating if similar instances obtain similar explanations. Stability can be evaluated through the *Lipschitz constant* (Alvarez-Melis and Jaakkola 2018) as $L_x = \max \frac{\|e_x - e_{x'}\|}{\|x - x'\|}, \forall x' \in \mathcal{N}_x$ where x is the instance, e_x the explanation and $\mathcal{N}_x$ is a neighborhood of instances similar to x.

[7] https://kdd-lab.github.io/XAISurvey/.

– *Deletion* and *Insertion* (Petsiuk et al. 2018) are metrics that remove the features that the explanation method f found important and see how the performance of b degrades. The intuition behind deletion is that removing the "cause" will force the black-box to change its decision. Among the deletion methods, there is the *Faithfulness* (Alvarez-Melis and Jaakkola 2018). It aims to validate if the relevance scores indicate true importance: we expect higher importance values for attributes that greatly influence the final prediction.[8] Given a black-box model b and the feature importance e extracted from an importance-based explanator f, the faithfulness method incrementally removes each of the attributes deemed important by f. At each removal, the effect on the performance of b is evaluated. In general, a sharp drop and a low area under the probability curve mean a good explanation. On the other hand, the insertion metric takes a complementary approach. Typically, insertion and deletion evaluations are tailored for specific types of explainers called Feature Importance explainers for tabular data, Saliency Maps for image data, and Sentence Highlighting for text data.
– *Monotonicity* (Luss et al. 2021) can be seen as an implementation of an insertion method: it evaluates the effect of b by incrementally adding each attribute in order of increasing importance. In this case, we expect that the black-box performance increases by adding more and more features, thereby resulting in monotonically increasing model performance.
– *Running Time*: the time needed to produce the explanation is also an important evaluation.

It is worth noting that, to the best of our knowledge, there are currently no objective evaluation measures that can select the best explainer. A different approach to evaluating explainers consists in generating synthetic ground truth explanations and comparing them with those returned by the explainers (Guidotti 2021). However, this evaluation with synthetic explanations cannot be transferred to an objective evaluation of real data because we would not need an explainer if we knew the ground truth explanation. *Qualitative evaluation* is important to understand the actual usability of explanations from the point of view of the end-user: they satisfy human curiosity, find meanings, safety, social acceptance, and trust. In (Doshi-Velez and Kim 2017), the evaluation criteria for the qualitative evaluation are systematized into three categories:

– *Functionally-grounded* metrics aim to evaluate interpretability by exploiting some formal definitions that are used as proxies. They do not require humans for validation. The challenge is to define the proxy to employ, depending on the context. As an example, we can validate the interpretability of a model by showing the improvements w.r.t. to another model already proven to be interpretable by human-based experiments.
– *Application-grounded* evaluation methods require human experts to validate the specific task under analysis (Williams et al. 2016; Suissa-Peleg et al. 2016). They are usually employed in specific settings. For example, if the model is an assistant in the decision-making process of doctors, the validation is done by the doctors.
– *Human-grounded* metrics evaluate the explanations through humans who are not experts. The goal is to measure the overall understandability of the explanation in

[8] An implementation of faithfulness is available in AIX360 presented in Sect. 9.

simplified tasks (Lakkaraju et al. 2016; Kim et al. 2015). This validation is most appropriate for general testing notions of the quality of an explanation.

In this work, we do not provide a qualitative evaluation from the human point of view, but we try to fulfill these criteria by providing the user with a visual example of what the explanation returned by the explainer looks like. This helps the user to understand whether the explainer produced fulfills his needs.

In the following section, besides illustrating explanation methods with respect to our explanation-based taxonomy, we report a qualitative comparison of the explanations. These examples might help the reader to understand how to interpret these explanations returned by the different methods. Additionally, when possible, we benchmarked the most widely used explanation methods on a set of datasets, and we report their evaluation using the measures described in this section.[9]

5 Explanations for tabular data

In this section, we present a selection of approaches for explaining the decision systems acting on tabular data. In particular, we present the following types of explanations based on: *Features Importance* (FI, Sect. 5.1), *Rule* (RB, Sect. 5.2), *Prototype* (PR), and *Counterfactual* (CF) (Sect. 5.3). Table 1 summarizes and categorizes the explainers. The methods are sorted by the explanation type they produce. Every explanation method is provided the author's name, the year of publication, and the data type it can handle. In addition, Table 1 specifies if the method is intrinsic (IN) or Post-hoc (PH), if it provides Global explanations (G) or Local one(L), and if it is an Agnostic method(A) or a model Specific one (S). Methods with code available are highlighted in blue. After the presentation of the methods, we report the results of experiments obtained from the application of the explainers on Logistic Regression (LG), XGBoost (XGB), and Catboost (CAT) classifiers.[10] trained on the `adult` and `german` dataset.[11]

5.1 Feature importance

Feature importance is one of the most popular types of explanation returned by local explanation methods. For feature importance-based explanation methods, the explainer assigns to each feature an importance value which represents how much that particular feature was important for the prediction under analysis. Formally, given a record x, an explainer $f(\cdot)$ models a feature importance explanation as a vector

[9] All the experiments in the next sections are performed on a server with GPU: 1xTesla K80, compute 3.7, having 2496 CUDA cores, 12GB GDDR5 VRAM, CPU: 1xsingle core hyperthreaded Xeon Processors @2.3Ghz, i.e. (1 core, 2 threads) with 16 GB of RAM, or on a server: CPU: 16x Intel(R) Xeon(R) Gold 5120 CPU @ 2.20GHz (64 bits), 63 GB RAM. The code for reproducing the results is available https://github.com/kdd-lab/XAI-Survey.

[10] LG: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html XGB: https://xgboost.readthedocs.io/en/stable/python/python_intro.html CAT: https://catboost.ai/en/docs/concepts/python-usages-examples

[11] `adult`: https://archive.ics.uci.edu/ml/datasets/adult, `german`: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data).

Table 1 Summary of methods for explaining black-boxes for tabular data

Type	Name	Ref.	Data type	IN/PH	G/L	A/S
FI	LRP	Bach et al. (2015)	ANY	PH	L	A
	LIME	Ribeiro et al. (2016)	ANY	PH	L	A
	SHAP	Lundberg and Lee (2017)	ANY	PH	G/L	A
	MAPLE	Plumb et al. (2018)	TAB	PH/IN	L	A
	EBM	Nori et al. (2019)	TAB	IN	G/L	A
	NAM	Agarwal et al. (2021)	TAB	IN	L	S
	CIU	Anjomshoae et al. (2020)	TAB	PH	L	A
	EEM	Chowdhury et al. (2022)	TAB	PH	G	A
	DALEX	Lipovetsky (2022)	ANY	PH	G/L	A
RB	TREPAN	Craven and Shavlik (1995)	TAB	PH	G	S
	MSFT	Chipman et al. (1998)	TAB	PH	G	S
	CMM	Domingos (1998)	TAB	PH	G	S
	DECTEXT	Boz (2002)	TAB	PH	G	S
	STA	Zhou and Hooker (2016)	TAB	PH	G	S
	SCALABLE- BRL	Yang et al. (2017)	TAB	IN	G/L	A
	LORE	Guidotti et al. (2019a)	TAB	PH	L	A
	RULEMATRIX	Ming et al. (2019)	TAB	PH	G/L	A
	ANCHOR	Ribeiro et al. (2018)	ANY	PH	G/L	A
	GLOCALX	Setzu et al. (2019)	TAB	PH	G/L	A
	SKOPERULE	Friedman and Popescu (2008)	TAB	PH	G/L	A
PR	PS	Bien and Tibshirani (2011)	TAB	IN	G/L	S
	MMD- CRITIC	Kim et al. (2016)	ANY	IN	G	S
	PROTODASH	Gurumoorthy et al. (2019)	ANY	IN	G	A
	TSP	Tan et al. (2020)	TAB	PH	L	S
CF	CEM	Dhurandhar et al. (2018)	ANY	PH	L	S
	CFX	Albini et al. (2020)	TAB	PH	L	S
	DICE	Mothilal et al. (2020)	TAB	PH	L	A
	C- CHAVE	Pawelczyk et al. (2020)	TAB	PH	L	A
	FACE	Poyiadzi et al. (2020)	ANY	PH	L	A
	ARES	Ley et al. (2022)	TAB	PH	G	A

The methods are sorted by explanation type: Features Importance (FI), Rule-Based (RB), Counterfactuals (CF), Prototypes (PR), and Decision Tree (DT). For every method, there is a data type on which it is possible to apply it: only on tabular (TAB) or any data (ANY). If it is an Intrinsic Model (IN) or a Post-Hoc one (PH), a local method (L) or a global one (G), and finally, if it is model-agnostic (A) or model-specific (S)

$e = \{e_1, e_2, \ldots, e_m\}$, in which the value $e_i \in e$ is the importance of the i^{th} feature for the decision made by the black-box model $b(x)$. For understanding the contribution of each feature, the sign and the magnitude of each value e_i are considered. W.r.t. the sign, if $e_i < 0$, it means that the feature contributes negatively to the outcome y; otherwise, if $e_i > 0$, the feature contributes positively. The magnitude, instead, represents how great the contribution of the feature is to the final prediction y. In particular, the greater

the value of $|e_i|$, the greater its contribution. Hence, when $e_i = 0$, it means that the i^{th} feature is showing no contribution to the output decision. An example of a feature based explanation is $e = \{age = 0.8, income = 0.0, education = -0.2\}$, $y = deny$. In this case, *age* is the most important feature for the decision *deny*, *income* is not affecting the outcome, and *education* has a small negative contribution.

LIME, Local Interpretable Model-agnostic Explanations (Ribeiro et al. 2016), is a local model-agnostic explainer that returns explanations as feature importance vectors. The main idea of LIME is that the explanation may be derived locally from records generated randomly in the neighborhood of the instance that has to be explained. The key factor is that it samples instances both in the vicinity of x (which have a high weight) and far away from x (low weight), exploiting π_x, a proximity measure able to capture the locality. We denote b as the black-box and x as the instance we want to explain. To learn the local behavior of b, LIME draws samples weighted by π_x. It samples these instances around x by drawing nonzero elements of x uniformly at random. This gives to LIME a perturbed sample of instances $\{z \in \mathbb{R}^d\}$ to fed to the model b and obtain $b(z)$. They are then used to train the explanation model $g(\cdot)$: a sparse linear model on the perturbed samples. The local feature importance explanation consists of the weights of the linear model. A number of papers focus on overcoming the limitations of LIME, providing several variants of it. DLIME (Zafar and Khan 2019) is a deterministic version in which the neighbors are selected from the training data by an agglomerative hierarchical clustering. ILIME (ElShawi et al. 2019) randomly generates the synthetic neighborhood using weighted instances. ALIME (Shankaranarayana and Runje 2019) runs the random data generation only once at "training time". KL- LIME (Peltola 2018) adopts a Kullback–Leibler divergence to explain Bayesian predictive models. QLIME (Bramhall et al. 2020) also consider nonlinear relationships using a quadratic approximation. As presented in the following, LIME can also be used on other data types. In Fig. 3 (upper part) are reported examples of LIME[12] explanations relative to our experimentation on `adult` (a/b) and `german` (c/d).[13] We predicted the same record using LG and CAT, and then we explained it. Interestingly, for `adult` (plots a/b), LIME considers a similar set of features as important (even if with different values of importance) for the two models: on 6 features, only one differs. A different scenario is obtained applying LIME on `german` (plots c/d): different features are considered important by the two classifiers. However, the confidence of the prediction between the two classifiers is quite different: both of them predict the output correctly, but CAT has a higher value, suggesting that this could be the cause of differences between the two explanations.

SHAP, SHapley Additive exPlanations (Lundberg and Lee 2017), is a local model-agnostic explanation method computing features importance by means of Shapley values,[14] a concept from cooperative game theory. The explanations returned by SHAP are *additive feature attributions* and respect the following definition: $g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i$, where $z' \approx x$ as a real number, $z' \in [0, 1]$, $\phi_i \in \mathbb{R}$ are effects assigned to each feature, while M is the number of simplified input features. SHAP retains

[12] We refer to the original version of LIME.

[13] For reproducibility reasons, we fixed the random seed.

[14] We refer the interested reader to: https://christophm.github.io/interpretable-ml-book/shapley.html.

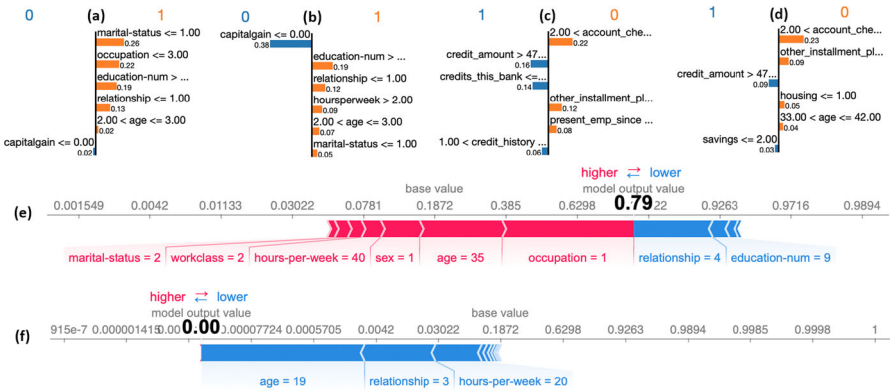

Fig. 3 *TOP*: LIME application on the same record for `adult` (a/b), `german` (c/d): a/c are the LG model explanation and b/d the CAT model explanation. All the models correctly predicted the output class. *BOTTOM*: Force plot returned by SHAP explaining XGB on two records of `adult`: (e), labeled as class 1 ($> 50K$) and, (f), labeled as class 0 ($\leq 50K$). Only the features that contributed more (i.e. higher values) to the classification are reported

three properties: (i) *local accuracy*, meaning that $g(x)$ matches $b(x)$; (ii) *missingness*, which allows for features $x_i = 0$ to have no attributed impact on the SHAP values; (iii) *stability*, meaning that if a model changes so that the marginal contribution of a feature value increases (or stays the same), the SHAP value also increases (or stays the same). The construction of the SHAP values allow to employ them both *locally*, in which each observation gets its own set of SHAP values, and *globally*, by exploiting collective SHAP values. We highlight that SHAP can be realized through different explanation models that differ in how they approximate the computation of the SHAP values. In particular, there are five strategies: *KernelExplainer* is the model-agnostic one, while *LinearExplainer*, *TreeExplainer*, *GradientExplainer*, and *DeepExplainer* are model-specific. Besides, similarly to LIME, SHAP can be used on other data types. We applied *LinearExplainer* to LG, *TreeExplainer* to XGB, and *KernelExplainer* to CAT. In Fig. 3 (lower part), we report the application of SHAP on `adult` through the *force plot* showing each feature contributes to push the output value away from the base value, which is an average among the training dataset's output values. The red features are pushing the output value higher, while the ones in blue are pushing it lower. For each feature, it is reported the actual value for the record under analysis. Only the features with the highest SHAP values are shown in this plot. In the first force plot, the features that are pushing the value higher are contributing more to the output value: from a base value of 0.18, it is reached an actual output value of 0.79. In the force plot on the right, the output value is 0.0, and *Age*, *Relationship*, and *Hours Per Week* are contributing to pushing it lower. Figure 4 (left and center) depicts the SHAP values through a *decision plots*: the contribution of all the features are reported in decreasing order of importance. The line represents the feature importance for the record under analysis, and it starts at its actual output value. In the first plot, predicted as $> 50k$, *Occupation* is the most important feature, followed by *Age* and *Relationship*. For the second plot, *Age*, *Relationship*, and *Hours Per Week* are the most important ones. SHAP also offers a

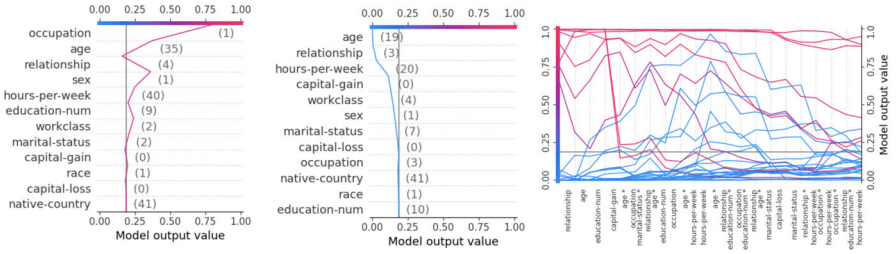

Fig. 4 SHAP application on `adult`: a record labelled $> 50K$ (top-left) and one as $\leq 50K$ (top-right). They are obtained by applying the TreeExplainer on an XGB model and then the *decision plot*, in which all the input features are shown. At the bottom, the application of SHAP to explain the outcome of a set of records by XGB on `adult`. The interaction values among the features are reported

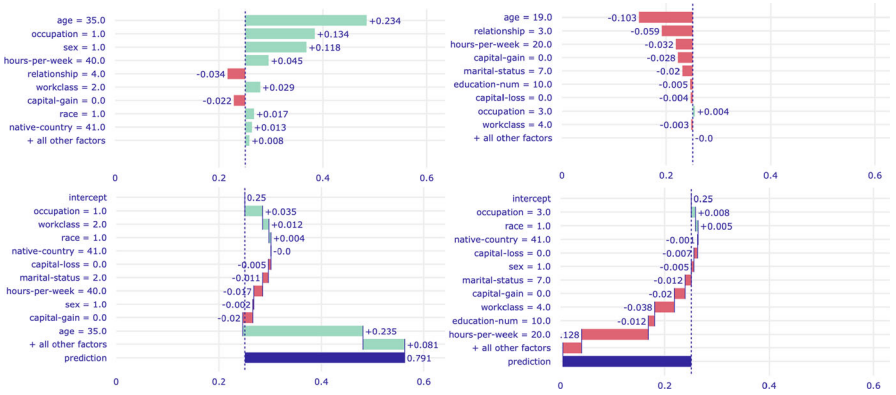

Fig. 5 Explanations of DALEX for two records of `adult`: $b(x) = 0\ (\leq 50)$ (left), $b(x) = 1\ (> 50K)$ (right) to explain an XGB in the form of Shapley values (top), break down plots (bottom). y-axis is the feature importance, x-axis is the positive/negative contribution

global interpretation of the model-driven by the local interpretations. Figure 4 (right) reports a global decision plot that represents the feature importance of 30 records of `adult`.

DALEX (Lipovetsky 2022) is a local and global, post-hoc, model-agnostic explanation method. DALEX reveals the features importance through an implementation of a *variable attribution* approach (Robnik-Šikonja and Kononenko 2008) that consists of a decomposition of the model's predictions in which each decomposition can be seen as a local gradient, and it is used to identify the contribution of each attribute. DALEX also allows the calculus of SHAP values. In Fig. 5 are reported local explanations returned by DALEX for XGB on`adult`. On the left are reported two explanations for a record classified as class $> 50k$, while on the right, for one classified as $< 50k$. On the top, there is a visualization based on Shapley values, which highlights as most important the feature *Age* (35 years old), followed by *occupation*. At the bottom, there is a *Breakdown* plot, in which the green bars represent positive changes in the mean predictions, while the red ones are negative changes. The plot also shows the *intercept*, which is the overall mean value for the predictions. It is interesting to see that *Age* and

occupation are the most important features that positively contributed to the prediction for both plots. In contrast, *Sex* is positively important for Shapley values but negatively important for the *Breakdown* plot. In this case, there are important differences in the feature considered most important by the two methods: for the Shapley values, *Age* and *Relationship* are the two most important features, while in the Breakdown plot *Hours Per Week* is the most important one.

MAPLE (Plumb et al. 2018) is a local post-hoc model-agnostic explainer that can also be used as a transparent model due to its internal structure. It combines random forests with feature selection methods to return feature importance based explanations. In particular, MAPLE is based on *SILO*, employed for obtaining a local training distribution based on the random forest leaves, and on *DStump* used to rank the features by importance. MAPLE considers the best k features from *DStump* to solve a weighted linear regression problem. Similarly to LIME, it returns these coefficients as features explanation.

CIU, Contextual Importance and Utility (Anjomshoae et al. 2020) is a local, post-hoc, model-agnostic explainer based on the idea that a feature that might be important in a context may be irrelevant in another one. CIU explains the model's outcome based on the *contextual importance (CI)* approximating the overall importance of a feature in the current context, and on the *contextual utility (CU)* estimating how good the current feature values are for a certain output class. CI and CU are calculated through Monte Carlo simulations.

EBM, Explainable Boosting Machine (Nori et al. 2019) is an intrinsic local and global model specific method. EBM is a variant of a Generalized Additive Model (GAM) (Hastie and Tibshirani 1990), i.e., a generalized linear model that incorporates nonlinear forms of the predictors. For each feature, EBM uses a boosting procedure to train the generalized linear model: it cycles over the features in a round-robin fashion to train one feature function at a time and mitigate the effects of co-linearity. In this way, the model learns the best set of feature functions, which can be exploited to understand how each feature contributes to the final prediction.

NAM, Neural Additive Models (Agarwal et al. 2021) is a local model-specific explainer defined as a variant of GAM but tailored for neural networks. NAM aims at combining the performance of deep neural networks with the inherent intelligibility of GAM. NAM is able to learn graphs that describe how the prediction is computed by training multiple deep neural networks in an additive fashion such that each network attends to a single feature.

CoFrNets Continued Fractions Nets (Puri et al. 2021) is similar to NAM, but instead of approximating activations using neural networks, it uses continued functions. The output of a neuron is calculated as a continuous fraction of the previous one until it gets to the input layer. The propriety of continued fractions to represent every possible real number allows CoFrNets to express any possible function as in Neural Networks. On the other hand, since it is a simple fraction calculation, it is possible to compute the contribution of each input to the final output and produce feature importance explanations.

Features Importance-based Explainers Comparison. Feature importance-based explanation methods provide an importance value for each feature of the record in the input. The importance of the features is computed in different ways, depending

on the kind of explanation methods exploited. The majority of the explainers are post-hoc and local, even if there are examples of methods that also provide global explanations, allowing an in-depth analysis of the overall behavior of the machine learning model (SHAP, DALEX and CIU). Some explainers, such as LIME and all its variants, create a synthetic neighborhood set used to train a surrogate model and extract the features' importance from it. These methods are suitable for context in which the explanation is online, since they are very efficient, as they use randomization techniques and surrogate models that are very simple and quick to train. As a weak point, the randomicity and the simplicity of the surrogate models may not best represent the data space under analysis. On the other hand, explainers that do not require the creation of a surrogate model but are based on some mathematical procedure, such as game theory for SHAP, the decomposition of predictions exploiting local gradients for DALEX, or Monte Carlo simulations for CIU might require a longer computational time w.r.t. LIME-like approaches. In addition, their internal workings may be difficult to understand, shifting the difficulty from understanding the machine learning model to understanding how the explanation method works. Among feature importance-based explainers, model-specific explainers, such as NAM and CoFrNets, are tailored for explaining neural networks and aim at approximating the activation functions. Overall, these explanation methods are fast, except for the model-agnostic variants of SHAP and DALEX, because they might require a greater computing time due to their different approximations. Unfortunately, the output provided by these explainers is usually quite difficult to understand for non-experts since there are several variables, and the plots provided are usually non-self-explanatory. As an example, we can think of SHAP: in the plots in output, each feature importance is given by the output value, the base value, and, depending on the kind of explanator exploited, one or more arrays of feature importance.

> Feature importance explainers, due to their complexity in the understating of the explanation, may be better suited for domain experts who know the meaning of the features employed, while they may be too difficult for ordinary end-users, especially when obtaining such importance values is complex.

5.2 Rule-based explanation

Decision rules give the end-user an explanation about the reasons that lead to the final prediction. A decision rule r, also called *factual* or *logic* rule (Guidotti et al. 2019a), has the form $p \rightarrow y$, in which p is a premise, composed of a Boolean condition on feature values, while y is the consequence of the rule. In particular, p is a conjunction of split conditions of the form $x_i \in [v_i^{(l)}, v_i^{(u)}]$, where x_i is a feature and $v_i^{(l)}$, $v_i^{(u)}$ are lower and upper bound values in the domain of x_i extended with $\pm\infty$. An instance x *satisfies* r, or r *covers* x, if every Boolean conditions of p evaluate to true for x. If the instance x to explain satisfies p, the rule $p \rightarrow y$ represents then a candidate explanation of the decision $g(x) = y$. Moreover, if the interpretable predictor mimics the behavior of the black-box in the neighborhood of x, we further conclude that the rule is a candidate local explanation of $b(x) = g(x) = y$. We highlight that, in the context of rules, we can also find the so-called *counterfactual rules* (Guidotti et al. 2019a). Counterfactual

$x_1 =$ { *Education* = *Bachelors*, *Occupation* = *Prof-specialty*, *Sex* = *Male*, *NativeCountry* = *Vietnam*, *Age* = *35*, *Workclass* = *3*, *HoursWeek* = *40*, *Race* = *Asian-Pac-Islander*, *MaritialStatus* =*Married-civ*, *Relationship* = *Husband*, *CapitalGain* = *0*, *CapitalLoss* = *0*}, $> 50k$

$r_{anchor} =$ { *EducationNum* $>$ *Bachelors*, *Occupation* $\leq$ *3.00*, *HoursWeek* $>$ *20*, *Relationship* $\leq$ *1.00*, *34* $<$ *Age* $\leq$ *41* } $\rightarrow > 50k$

$r_{lore} =$ { *Education* $>$ *5-6th*, *Race* $>$ *0.86*, *WorkClass* $\leq$ *3.41*, *CapitalGain* $\leq$ *20000*, *CapitalLoss* $\leq$ *1306* } $\rightarrow > 50k$

$c_{lore} =$ {*CapitalLoss* $\geq$ *436* } $\rightarrow \leq 50k$

$x_2 =$ { *Education* = *College*, *Occupation* = *Sales*, *Sex* = *Male*, *NativeCountry* = *US*, *Age* = *19*, *Workclass* = *2*, *HoursWeek* = *15*, *Race* = *White*, *MaritialStatus* = *Married-civ*, *Relationship* = *Husband*, *CapitalGain* = *2880*, *CapitalLoss* = *0* }, $\leq 50k$

$r_{anchor} =$ {*Education* $\leq$ *College*, *MaritialStatus* $>$ *1.00* } $\rightarrow \leq 50k$

$r_{lore} =$ {*Education* $\leq$ *Masters*, *Occupation* $>$ *-0.34*, *HoursWeek* $\leq$ *40*, *WorkClass* $\leq$ *3.50* *CapitalGain* $\leq$ *10000*, *Age* $\leq$ *34*} $\rightarrow \leq 50k$

$c_{lore} =$ {*Education* $>$ *Masters* } $\rightarrow > 50k$
{*CapitalGain* $>$ *20000* } $\rightarrow > 50k$
{*Occupation* $\leq$ *-0.34* } $\rightarrow > 50k$

Fig. 6 Explanations of ANCHOR and LORE for `adult` and XGB. x_i is the input, r_{method} are the rules provided by ANCHOR/LORE, while c_{lore} are the counterfactual rules of LORE

rules have the same structure as decision rules, with the only difference being that the consequence of the rule $\overline{y}$ is different w.r.t. $b(x) = y$. They are important to explain to the end-user what should be changed to obtain a different output. An example of a rule explanation is $r = \{age < 40, income < 30k, education \leq Bachelor\}, y = deny$. In this case, the record $\{age = 18, income = 15k, education = Highschool\}$ satisfies the rule above. A possible counterfactual rule, instead can be: $r = \{income > 40k, education \geq Bachelor\}, y = allow$.

ANCHOR (Ribeiro et al. 2018) is a global and local model-agnostic method that outputs rules, called *anchors*, as explanations. The idea is that, for decisions on which the anchor holds, changes in the rest of the instance's feature values do not change the outcome. Formally, given a record x, r is an anchor if $r(x) = b(x)$. To obtain the anchors, ANCHOR perturbs the instance x, obtaining a set of synthetic records employed to extract anchors with precision above a user-defined threshold. ANCHOR exploits a multi-armed bandit algorithm (Katehakis and Jr. 1987) for the synthetic generation of the dataset and relies on a bottom-up approach and a beam search to find the anchors. Figure 6 reports some rules obtained by applying ANCHOR to explain XGB trained on `adult`. The first rule has a high precision (0.96%) but a very low coverage (0.01%). It is interesting to note that the first rule contains *Relationship* and *Education Num*, which are the features highlighted by most of the explainers analyzed so far. In particular, in this case, for having a classification $> 50k$, the *Relationship* should be husband and the *Education Num* at least a bachelor's degree. *Education Num* can also be found in the second rule, in which case has to be less or equal to College, followed by the *Maritial Status*, which can be anything other than married with a civilian. This rule has an even better precision (0.97%) and suitable coverage (0.37%).

LORE, LOcal Rule-based Explainer (Guidotti et al. 2019a), is a local model-agnostic explainer that provides explanations in the form of rules and counterfactual rules. LORE is tailored explicitly for tabular data. It exploits a genetic algorithm to create the neighborhood of the record to explain. Such a neighborhood produces a more faithful and dense representation of the vicinity of x w.r.t. LIME. Given a black-box model b and an instance x, with $b(x) = y$, LORE first generates a synthetic set Z of neighbors through a genetic algorithm. Then, it trains a decision tree classifier g on this set labeled with the black-box outcome $b(Z)$. From g, it retrieves an explanation that consists of (i) a *factual* decision rule, that corresponds to the path on the decision tree followed by the instance x to reach the decision y, and (ii) a set of counterfactual rules, which have a different classification w.r.t. y. These counterfactual rules show the conditions that can be varied on x in order to change the output decision. In Fig. 6, we report the factual and counterfactual rules of LORE for the explanation of the same records showed for ANCHOR. It is interesting to note that, differently from ANCHOR and the other models proposed above, LORE explanations focus more on the *Education Num*, *Occupation*, *Capital Gain* and *Capital Loss*, while the features about the relationship are not present.

RuleMatrix (Ming et al. 2019) is a model-agnostic explainer that provides both local and global explanations specifically tailored for the visualization of the rules extracted. Given a training dataset and a black-box, RULEMATRIX executes a rule induction step, in which a *rule list* is extracted by sampling the input data and their predicted label by the black-box. Then, the rules extracted are filtered based on thresholds of confidence and support. Finally, it outputs a visual representation of the rules.

Local rule-based explainers comparison The rule-based methods presented are all based on creating a surrogate model from which to extract the rules. In this category, we find ANCHOR and RULEMATRIX, which provide both local and global explanations by relying on simple rule extraction methods. The simplicity of these methods makes them efficient even if, as in the case of LIME-like approaches, they may suffer in terms of the goodness of explanations provided. LORE is the only explainer that provides only local explanations. Differently from the other methods, does not require to have access to the original training data, and, due to its synthetic generation process, provides more faithful explanations. Therefore it may be good in settings where the black-box training dataset is unavailable, while RULEMATRIX and ANCHOR need to access the training data. Rule-based explanations are considered closer to human reasoning w.r.t. the feature importance-based explanations. In addition, they exploit easy to understand mechanisms, allowing users of different backgrounds to understand how the explanation method works, increasing trust. However, these explainers usually require a greater running time w.r.t. the feature importance ones.

> Local rule-based explainers produce logical rules which are close to human reasoning and make them suitable for non-experts.

Global tree-based explainers One of the most popular ways to generate explanation rules is by extracting them from a decision tree. In particular, due to the method's simplicity and interpretability, decision trees explain black-box models' overall behavior. Some explanation methods acting in this setting are model-specific explainers exploiting structural information of the black-box model under analysis. TREPAN (Craven

and Shavlik 1995) is a model-specific global explainer tailored for neural networks. Given a neural network b, TREPAN generates a decision tree g that approximates the network by maximizing the gain ratio and the model fidelity. In particular, to leverage abstraction, TREPAN adopts n-of-m decision rules on which only n out of m conditions must be satisfied in order to fire the rule. DecText is a global model-specific explainer tailored for neural networks (Boz 2002). DECTEXT resembles TREPAN with the difference that it considers four different splitting methods. Moreover, it also considers a pruning strategy based on fidelity to reduce the final explanation tree's size. In this way, DECTEXT can maximize the fidelity while keeping the model simple. Both TREPAN and DECTEXT are presented as model-specific explainers, but they can be practically employed to explain any black-box as they do not use any internal information of neural networks. MSFT (Chipman et al. 1998) is a global, post-hoc, model-specific explainer for random forests that returns a decision tree. MSFT is based on the observation that, even if random forests contain hundreds of different trees, they are quite similar, differing only for a few nodes. Hence, it adopts dissimilarity metrics to summarize the random forest trees using a clustering method. Then, for each cluster, an archetype is retrieved as an explanation. CMM, Combined Multiple Model procedure (Domingos 1998), is another global, post-hoc, model-specific explainer for tree ensembles. The key point of CMM is the data enrichment. Given an input dataset X, CMM first modifies it n times. On the n variants of the dataset, it learns a black-box. Random records are generated and labeled using a bagging strategy on the black-boxes. The authors were able to increase the size of the dataset to build the final decision tree. STA, Single Tree Approximation (Zhou and Hooker 2016), is another global, post-hoc, model-specific explainer tailored for random forests. In STA, the decision tree is constructed by exploiting test hypothesis on the trees in the forest to find the best splits.

The explainers proposed are tailored for a specific machine learning model: TREPAN and DECTEXT explain neural networks, while CMM and STA are tailored for random forests and MSFT is for any ensemble method. Among them, some explainers exploit an enrichment of the data to improve the extraction of the tree (CMM, TREPAN, DECTEXT), while the others exploit the training dataset by applying some strategies based on dissimilarity metrics (MSFT) or test hypothesis (STA). Among the different methods, only DECTEXT and TREPAN apply some strategies with the goal of maximizing the model fidelity, even if they are tailored for small feed-forward neural networks. The exploitation of trees to explain the global behavior of a more complex machine learning model have several benefits, such as fast computation and a simple process to extract explanations based on transparent strategies. However, the trees extracted may be very deep, making the explanation model difficult to understand even in cases of simple datasets. Furthermore, the effectiveness of such explanations for very deep feed forward networks has not been judged yet.

> Global tree-based explainers produce a transparent model allowing the understanding of the general behavior of the black-box. Depending on the complexity of the tree, the actual ease of understanding of the explanation could be affected by this.

Global rule-based explainers In this section, we present global explainers that do not extract decision trees as a global interpretable model but as lists or sets of rules. The

majority of the methods described in the following extract rules by exploiting ensemble methods or rule-based classifiers. The explainers considered are all agnostic. Skope-Rules is a global, post-hoc, model-agnostic[15] explainer on the RULEFIT (Friedman and Popescu 2008) idea to define an ensemble method and then extract the rules from it. SKOPE- RULES employs fast algorithms such as bagging or gradient boosting decision trees. After extracting all the possible rules, SKOPE- RULES removes rules redundant or too similar by a similarity threshold. Differently from RULEFIT, the scoring method does not solve the L1 regularization. Instead, the weights are given depending on the precision score of the rule. Scalable-BRL (Yang et al. 2017) is an interpretable rule-based model that optimizes the posterior probability of a Bayesian hierarchical model over the rule lists. The theoretical part of this approach is based on (Letham et al. 2015). GLocalX (Setzu et al. 2021) is a global model-agnostic post-hoc explainer that adopts the *local to global* paradigm, i.e., to derive a global explanation by subsuming local logical rules. GLOCALX starts from an array of local explanation rules and follows a hierarchical bottom-up approach merging similar records expressing the same conditions. This small section comprises global explanation methods that extract rules in entirely different ways: either they exploit an ensemble method (SKOPE- RULES), a rule-based model (SCALABLE- BRL) or several local explanations (GLOCALX). In terms of goodness of explanations, SKOPE- RULES and SCALABLE- BRL are tailored for an overall explanation of the machine learning model, focusing mostly on the data in input. GLOCALX, instead, exploits local explanations and hence is able to tackle the problem from a different point of view, merging several local explanations. The output of these methods is a list of rules, and even if there are techniques to filter out meaningless rules, the complexity of the explanation produced may be huge.

> Global rule-based explainers produce sets of rules describing the overall behavior of the model for each target class. Depending on the filters applied, the list of rules extracted may be long and difficult to understand.

Rules-based explainers comparison In this section, we presented a great variety of methods that provide logical rules as explanations exploiting different strategies. Independently from the strategy, due to the simplicity of the rules, they are often the preferred explanation for non-expert people. The majority of the explainers presented in this section are based on the extraction of decision trees as surrogate models (LORE, TREPAN, CMM, STA, DECTEXT, MSFT), or ensemble methods based on decision trees, such as SKOPE- RULES. The remaining methods that do not rely on decision trees extract the rules in other ways, such as rule-based classifiers (again a surrogate model), as in the case of ANCHOR, SCALABLE- BRL and of RULEMATRIX. To further increase the comprehensibility of the explanation, some explainers correlate the explanations by graphical visualizations, such as RULEMATRIX, ANCHOR, and SKOPE- RULES. Overall, the majority of the explainers require a long computing time due to the different enrichment of the data or the use of rule-based classifiers, which are among the longest interpretable models to train. Hence, they may be better fitted for offline explanations. Depending on the complexity of the machine learning model in input, the explanations may be complex, such as deep trees or long lists of rules.

[15] https://skope-rules.readthedocs.io/en/latest/skope_rules.html.

Rules-based explanation methods extract rules exploiting different approaches, which may require a longer time w.r.t. feature importance methods, making them more suitable to offline settings. However, rules-based methods are tailored for common end-users due to their logical structure and simplicity.

5.3 Prototype-based explanations

A prototype, also called archetype or artifact, is a record highlighting the characteristics which identify a group of objects belonging to the same class. Prototypes serve as examples, i.e., the user can understand the black-box reasoning by looking at records similar to the prototype. Thus, a prototype is a local explanation. A prototype can be (i) a record of the training set close to the input data x, (ii) a centroid of a cluster to which the input x belongs to, or (iii) even a synthetic record generated following an ad-hoc process. Depending on the explanation method considered, different definitions and requirements to find a prototype can be considered. MMD-CRITIC (Kim et al. 2016) is a *before the model* explanation method which produces prototypes and criticisms as explanations using *Maximum Mean Discrepancy (MMD)* measure. While prototypes explain the dataset's general behavior, criticisms represent records that are not well explained by the prototypes. MMD- CRITIC selects prototypes by measuring the difference between the distribution of the instances and the instances in the whole dataset. The set of instances nearer to the data distribution are called prototypes, and the farthest are called criticisms. ProtoDash (Gurumoorthy et al. 2019) is a variant of MMD- CRITIC. However, differently from MMD- CRITIC, PROTODASH also returns non-negative weights which indicate the importance of each prototype. PS, Prototype Selection (PS) (Bien and Tibshirani 2011) seeks a set of prototypes that better represent the data under analysis by solving a set cover optimization problem with constraints on the properties of the prototypes. After that, the prototypes are employed to learn a nearest neighbor rule classifier to be used as a model. TSP, Tree Space Prototype (Tan et al. 2020), is a local post-hoc explainer tailored for explaining tree ensemble methods. The goal of TSP is to find prototypes for each class in the tree space of the tree ensemble b w.r.t. a given notion of proximity between trees. Privacy-Preserving Explanations(PPE) (Blanco-Justicia et al. 2020) is a local post-hoc model-agnostic explainer that outputs prototypes and shallow trees as explanations while considering the concept of *privacy in explainability* while producing *privacy protected explanations*. The trade-off between privacy and comprehensibility is obtained through *micro aggregation* of the data, i.e., clustering. The clusters' centroids are used as prototypes for the final explanation/prediction.

The strength of prototypes is the possibility of analyzing black-box behavior by comparison between the record under analysis and its analogs, which is a type of reasoning widely used by humans. Moreover, they allow data analysis before and after the black-box is applied. For this reason, in this section, we may find explanation methods that are *before the model*, i.e., they explain the dataset without considering the black-box model, such as MMD- CRITIC, PS and PROTODASH. On the other hand, local post-hoc explainers, such as TSP and PPE, provide prototypes based on the decisions of the black-boxes. Among the different methods proposed, one of the most promising

ones is MMD- CRITIC because it outputs both prototypes and criticisms. In this setting, we can also find a novel application, namely PPE, which produces privacy-protected prototypes, creating a link between two crucial ethical concepts: transparency and privacy. Indeed, using prototypes as explanations, although it may be useful for end-user understanding, may release sensitive information about the users in the training set when the explanation method exploits the training dataset.

Prototype-based explanations allow the users to reason by similarity and differences. Most of the methods in this setting are tailored to explain the data in input and not the black-box decisions.

5.4 Counterfactual-based explanations

Counterfactual explanations suggest what should be different in the input instance to change the outcome of the black-box model (Wachter et al. 2017; Lucic et al. 2020), i.e., they describe a dependency on the attributes that led to a particular decision. Counterfactual explanations can be considered as prototypes' opposite. Thus, also counterfactuals are local explanations. In (Guidotti 2022), counterfactual explanations are formalized as follows. Given a black-box model b that outputs the decision $y = b(x)$ for an instance x, a counterfactual explanation consists of an instance x' such that the decision for b on x' is different from y, i.e., $b(x') \neq y$, and such that the difference between x and x' is *minimal*. The different values between x and a counterfactual x' reveal what should have been different in x for having a different outcome. An ideal counterfactual is *minimal* because it should alter the values of the variables as little as possible to find the closest setting under which y is returned instead of $\neg y$. Concerning counterfactual explanations, there are many properties that are desired for this kind of explanation and for the explanation methods returning them. Examples are validity, minimality, similarity, plausibility, discriminative power, actionability, causality, diversity, efficiency, robustness, etc. (Wachter et al. 2017; Karimi et al. 2020a; Kanamori et al. 2020). To better understand the complex context and the many available possibilities, we refer the interested reader to (Guidotti 2022; Artelt and Hammer 2019; Verma et al. 2020; Byrne and Johnson-Laird 2020) while we briefly present only the most representative methods in this category. WACH (Wachter et al. 2017) is among the first paper to propose a counterfactual explainer and probably the most famous one. The loss function minimized by (Wachter et al. 2017) is defined as $\lambda(b(x') - y')^2 + d(x, x')$ where the first term is the quadratic distance between the desired outcome y' and the classifier prediction on x', and the second term is the distance d between x and x'. λ balances the contribution of the first term against the second term. The distance function d adopted is a crucial characteristic in any counterfactual explainer. In (Wachter et al. 2017) is adopted the Manhattan distance weighted with the inverse median absolute deviation of each feature. CEM, Contrastive Explanations Method (Dhurandhar et al. 2018), is a post-hoc and model-specific explainer tailored for neural networks. In particular, CEM can return *Pertinent Positives (PP)*, which can be seen as prototypes and are the minimal and sufficient factors that have to be present to obtain the output y; and *Pertinent Negatives (PN)*, which are counterfactuals factors, that should be minimally and necessarily absent. Given x, CEM

considers $x_1 = x + \delta$, where δ is a perturbation applied to x. CEM is formulated as an optimization problem over the perturbation variable δ. C-CHVAE (Pawelczyk et al. 2020) is a local model-agnostic post-hoc explainer that accounts for *plausibility* when generating counterfactuals. Indeed, the loss function optimized controls that counterfactuals are not local outliers and that are close to correctly classified observations. Moreover, this method can generate counterfactuals without requiring a distance function for the input space at the cost of using a Variational AutoEncoder. DICE, Diverse Counterfactual Explanations (Mothilal et al. 2020) is a model-agnostic post-hoc explainer that solves an optimization problem with constraints to account for *plausibility* and *diversity* evaluated through distance functions. Plausibility avoids the generation of unfeasible counterfactuals, while diversity provides different ways of changing the outcome class. FACE, Feasible and Actionable Counterfactual Explanations (Poyiadzi et al. 2020) is a model-agnostic post-hoc explainer that focuses on returning actionable counterfactuals, i.e., records *coherent* with the input data distribution. In particular, FACE uncovers "feasible paths" for generating counterfactuals, i.e., the shortest path defined via density-weighted metrics starting from the input instance. Finally, it uses a shortest path algorithm to find all the records that satisfy the requirements. CFX (Albini et al. 2020) is a model-specific post-hoc explainer for Bayesian Network Classifiers. The explanations are built from relations of influence between variables, indicating the reasons for the classification. In particular, this method's main achievement is that it can find pivotal factors for the classification task that, if removed, would lead to a different classification.

Counterfactual-based explanations are gaining attention during the past few years due to their ability to suggest what to do to achieve a different outcome w.r.t. the one predicted by the black-box. There are several characteristics to consider in a counterfactual, such as plausibility, which requires the explanations to be feasible, and actionability, so that the counterfactual can not suggest changing the values of unfeasible variables, such as age or sex. Satisfying these characteristics is of utmost importance because otherwise, the counterfactuals obtained may not be applicable or understandable by the end user. For example, a counterfactual might require changing age or height, factors that cannot be changed, thus making the counterfactual unfeasible. Among the methods proposed, C- CHVAE deals with the plausibility of the counterfactuals proposed, and FACE tackles both the plausibility and the actionability. The majority of the algorithms proposed solves an optimization problem based on a distance function and some perturbation of the original data (CEM, WACH, C- CHVAE) and only a few methods exploit different approaches, such as VAE, as C- CHVAE. Most of the proposed methods are local and post-hoc, with CFX and CEM specifically designed for certain models, while the others are agnostic. Among the methods proposed, CEM is a promising solution since it provides both prototypes and counterfactuals, allowing for an in-depth analysis, such as MMD- CRITIC and LORE.

> Counterfactuals-based explanations allow the users to understand what to do to achieve a different outcome. This kind of reasoning is close to how human reason, hence it is becoming quite popular. To make counterfactuals as realistic as possible, they must meet criteria such as plausibility and actionability.

Table 2 Comparison of the fidelity and the faithfulness metrics of different explanation methods

Dataset	Black-Box	Fidelity				Faithfulness	
		LIME	SHAP	ANCHOR	LORE	LIME	SHAP
adult	LG	0.979	0.613	**0.989**	0.984	0.099 (0.30)	**0.38** (0.37)
	XGB	0.977	0.877	0.978	**0.982**	0.030 (0.32)	**0.36** (0.49)
	CAT	0.96	0.777	0.988	**0.989**	0.077 (0.32)	**0.44** (0.37)
german	LG	**0.984**	0.910	0.730	0.983	**0.23** (0.60)	0.19 (0.63)
	XGB	**0.999**	0.821	0.802	0.982	0.16 (0.26)	**0.44** (0.21)
	CAT	0.979	0.670	0.620	**0.981**	0.34 (0.33)	**0.43** (0.32)

Bold values indicate the best results
For every evaluation, we report the mean and the standard deviation over a subset of 50 test set records

5.5 Tabular data explainers quantitative comparison

We validate explanation methods working on tabular data by considering the two metrics most widespread in the literature, i.e., *fidelity* and *stability*. We remark that we focus our benchmarking on the subset of explanation methods that are most widely adopted/extended in the literature and work according to different ideas to retrieve explanations. In particular, in this section, we focus on LIME, SHAP,[16] ANCHOR and LORE. Another limitation in the benchmarking is that even though these methods are acting on the same data type, the explanations returned are of a different type. Thus, it is not only possible to compare explainers providing different types of explanations. The results of the fidelity are reported in Table 2. The fidelity values are above 90% for all the methods highlighting that the local surrogates are good at mimicking the black-box models. Regarding the feature importance-based models, LIME shows higher values of fidelity w.r.t. SHAP, especially for adult. In particular, SHAP has lower values for CAT (both german and adult), suggesting that it may not be good in explaining this kind of ensemble model. Concerning rule-based models, the fidelity is high for both of them. However, we notice that ANCHOR shows lower values of fidelity for CAT-german, a behavior which is similar to the one of SHAP. Besides fidelity, we compare LIME and SHAP also on faithfulness and monotonicity. Overall, we did not find any model to be monotonic, and hence we do not report any results. On the other hand, the results for the faithfulness are reported in Table 2. For adult, the faithfulness is quite low, especially for LIME. The explainer with the highest faithfulness is SHAP explaining CAT. Regarding german, instead, the values are higher, highlighting a better faithfulness overall. However, also for this dataset, SHAP has a better faithfulness w.r.t. LIME. In Table 3 are reported the results obtained from the analysis on the stability: a high value means that the explainer presents high instability, meaning that we can have quite different explanations for similar inputs. None of the methods is remarkably stable according to this metric. This weakness is widely shared by many explainers independently from the data type and explanation type. Therefore, an important insight from these experiments is to work toward the stabilization of these procedures. Table 4

[16] Since SHAP is not training a local surrogate, we evaluate the fidelity of SHAP by learning a classifier on the sum of the SHAP's values.

Table 3 Comparison on the stability metric

Dataset	Black-box	LIME	SHAP	ANCHOR	LORE
adult	LG	24.37 (2.74)	1.52 (4.49)	22.36 (8.37)	21.76 (11.80)
	XGB	10.16 (6.48)	2.17 (2.18)	26.53 (13.08)	30.01 (20.52)
	CAT	0.35 (0.43)	0.03 (0.01)	6.51 (4.40)	27.80 (70.05)
german	LG	18.87 (0.73)	19.01 (23.44)	101.07 (62.75)	622.12 (256.70)
	XGB	26.08 (14.50)	38.43 (30.66)	121.40 (98.43)	725.81 (337.26)
	CAT	2.49 (9.91)	15.92 (10.71)	123.79 (76.86)	756.70 (348.21)

We report the mean and the standard deviation over a subset of 30 test records

Table 4 Explanation runtime expressed in seconds for explainers of tabular classifiers, the standard deviation is shown in parentheses

Dataset	Black-box	LIME	SHAP	DALEX	ANCHOR	LORE	SKOPERULE
adult	LG	0.1 (0.01)	**0.001** (0.00)	90 (0.09)	2 (0.10)	15 (0.32)	100 (0.32)
	XGB	**0.1** (0.02)	0.2 (0.03)	108 (0.10)	5 (0.11)	50 (0.13)	–
	CAT	**0.2** (0.00)	3 (0.02)	110 (0.12)	3 (0.21)	35 (0.24)	–
german	LG	0.007 (0.00)	**0.0008** (0.00)	0.8 (0.00)	2 (0.17)	2 (0.31)	70 (0.12)
	XGB	0.03 (0.01)	**0.002** (0.00)	2 (0.12)	2 (0.12)	4 (0.32)	–
	CAT	0.03 (0.00)	**0.002** (0.02)	1 (0.20)	2 (0.42)	6 (0.20)	–

Bold values indicate the best results

shows the explanation runtime approximated as order of magnitude. Overall, feature importance explanation algorithms are faster w.r.t. the rule-based ones. In particular, SHAP is the most efficient, followed by LIME. We remark that the computation time of LORE depends on the number of neighbors to generate exploiting a genetic algorithm (in this case, we considered 1000 samples). ANCHOR, instead, requires a minimum precision as well as SKOPERULE (we selected min precision of 0.40).

Overall, there are many explanation methods available for tabular data. As already stated, we focused our analysis and benchmark on the most widely used. We redirect the interested reader to our XAI Living Survey for more methods and details. Depending on the final user, different kinds of explanations are better suited than others. During the past few years, counterfactuals have witnessed great interest due to their logical structure and the possibility of explaining what to do instead of obtaining another prediction. In our opinion, even if rules, prototypes, and counterfactuals seem to be the best solutions w.r.t. human readability, features importance-based explainers are still the most widely used mainly due to the fact that they were the first proposed, with well-maintained libraries, and that are typically faster than the others. Moreover, regarding rules, prototypes, and counterfactuals, there are still several open questions and challenges, such as improving the efficiency and effectiveness as well as considering the constraints of the domain in which the model is being employed.

Table 5 Explainers for black-boxes classifying image data sorted by explanation type: Maps (SM), Concept Attributions (CA), Counterfactuals (CF), and Prototypes (PR)

Type	Name	References	Data type	IN/PH	G/L	A/S
SM	ϵ- LRP	Bach et al. (2015)	ANY	PH	L	S
	LIME	Ribeiro et al. (2016)	ANY	PH	L	A
	SHAP	Lundberg and Lee (2017)	ANY	PH	L	A
	GRAD- CAM	Selvaraju et al. (2020)	IMG	PH	L	S
	DEEPLIFT	Shrikumar et al. (2017)	ANY	PH	L	S
	SMOOTHGRAD	Smilkov et al. (2017)	IMG	PH	L	S
	INTGRAD	Sundararajan et al. (2017)	ANY	PH	L	S
	GRAD- CAM++	Chattopadhay et al. (2018)	IMG	PH	L	S
	RISE	Petsiuk et al. (2018)	IMG	PH	L	S
	ANCHOR	Ribeiro et al. (2018)	ANY	PH	L	A
	EXTREME PERTURBATION	Fong et al. (2019)	IMG	PH	L	S
	XRAI	Kapishnikov et al. (2019)	ANY	PH	L	S
	CXPLAIN	Schwab and Karlen (2019)	IMG	PH	L	S
	EIGEN- CAM	Muhammad and Yeasin (2020)	IMG	PH	L	S
	ABLATION- CAM	Desai and Ramaswamy (2020)	IMG	PH	L	S
	SCORE- CAM	Wang et al. (2020)	IMG	PH	L	S
	OPTI- CAM	Zhang et al. (2023)	IMG	PH	L	S
CA	TCAV	Kim et al. (2018)	IMG	PH	L	A
	ICNN	Shen et al. (2021)	IMG	IN	G	S
	ACE	Ghorbani et al. (2019)	IMG	PH	G	A
	CACE	Goyal et al. (2019)	IMG	IN	G	A
	CONCEPTSHAP	Yeh et al. (2020)	IMG	PH	G	A
	PACE	Kamakshi et al. (2021)	IMG	PH	G	S
	GAN STYLE	Lang et al. (2021)	IMG	PH	G	A

6 Explanations for image data

In this section, we present a selection of approaches for explaining decision systems acting on images. In particular, we distinguish the following types of explanations: *Saliency Maps* (SM, Sect. 6.1, *Concept Attribution* (CA, Sect. 6.2), *Prototypes* (PR, Sect. 6.3) and *Counterfactuals* (CF, Sect. 6.4). Table 5 (organized as the previous one) summarizes and categorizes the explanation methods acting on image data. As for the previous section, after the presentation of the methods, we report the results of experiments for which we considered three datasets:[17] `mnist`, `cifar` in its 10 class flavor and `imagenet`. We selected these datasets because they are widely used as benchmarks in ML in general and also in experimenting with XAI approaches. On these three datasets, we trained the models most used in literature to evaluate the explanation methods: for `mnist` and `cifar`, we trained a CNN with two convolutions and two

[17] `mnist`: http://yann.lecun.com/exdb/mnist/, `cifar`: https://www.cs.toronto.edu/~kriz/cifar.html, and `imagenet`: http://image-net.org/.

Table 5 continued

Type	Name	References	Data type	IN/PH	G/L	A/S
CF	L2X	Chen et al. (2018)	ANY	PH	L	A
	CEM	Dhurandhar et al. (2018)	IMG	PH	L	A
	GUIDED PROTO	Looveren and Klaise (2021)	IMG	PH	L	A
	ABELE	Guidotti et al. (2020a)	IMG	PH	L	A
	PIECE	Kenny and Keane (2021)	IMG	PH	L	S
	SEDC	Vermeire et al. (2022)	IMG	PH	L	A
	ECINN	Hvilshøj et al. (2021)	IMG	PH	L	A
PR	MMD- CRITIC	Kim et al. (2016)	ANY	IN	G	A
	INFLUENCE FUNCTIONS	Koh and Liang (2017)	ANY	PH	L	A
	PROTOPNET	Chen et al. (2019)	IMG	IN	G	S
	PROTOTREE	Nauta et al. (2021)	IMG	IN	G	S
	DEFORMABLE PROTOPNET	Donnelly et al. (2022)	IMG	IN	G	S

For every method is indicated if it is possible to use it for images (IMG) only or for ANY type of data, if it is an Intrinsic (IN) or a Post-Hoc (PH) model, Local (L) or Global (G), and if it is model-Agnostic (A) or model-Specific (S)

linear layers, while for `imagenet`, we decided to use the VGG16 network (Simonyan and Zisserman 2015).

6.1 Saliency maps

A *Saliency Map (SM)* is an image in which a pixel's brightness represents how salient the pixel is. Formally, a SM is modeled as a matrix S, which dimensions are the sizes of the image for which we want to explain a decision, and the values s_{ij} are the saliency values of the pixels ij. The greater the value of s_{ij}, the bigger the saliency of that pixel. To visualize a SM, divergent color maps are typically used with colors ranging, for instance, from red to blue. A positive value (red) means that the pixel ij has contributed positively to the classification, while a negative one (blue) means that it has contributed negatively. Examples of SM are reported in Fig. 7. There are two big families of methods for creating SMs. The first one assigns to every pixel a saliency value. The second one segments the image into different pixel groups using a segmentation method, and then it assigns a saliency value for each segmented group. We underline that, from a pure interpretability perspective, SMs are the correspondent of FI for images. Indeed, as shown in the following, data-agnostic approaches that can be used for extracting FI explanations on tabular data can also be used for extracting SM on images.

LIME is a local post-hoc model-agnostic explainer (already presented in Sect. 5) that can also be used to retrieve SM for image classifiers. In particular, LIME returns segmentation-based SM where each segment is called *superpixel*. After the input image segmentation, LIME adopts a one-hot vector representation of the image where the input image is a vector composed of m ones if m is the number of segments identified. Then, it creates the synthetic neighborhood by randomly substituting the

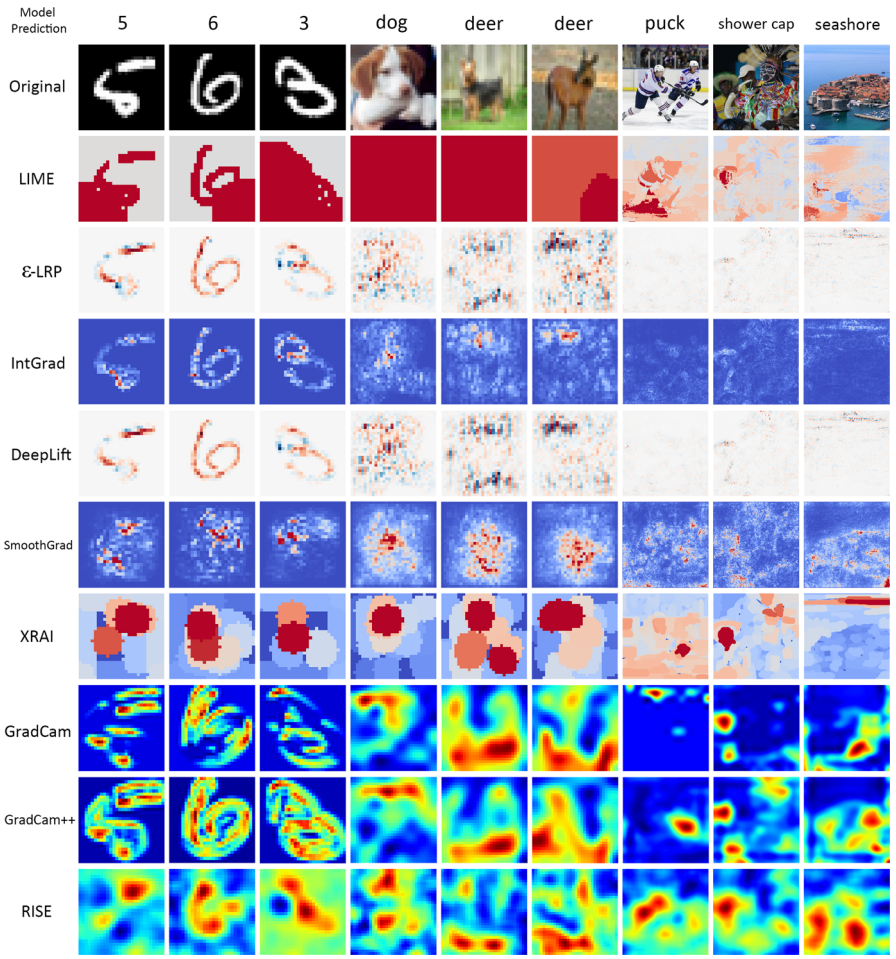

Fig. 7 Examples of SMs obtained with the explainers presented in Sect. 6.1 on `mnist`, `cifar` and `imagenet` datasets. The first row reports the images of the datasets, while the header shows the class predicted class from the black-box. The remaining rows report the SMs obtained

superpixels with a uniform, possibly neutral, color and by also storing the one-hot representations. The neighborhood of synthetic images is then fed into the black-box, and the one-hot representation of the neighborhood, together with the black-box prediction, is used to train a sparse linear model. Finally, the coefficients of the linear model are used as the importance of the superpixels. Examples of the explanations returned by LIME are shown in the second row of Fig. 7. A critical aspect for obtaining a good explanation with this approach is the choice of the segmentation algorithm and its hyper-parameters. Indeed, for small-resolution images, the segmentation in LIME does not work out of the box, resulting in the algorithm selecting all the images as a superpixel. Recently, many research improved and extended LIME for image

classifiers[18] (Shi et al. 2020; Peltola 2018; Zafar and Khan 2019; Bramhall et al. 2020).

LRP, Layer-wise Relevance Propagation (Bach et al. 2015), commonly known as ϵ- LRP, is a local post-hoc model-specific explainer that is designed for images but can be practically applied for any data type. ϵ- LRP was introduced for feed-forward neural networks (Arras et al. 2017) and then adapted to different type of models. ϵ- LRP decomposes the prediction y backward using local redistribution rules until it assigns a relevance score R_i to each pixel value. The simple ϵ- LRP rule redistributes relevance from layer $l+1$ to layer l is: $R_i = \sum_j \frac{a_i w_{ij}}{\sum_i a_i w_{ij} + \epsilon} R_j$ where a_i is the neuron activations at layer l, R_j the relevance scores associated to the neurons at layer $l+1$, w_{ij} the weight connecting neuron i to neuron j, and ϵ is added to prevent division by zero. Intuitively, this rule redistributes relevance proportionally from layer $l+1$ to each neuron in l based on the connection weights. The final explanation is the relevance of the input layer. Figure 7 shows examples of ϵ- LRP SM in the third row. As with all the pixel-wise explanation methods, ϵ- LRP works very well on datasets with low-resolution images like `mnist`, while the explanations returned for more complex images such as those of `cifar` and `imagenet` are quite unclear and only limitedly interpretable. In the literature, are introduced other variations of the LRP algorithm. γ-LRP favors the effect of positive contributions over negative contributions by separating the weights w_{ij} into $w_{ij}^- + w_{ij}^+$ and adding a multiplier to the positive ones: $w_{ij}^- + \gamma w_{ij}^+$ Another variant of ϵ- LRP is SPRAY (Lapuschkin et al. 2019), which builds a spectral clustering on top of the local instance-based ϵ- LRP explanations. Similarly to (Li et al. 2019), it starts with the ϵ- LRP of the input instance and finds the LRP attribution relevance for a single input of interest x.

INTGRAD, Integrated Gradient (Sundararajan et al. 2017), similarly to ϵ- LRP, is a local post-hoc model-specific data-agnostic explainer designed for images. INTGRAD utilizes the gradients of a black-box providing access to the gradient, such as a neural network, along with sensitivity techniques. For this reason, it can be applied only to differentiable models. Formally, given b and x, and let x' be the baseline input,[19] INTGRAD constructs a path from x' to x and computes the gradients of points along the path. For images, the points are taken by overlapping x on x' and gradually modifying the opacity of x. Integrated gradients are obtained by cumulating the gradients of these points. Formally, the integrated gradient along the i^{th} dimension for an input x and baseline x' is defined as follows. Here, $\partial b(x)/\partial x_i$ is the gradient of $b(x)$ along the i^{th} dimension. The equation for computing the scores is: $e_i(x) = (x_i - x_i') \int_{\alpha=0}^{1} \frac{\partial b(x' + \alpha(x - x'))}{\partial x_i} \, d\alpha$. Examples of SM produced by INTGRAD are in Fig. 7. They tend to have more uniform pixels with a similar saliency than ϵ- LRP. As in ϵ- LRP, INTGRAD highlighted the background when explaining the "deer" image. However, an arbitrary choice of baselines could cause issues. For example, a black baseline image could cause INTGRAD to lower the importance of black pixels in the source image. This problem is due to the difference between the image's pixel and the baseline $(x_i - x_i')$ present in the integral equation. *Expected Gradients* (Erion et al. 2019) tries to overcome this problem by

[18] DLIME: https://github.com/rehmanzafar/dlime_experiments.

[19] The baseline x' is generally chosen as a zero matrix or vector. For example, for the image domain, the baseline is generally a black or a white image.

averaging INTGRAD to different baselines. A different variation of INTGRAD is called *Adversarial Gradient Integration (AGI)*. An adversarial example of an image is the image with a different class, most similar to the original one. *AGI* (Pan et al. 2021) integrates the gradients from adversarial examples of different classes to the target example along the curve of the steepest ascent. The contributions from the different examples are then summed up to obtain the final SM. By not relying on the choice of references, it is possible to reduce the sparsity of INTGRAD.

DEEPLIFT (Shrikumar et al. 2017), is local post-hoc model-specific explainer for differentiable models. It computes SMs in a backward fashion, similarly to ϵ- LRP, but it uses a baseline reference like INTGRAD. DEEPLIFT exploits the slope, instead of the gradients, which describes how the output $y = b(x)$ changes as the input x differs from a baseline x'. Like ϵ- LRP, an attribution value r is assigned to each unit i of the neural network going backward from the output y. This attribution represents the relative effect of the unit activated at the original network input x compared to the activation at the baseline reference x'. DEEPLIFT computes the starting values of the last layer L by the difference between the output of the input and baseline. Then, it uses the following recursive equation to compute the attribution values of layer l using the attributions of layer $l+1$ to obtain the values of the starting layer: $r_i^{(l)} = \sum_j \frac{a_{ji} - a'_{ji}}{\sum_i a_{ji} - \sum_i a'_{ji}} r_j^{(l+1)}, a_{ji} = w_{ji}^{(l+1,l)} x_i^{(l)}, a'_{ji} = w_{ji}^{(l+1,l)} x_i^{'(l)}$ where $w_{ij}^{l+1,l}$ are the weights of the network between the layer l and the layer $l+1$, and a are the activation values. As for INTGRAD, picking a baseline is not trivial and might require domain experts. From Fig. 7, we notice that the SMs obtained with DEEPLIFT are very similar to those obtained with ϵ- LRP.

SMOOTHGRAD (Smilkov et al. 2017) is a local post-hoc model-specific explanation method. Since SMs tend to be noisy, especially for pixel-wise SMs, SMOOTHGRAD tries to overcome produce less noisy SM through *smoothing*. Usually, a SM is created directly on the gradient of the black-box output w.r.t. the input $\partial y / \partial x$. SMOOTHGRAD augments this process by smoothing the gradients with a Gaussian noise kernel. More in detail, it takes the image x, applies Gaussian noise to it, and retrieves the SM for every perturbed image, using the gradient. The final SM is an average of these perturbed SMs. Formally, given a saliency method $f(x)$ which produces a SM s, its smoothed version $\hat{f}$ can be expressed as: $\hat{f} = \frac{1}{n} \sum_1^n f(x + \mathcal{N}(0, \sigma^2))$ where n is the number of samples, and $\mathcal{N}(0, \sigma^2)$ is the Gaussian noise. In (Adebayo et al. 2018, 2020), some weaknesses of SMOOTHGRAD are shown: people tend to evaluate SMs on what they are expected to see. For example, in a bird image, we want to see the shape of a bird. However, this does not mean that this is what the network is looking at. Figure 8 highlights this problem. We obtained the SMs taking the gradient of the output w.r.t. the input, and then we used SMOOTHGRAD. We observe that the SMs completely changed their behavior, moving in the direction of the subject.

SHAP provides two versions that can be employed for deep networks tailored for image classification: DEEP- SHAP and GRAD- SHAP. Therefore, even though SHAP is a local post-hoc model-agnostic explainer, the DEEP- SHAP and GRAD- SHAP versions are model-specific implementations. DEEP- SHAP is a high-speed approximation algorithm for SHAP values in deep learning models that builds on a connection with DEEPLIFT. The implementation is different from the original DEEPLIFT by using as baseline a dis-

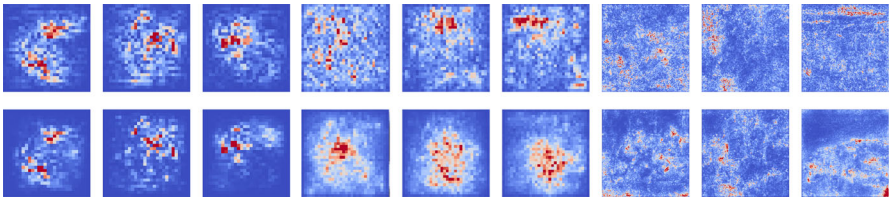

Fig. 8 Visual Comparison of SMs obtained by taking the gradient of the output y w.r.t. the input image x (center) and SMOOTHGRAD (bottom). In the three images in the center, the SM changes drastically. In all three cases, it focuses on the subject of the image, completely changing the original values. This is also true for the seashore image on the far right

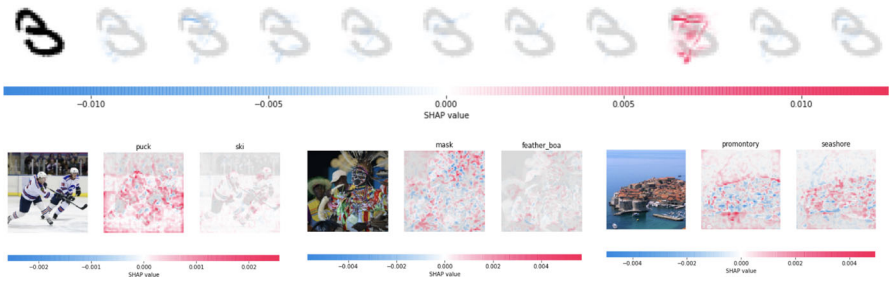

Fig. 9 Explanations of DEEP- SHAP on `mnist` (top), GRAD- SHAP on `imagenet` (bottom)

tribution of background samples instead of a single value and using Shapley equations to linearise non-linear components of the black-box such as max, softmax, products, divisions, etc. GRAD- SHAP, instead, is based on INTGRAD and SMOOTHGRAD (Sundararajan et al. 2017; Smilkov et al. 2017). INTGRAD values are a bit different from SHAP values and require a single reference value to integrate. As an adaptation to make them approximate SHAP values, GRAD- SHAP reformulates the integral as an expectation and combines that expectation with sampling reference values from the background dataset as in SMOOTHGRAD. Examples of explanations of DEEP- SHAP and GRAD- SHAP are shown in Fig. 9. GRAD- SHAP produces a single SM, while DEEP- SHAP produces a SM for each class in the input image. Even if more expensive to compute, having a SM for each class could be useful to compare the importance of the predicted class with different classes.

XRAI (Kapishnikov et al. 2019) is based on INTGRAD and inherits its properties. Differently from INTGRAD, XRAI first over-segments the image. It iteratively tests each region's importance, fusing smaller regions into larger segments based on attribution scores. XRAI follows three steps: *segmentation*, *get attribution*, and *selecting regions*. The segmentation is repeated several times with different segments to reduce the dependency on image segmentation. For attribution, XRAI uses INTGRAD with black and white baselines averaged. Finally, to select regions, XRAI leverages the fact that, given two regions, the one that sums to the more positive value should be more important to the black-box. Thus, XRAI starts with an empty mask and then selectively adds the regions that yield the maximum gain in the total attributions per area. From Fig. 7, we notice how the SMs obtained by XRAI are very different from the others already

presented. Like all segmentation methods, XRAI performs best when high-resolution images are available. However, because the segmentation is performed on the values obtained from INTGRAD and not on the raw image, good results are obtained even with low-resolution images.

GRAD-CAM (Selvaraju et al. 2020) is a local post-hoc model-specific explainer for image data. It uses the gradient information flowing into the last convolutional layer of a CNN to assign saliency values to each neuron for a particular decision. Convolutional layers naturally retain spatial information in fully-connected layers, so it assumes that the last convolutional layers have the best compromise between high-level semantics and detailed spatial information. To create the SM, GRAD- CAM takes the feature maps created at the last layer of the convolutional network a. Then, it computes the gradient of an output of a particular class y^c for every feature map activations k, i.e., $\partial y^c / \partial a^k$. This equation returns a tensor of dimensions $[k, v, u]$ where k is the number of features maps and u and v are the height and the width of the image. GRAD- CAM compute the saliency value for every feature map by pooling the dimensions of the image. The final heatmap is calculated as a weighted sum of these values. This results in a coarse heatmap of the same size as the convolutional feature maps. An up-sampling technique is applied to the final result to produce a map of the initial image dimension. From Fig. 7 is clear that this coarse grain heatmap style is very characteristic of GRAD- CAM. These heat maps highlight very different parts of the image compared to other methods.

GRAD-CAM++ (Chattopadhay et al. 2018) extends GRAD- CAM solving the following issue. The spatial footprint in an image is essential for GRAD- CAM's visualizations to be robust. Hence, if multiple objects have slightly different orientations or views, different feature maps may be activated with differing spatial footprints. The one with lesser footprints fades away in the final sum. GRAD- CAM++ fix this problem by taking a weighted average of the pixel-wise gradients. In particular, GRAD- CAM++ reformulates GRAD- CAM by explicitly coding the structure of the weights α_k^c as: $\alpha_k^c = \sum_i \sum_j w_{ij}^{kc} \cdot \text{ReLU}\left(\partial y^c / \partial a_{ij}^k\right)$ where *ReLU* is the Rectified Linear Unit activation function, and w_{ij}^{kc} are the weighting co-efficients for the pixel-wise gradients for class c and convolutional feature map a^k. The idea is that w_k^c captures the importance of a particular activation map a^k, and positive gradients are preferred to indicate visual features that increase the output neuron's activation rather than those that suppress it. Besides GRAD- CAM++, there are many variations of pixel masking methods based on GRAD- CAM. *Score-CAM* (Wang et al. 2020) gets rid of the dependence on gradients by obtaining the weight of each activation map through its forward passing score on the target class, the final result is obtained by a linear combination of weights and activation maps. *Eigen-CAM* (Muhammad and Yeasin 2020) extracts the importance scores by projecting the last convolutional layer into the first eigenvector extracted from the same layer. The scores produced appear more uniform and less sparse. *Ablation-CAM* (Desai and Ramaswamy 2020) uses ablation analysis, i.e., setting activation value to zero, to determine the importance of individual feature map units with respect to class. *Extreme Perturbation* (Fong et al. 2019) are regions of an image that, for a given area, maximally affect the activation of a class in the last layer of a neural network. As the perturbation area increases, the explainer reveals more

of the image in order of decreasing importance. *CXPlain* (Schwab and Karlen 2019) frames the task of providing explanations as a causal learning task. It trains a causal explainer that learns to estimate to what degree certain inputs cause outputs in another model.

RISE (Petsiuk et al. 2018) is a local post-hoc model-agnostic explainer for image data. To produce a SM for an image x, RISE generate N random mask $M_i \in [0, 1]$ from Gaussian noise. The input x is element-wise multiplied with these masks M_i, and the result is fed to the base model. The SM is obtained as a linear combination of the masks M_i with the predictions from the black-box corresponding to the respective masked inputs. The intuition behind this is that $b(x \odot M_i)$ is high when pixels preserved by mask M_i are essential.

Saliency maps-based explainers comparison Saliency Maps (SM) are the most diffused explanations for images. The literature presents a multitude of explainers that are capable of producing such a type of explanation. As seen in Fig. 7, LIME can lead to useless SM for `mnist` and `cifar` as it segments the images in superpixels big as the whole image in some cases. On the other hand, those produced by XRAI are much more clear. LIME computes the segmentation at the very beginning of the algorithm on the raw images. Thus, for low-resolution images, segmentation algorithms are more difficult to calibrate. XRAI instead firstly compute INTGRAD values and then agglomerate them using segmentation. This result seems to be much clearer, even with very small images. In general, we observe that segmentation methods work best for high-resolution images where the concept of the image can be easily separated. For instance, in the SM of the "seashore" image produced by XRAI is very clear how the method selected three parts of the image: the horizon, the sea, and the promontory. Since pixel wise-methods produce SMs in terms of single pixels, which are low-level features, they are useful only for an expert user who wants to check the robustness of the black-box. Overall, we deduce that SMs returned by the segmentation methods are more human-friendly than the ones returned by pixel-wise methods.

Pixel based explainers like DEEP- SHAP, GRAD- SHAP, DEEPLIFT, INTGRAD, and ϵ- LRP, typically return very similar results. DEEPLIFT and ϵ- LRP return similar results, probably due to their similarity in the computation of the SM values. However, DEEPLIFT is model-agnostic, while ϵ- LRP is model-specific, and it needs adjustments for nonstandard neural networks. In conclusion, DEEPLIFT result to be the best competitor. GRAD- CAM, GRAD- CAM++, and RISE return SMs that could resemble those returned by segmentation-based explainers. This is due to the fact that scores of these explainers are computed on smaller layers with lower resolution and then interpolated on the original data resolution. GRAD- CAM++ produces better images than GRAD-CAM due to the usage of the second order derivative. Another problem with SMs is the confirmation bias, i.e., a user can hardly realize if a SM is a good explanation or only shows what the user wants to see (Adebayo et al. 2018). SMOOTHGRAD and most guided methods that use the target label to alter the SM suffer from this bias, moving the salient values from the background to the subject.

Analyzing the images by dataset: in `cifar` we notice that all the methods highlight the background of the images in particular in the "deer" class. This result is a problem in the learning phase of the black-box and should not be referred to as problems of the explainers. On the other hand, for `imagenet`, we observe very different SMs.

For instance, for the ice hockey image in Fig. 7, the class in the dataset is "puck", i.e., the hockey disk. LIME highlights the ice as important, while XRAI and GRAD- CAM++ highlight the stick of the player, GRAD- CAM highlights the fans while RISE the hockey player. Thus, for the same image, we can obtain very different explanations, further highlighting the fragility of the SMs. Regarding the second image of `imagenet` (the second from the right), we can observe that all the methods capture the same pattern. A straw hat in the background triggered the class "shower cap" while the correct one was "mask". Finally, in the "seashore" of `imagenet`, we have an island in the sea. The top three predicted classes are seashore (0.91), promontory (0.04), and cliff (0.01). Half of the tested methods like LIME, SMOOTHGRAD, RISE, and GRAD- CAM were fooled that the promontory is important to the class "seashore". We can conclude that SMs are very fragile when we have multiple classes in the image, even if these classes have a very low predicted probability.

Segmentation methods are more human-understandable than pixel-wise methods. Guided propagation methods can hardly be trusted due to confirmation bias, and therefore it is better not to adopt them.

6.2 Concept attribution

Most ML models are designed to operate on low-level features like edges and lines in a picture that does not correspond to high-level concepts that humans can easily understand. In (Adebayo et al. 2018; Yang and Kim 2019), the authors pointed out that feature-based explanations applied to state-of-the-art black-box models can yield non-sensible explanations. For example, we can consider SM as low-level explanations for images, as they assign to every pixel a saliency value. Although it is possible to look at every pixel and infer their numerical values, these make little to no sense to humans: we do not say that the 5th pixel of an image has a value of 28. On the other hand, *Concept Attribution (CA)* methods quantify, for instance, how much the concepts "stripes" has contributed to the class prediction of "zebra". Indeed, CA-based explanation methods construct the explanation based on human-defined concepts rather than representing the inputs based on features and internal model (activation) states. Hence, this idea of high-level features might be more familiar to humans, that can be more likely to accept it (Hartmann et al. 2022; Renard et al. 2019). Formally, given a set of images belonging to a concept $[x^{(1)}, x^{(2)}, \ldots, x^{(i)}]$ with $x^{(i)} \in C$, CA methods can be thought as a function $f : (b, [x^{(i)}]) \rightarrow e$ which assign a score e to the concept C basing on the predictions and the values of the black-box b on the set $[x^{(i)}]$.

TCAV, Testing with Concept Activation Vectors (Kim et al. 2018) is a global post-hoc model-agnostic explainer for image classifiers that provides a quantitative explanation of how important a concept is for the prediction. In TCAV, every concept is represented by a particular vector called *Concept Activation Vectors (CAVs)* created by interpreting an internal state of a neural network in terms of human-friendly concepts. TCAV uses directional derivatives to quantify the degree to which a user-defined concept is *vital* to a classification result. For example, how sensitive a prediction of "zebra" is to the presence of "stripes". TCAV requires two main ingredients: (i) concept-containing inputs and negative samples, i.e., random inputs, and (ii) pre-trained ML models on

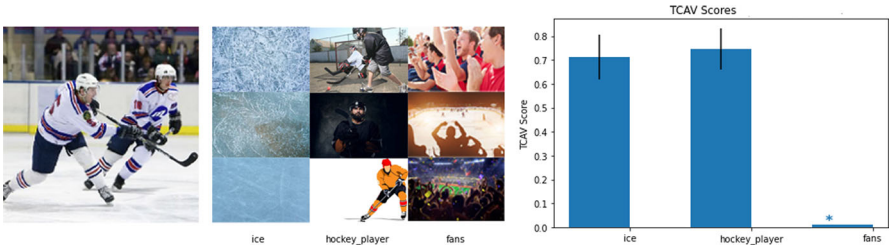

Fig. 10 TCAV scores for three concepts: *ice, Hockey player, and cheering people (fans)* for the class *puck* of `imagenet`. On the left is the query image; on the center, some samples of the image tested in TCAV as concepts, and on the right is the histogram of the scores with errors. The *hockey players* has been classified as a *puck*, but the SMs are very different alongside methods. In the histogram on the left, it is possible to see that the *ice* and the *hockey players* are important concepts, while the background fans are not significant

which the concepts are tested. Inputs containing concepts and random inputs are fed into the black-box to obtain predictions that will be used by TCAV to test the ability of the machine learning model to capture a particular concept. Then, a linear classifier is trained to distinguish the activation of the network due to concept-containing vs. random inputs. The result of this training is *concept activation vectors (CAVs)*. Once CAVs are defined, the directional derivative of the class probability along CAVs can be computed for each instance that belongs to a class. The "concept importance" for a class is computed as a fraction of the class instances that get positively activated by the concept containing inputs vs. random inputs. In Fig. 10, we report an example of TCAV explanation. We remark that to obtain this explanation, the user must collect concepts like "ice", "hockey player" and "fans". Then TCAV computes the scores unveiling which one has more impact on the prediction.

Several extensions of TCAV are present in the literature. ACE, Automated Concept-based Explanation (Ghorbani et al. 2019), is the evolution of TCAV, and it does not require concepts in input as it can automatically discover them. A set of images from the same class is segmented with multiple resolutions resulting in a pool of segments all coming from the same class. Then the activation space of one bottleneck layer of a CNN classifier is used as a similarity space. After that, similar segments are clustered in the activation space, and, for each concept, its TCAV importance score is computed given its examples segments. ConceptSHAP (Yeh et al. 2020) is an evolution of SHAP which tries to define an importance score for each concept discovered. CONCEPTSHAP finds the importance of each individual concept from a set of m concept vectors $C_s = \{c_1, c_2, \ldots, c_m\}$ by utilizing Shapley values. Similar to ACE, CONCEPTSHAP aims at having concepts clustered to certain coherent spatial regions. CaCE, Causal Concept Effect (Goyal et al. 2019) is another variation of TCAV that looks at the causal effect of the presence or absence of high-level concepts on the black-box prediction. Indeed, TCAV can suffer from confounding concepts that could happen if the training data instances have multiple classes, even with a low correlation. PACE (Kamakshi et al. 2021) is a variation of ACE that introduce the use of the black-box for identifying concepts by automatically extracting small sub-regions of the image, called concepts, relevant to the black-box prediction. In (Shen et al. 2021) is presented ICNN, an interpretable method to modify traditional CNNs into *interpretable CNN*. In an inter-

pretable CNN, each filter in a high convolutional layer represents a specific object part. An interpretable CNNs use the same training data as ordinary CNNs without additional annotations of object parts or textures for supervision. The interpretable CNN automatically assigns each filter in a high convolutional layer with an object part during the learning process. The explicit knowledge of interpretable CNN can help people understand their logic, i.e., what patterns are memorized by the CNN for prediction.

Concept attribution-based explainers comparison SMs are the best known and most widely used type of explanation. However, they are very fragile and suffer from interpretation problems as their usage from a cognitive perspective is unclear. On the other hand, the usage of concepts is a new approach that seeks to overcome SMs problems by producing high-level explanations that are more understandable by the end user. Concept-based explanations are a very recent type of explanation for images, and they have potential improvements. It is the first step in the direction of human-like explanations. Human-friendly concepts make it possible to build straightforward and valuable explanations. Humans still need to map images to concepts, but it is a small price to pay to augment human–machine interaction. Generally, concept attribution explainers compute a score that evaluates the probability that a selected concept has influenced the prediction. The main problem, for now, is that this concept has to be manually selected. TCAV is the most advanced algorithm. However, the concepts need to be provided as a set of images reflecting the selected concept, e.g., a set of ice images, a set of people, and a set of stripes. Several variations of TCAV have been introduced to automate the concept selection phase. PACE is the most advanced search when looking for concepts concerning black-box behavior. Researchers are focusing on finding concepts in an automated way, but for the time being, there is no explainer to ensure finding a concept that is humanly understandable.

There is a need to build an explanation in terms of higher features called concepts for a general audience.

6.3 Prototype-based explanations

Another valid explanation type for images is a set of prototypical images that represent a particular class. Human reasoning is often prototype-based, using representative examples as a basis for categorization and decision-making. Similarly, prototype explanation models use representative examples to explain and cluster data. MMD-CRITIC (Kim et al. 2016), already presented in Sect. 5, is an interpretable approach that can be applied to retrieve image prototypes and criticisms. In Fig. 11 is presented an application of MMD- CRITIC on `cifar`. We can extract some interesting knowledge from these methods. For example, from the prototype set, we can deduce that birds usually stand on a tree or fly in the sky, while, in the criticism images, we see that planes are all on a white background or have a different shape from the usual one used for passengers. Influence Functions (Koh and Liang 2017) is a global post-hoc model-agnostic explainer that tries to find the images most responsible for a given prediction through influence functions. The usage of influence functions is a technique from robust statistics to trace a model's prediction through the learning algorithm and

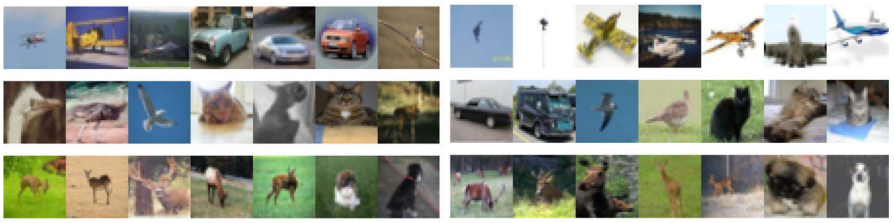

Fig. 11 Prototypes (left) and criticisms (right) returned by MMD- CRITIC on `cifar`

back to its training data, thereby identifying training points most responsible for a given prediction. Visualizing the training points most responsible for a prediction can be useful for more in-depth insights into the black-box behavior. PROTOPNET (Chen et al. 2019) is a global interpretable model for image data that aims at identifying prototypical parts of images (named prototypes) and using them to implement an interpretable classification process. A special deep learning architecture is designed to retrieve these prototypes. The network learns a limited number of prototypical parts from the training set and then identifies parts on the test image that look like the prototypical parts. Then, it predicts based on a weighted combination of the similarity scores between parts of the image and the learned prototypes.

Prototype-based explainers comparison Prototype explanations are not widespread as explanations for image classifiers. The explainers presented have very different approaches: MMD- CRITIC use the Maximum Mean Discrepancy measure to select the prototype set, PROTOPNET instead used part of the image as prototypes. MMD- CRITIC is faster than PROTOPNET but the explanations provided are more obscure.

> Explanation prototypes are not very common for images because the usefulness of such explanations is not clear.

6.4 Counterfactual-based explanations

In parallel with prototypes, also counterfactuals are another widely adopted form of explanation for images. In this setting, counterfactual explainers for images produce images similar to the input image but with a different prediction from the black-box under analysis. Therefore, also in this case, counterfactual methods are post-hoc explainers. Similarly to SM, we can distinguish between counterfactual explainers altering single pixels against those altering the whole image or part of it. CEM, Contrastive Explanation Method (CEM) (Dhurandhar et al. 2018), already presented in Sect. 5, is a local post-hoc model-specific counterfactual explainer that can also be applied on image data. In this setting, Pertinent Positives (PP) or Pertinent Negatives (PN) are the pixels that lead to the same or a different class w.r.t. the original instance. To create PP's and PN's, feature-wise perturbation is done by keeping the perturbations sparse and close to the original instance through an objective function that contains an elastic net $\beta L_1 + L_2$ regularizer and also consider an auto-encoder trained to reconstruct images of the training set. As a result, the perturbed instance lies close to the training data manifold. In Fig. 12, we can see that very few pixels are obtained

Fig. 12 Explanation on `mnist`. **a** CEM: input on the center PN left and PP right. **b** GUIDEDPROTO: left to right, input, the closest counterfactuals labeled as 6, and 8. **c** ABELE: on the left the input query and on the right the counterfactual with 8 as target class

as the counterfactual explanation on `mnist`. However, from a human perspective, this approach might seem much more adversarial than useful for an explanation (Guidotti 2022). An extension of CEM to resolve this problem is presented in (Luss et al. 2021), where the authors leverage the usage of latent features created by a generative model to produce more trustful perturbations. L2X (Chen et al. 2018) is a local post-hoc model-agnostic explanation method that searches for the minimal number of pixels that change the classification. It is based on learning a function for extracting a subset of the most informative features for each given sample using Mutual Information. L2X adopts a variational approximation to efficiently compute the Mutual Information and gives a value for a group of pixels called *patches*. If the value is positive, a group contributed positively to the prediction. Otherwise, it contributed negatively. Guided Prototypes, Interpretable Counterfactual Explanations Guided by Prototypes (GUIDEDPROTO) (Looveren and Klaise 2021) proposes a local post-hoc model-specific explainer that perturbs the input image by using a loss function $\mathcal{L} = cL_{pred} + \beta L_1 + L_2$ optimized using gradient descent. The first term, cL_{pred}, encourages the perturbed instance to predict another class then x, while the others are regularisation terms. In Fig. 12, we show the application of GUIDEDPROTO on `mnist`. It is interesting to notice that the counterfactuals unveil how easy it is to change the class with very few pixels. However, this kind of explanation is not easily human understandable because the few pixels modified can barely be noticed by human eyes. ABELE, Adversarial black-box Explainer generating Latent Exemplars) (Guidotti et al. 2019b), is a local post-hoc model-agnostic explainer that produces explanations composed of: (i) a set of exemplars and counter-exemplar images, i.e., prototypes and counterfactuals, and (ii) a SM. ABELE exploits an adversarial autoencoder (AAE) to generate the synthetic images used in the neighborhood to train the surrogate model used to explain x. Indeed, it builds a latent local decision tree that mimics the behavior of b and selects prototypes and counterfactuals from the synthetic neighborhood exploiting the tree. Finally, the SM is obtained by a pixel-by-pixel difference between x and the exemplars. In Fig. 12, we report an example of the application of ABELE on `mnist`.

Counterfactual-based explainers comparison The goal of counterfactual explainers is to produce examples similar to the input but with a different predicted class. They are more intuitive than the other types of explanations because counterfactual thinking is typical from the human cognitive point of view. There is a wide offer of counterfactual explainers (Guidotti 2022), and it is difficult to define which counterfactual explainer is better than the others for the image domain. If the priority is the execution time, then CEM is the fastest method. Methods such as ABELE and GUIDED PROTOTYPES create more realistic counterfactuals but require training additional models such as autoencoders which are notoriously difficult to train (Doersch 2016).

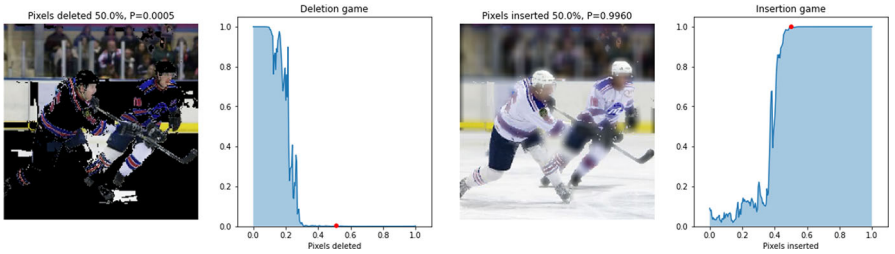

Fig. 13 Example of insertion (left) and deletion (right) scores performed for the SM returned by LIME. The area under the curve is 0.215 for deletion and 0.594 for insertion

Counterfactual explanations are more user-friendly than prototypes and other forms of explanations because they highlight the changes to make to obtain the desired prediction.

6.5 Image explainers quantitative comparison

To quantitatively investigate the performance of the SM explainers analyzed, we computed the *deletion* and the *insertion* metric, presented in Sect. 4. For the computation of the *deletion* metric, we substitute with black pixels the original pixels in increasing order of importance w.r.t. the scores given by the SM. On the other hand, for the *insertion* metric, we blur the whole image with a Gaussian Kernel, and then we slowly insert highly salient pixels w.r.t the SM. For every substitution, we query the black-box with the image, and we measure the performance. The final score is obtained by taking the area under the curve (AUC) (Hand and Till 2001) as a function of the percentage of pixels removed/inserted. In Fig. 13, we show an example of these metrics computed on an image of `imagenet`. Table 6 reports the average results of the calculation of these metrics for a set of 100 randomly selected images for every dataset. Independently from the explainer adopted, we notice that insertion scores decrease while augmenting the dataset dimensions because we have higher information and more pixels have to be inserted to increase the performance. On the other hand, the deletion scores decrease. This might be tied to the fact that since we have more information, it is easier to decrease the performance. The best explainers are highlighted in bold. We notice that RISE is the best approach overall, followed by INTGRAD, DEEPLIFT, and ϵ-LRP. We notice that all these approaches are pixel-wise-based methods. Thus it seems that this kind of evaluation can be advantageous for these explainers. On the contrary, the segmentation-based explainers LIME and XRAI struggle in general and even more when handling low-resolution images.

Table 7 shows the runtime for the image explainers experimented with. For `imagenet`, we only tested SMs methods due to the increasing computational costs of the other explainers. On the contrary, we tested TCAV only on `imagenet` as TCAV needs to obtain different images representing different concepts, and this is difficult to do for very simple images like the one in `mnist` and `cifar`. From the results, we notice that GRAD- CAM and GRAD- CAM++ are the fastest methods, especially for

Table 6 Insertion (top) and deletion (bottom) metrics expressed as AUC of accuracy versus percentage of removed/inserted pixels

	mnist	cifar	imagenet
LIME	0.807 (0.14)	0.41 (0.21)	0.34 (0.25)
ϵ- LRP	0.976 (0.02)	0.56 (0.20)	0.28 (0.19)
INTGRAD	0.975 (0.03)	**0.64** (0.22)	0.37 (0.23)
DEEPLIFT	0.976 (0.02)	0.57 (0.20)	0.28 (0.19)
SMOOTHGRAD	0.959 (0.03)	0.55 (0.23)	0.34 (0.26)
XRAI	0.956 (0.04)	0.58 (0.21)	0.40 (0.26)
GRAD- CAM	0.941 (0.04)	0.57 (0.20)	0.21 (0.19)
GRAD- CAM++	0.941 (0.04)	0.52 (0.22)	0.32 (0.26)
RISE	**0.978** (0.03)	0.61 (0.21)	**0.50** (0.26)
LIME	0.388 (0.21)	0.221 (0.19)	0.051 (0.05)
ϵ- LRP	**0.120** (0.01)	0.127 (0.11)	**0.014** (0.02)
INTGRAD	0.128 (0.01)	**0.118** (0.07)	0.019 (0.04)
DEEPLIFT	**0.120** (0.01)	0.127 (0.11)	**0.014** (0.02)
SMOOTHGRAD	0.135 (0.04)	0.153 (0.13)	0.033 (0.05)
XRAI	0.151 (0.04)	0.144 (0.07)	0.086 (0.11)
GRAD- CAM	0.297 (0.20)	0.153 (0.12)	0.139 (0.12)
GRAD- CAM++	0.252 (0.13)	0.283 (0.24)	0.081 (0.10)
RISE	**0.120** (0.01)	0.124 (0.07)	0.044 (0.05)

The reported value represents the mean of the scores obtained on a subset of 100 instances of the dataset, and the value on the parenthesis is the standard deviation

complex models like the VGG network. In general, SM pixel-wise explanations are faster to achieve because segmentation slows down a lot, especially for high-resolution images. CA, CF, and PR methods are very slow compared to SM. This problem happens because these methods require additional training or use some search algorithm to return their explanations. CA, CF, and PR methods produce more useful explanations, but since SMs are easier and faster to obtain, they are seen more applied in the literature than the other methods making them more widespread.

7 Explanations for text data

For text data, we can distinguish the following types of explanations: *Sentence Highlighting (SH)*, described in Sect. 7.1, *Attention-Based methods (AB)*, described in Sect. 7.2, *Other Methods*, detailed in Sect. 7.4. Additional details are available in (Danilevsky et al. 2020). Table 8 summarizes the explanation methods acting on text data. Text, unlike tabular and image data, does not have a regular structure. Indeed, the variety and complexity of tasks related to text are enormous, and in literature, it is known as *Natural Language Processing (NLP)* (Chowdhary 2020). In the following, we analyze text classification in detail because, among information retrieval, machine translation, and question answering, text classification is the main topic where XAI

Table 7 Explanation runtime of the tested methods, expressed in seconds for explainers of image classifiers, uncertainty is on the last decimal

Dataset	Black-box	LIME	ϵ- LRP	INTGRAD	DEEPLIFT	SMOOTHGRAD	XRAI	GRAD- CAM	GRAD- CAM++	RISE	TCAV	MMD- CRITIC	CEM	GUIDEDPROP	ABELE
mnist	CNN	1	1	0.03	2	0.04	1	0.1	0.1	0.5	–	124	580	11	2000
cifar	CNN	10	1	0.06	1	0.07	1.5	0.15	0.15	2	–	277	765	153	1800
imagenet	VGG16	50	2	5	3	0.8	18	0.25	0.25	21	300	–	–	–	–

Table 8 Summary of methods for opening and explaining black-boxes for text data

Type	Name	References	Data type	IN/PH	G/L	A/S
SH	SHAP	Lundberg and Lee (2017)	ANY	PH	L	S
	LIME	Ribeiro et al. (2016)	ANY	PH	L	A
	DEEPLIFT	Shrikumar et al. (2017)	ANY	PH	L	S
	INTGRAD	Sundararajan et al. (2017)	ANY	PH	L	S
	L2X	Chen et al. (2018)	ANY	PH	L	A
	LIONETS	Mollas et al. (2019)	ANY	PH	L	S
AB	–	Li et al. (2016)	TXT	PH	L	S
	ATTENTIONMATRIX	Vaswani et al. (2017)	TXT	PH	L	S
	EXBERT	Hoover et al. (2019)	TXT	PH	L	S
CF	SEDC	Martens and Provost (2014)	TXT	PH	L	A
	LASTS	Guidotti et al. (2020b)	TXT	PH	L	S
	XSPELLS	Lampridis et al. (2020)	TXT	PH	L	S
	CAT	Chemmengath et al. (2022)	TXT	PH	L	A
	POLYJUICE	Pezeshkpour et al. (2019)	TXT	PH	L	A
Other	GYC	Madaan et al. (2021)	TXT	PH	L	A
	QUINT	Abujabal et al. (2017)	TXT	PH	L	S
	ANCHOR	Ribeiro et al. (2018)	ANY	PH	L	A
	CRIAGE	Pezeshkpour et al. (2019)	TXT	PH	L	S
	–	Rajani et al. (2019)	TXT	PH	L	S
	LASTS	Guidotti et al. (2020b)	TXT	PH	L	S
	DOCTORXAI	Panigutti et al. (2020)	ANY	PH	L	S

methods exist in literature. Examples of usage in text classification are sentiment analysis, topic labeling, spam, and hate detection (Aggarwal and Zhai 2012). Text classification is the process of assigning tags or categories to text according to its content. Using labeled examples as training data, a ML model can learn the different associations between pieces of text and a particular output called tags. Tags can be thought of as labels that distinguish different types of text. For sentiment analysis, it is possible to have tags as positive, negative, or neutral. XAI techniques are generally applied to understand which (sets of) words are the most relevant for a specific tag assignment. We experimented on three datasets: `sst`, `imdb`, and `yelp`. We selected these datasets[20] because they are the most used on sentiment classification and have different dimensions. On these datasets, we trained different black-box models. For every explainer, we present an example of an application on one or more datasets.

[20] `sst`: https://nlp.stanford.edu/sentiment/index.html, `imdb`: https://ai.stanford.edu/~amaas/data/sentiment/, `yelp`: https://www.kaggle.com/yelp-dataset/yelp-dataset.

Fig. 14 Example of sentence highlighting, on top, we have the score produced by INTGRAD and below we have in order, LIME, DEEPLIFT and the baseline, which consists of multiplying the input with the gradient w.r.t. input. The sentence is taken from `imdb`

7.1 Sentence highlighting

As seen in Sect. 6.1, saliency-based explanations are prevalent because they present visually perceptive explanations. *Saliency Highlighting (SH)* can be seen as a version of SMs applied to text. Indeed, it practically consists of assigning to every word (or set of words) a score based on the importance that that word (or set of words) had in the final prediction. Formally, Sentence Highlighting (SH) is modeled as a vector s that explains a classification $y = b(x)$ of a black-box b on x. The dimensions of s are the words present in the sentence x we want to explain, and the value s_i is the saliency value of the word i. The greater the value of s_i, the greater the importance of that word. A positive value indicates a positive contribution towards y, while a negative one means that the word has contributed negatively. Examples of SH explanations are reported in Fig. 14.

To obtain such an explanation, it is possible to adapt some of the SMs methods presented in Sect. 6.1. LIME (Ribeiro et al. 2016), presented in Sect. 5, can be applied to text with a modification to the perturbation of the original input. Given an input sentence x, LIME creates a neighborhood of sentences by replacing one or multiple words with empty spaces. Another possible variation is to insert a stop word instead of removing it to maintain the meaning of the sentence. INTGRAD (Sundararajan et al. 2017), presented in Sect. 6, can also be exploited to explain text classifiers. Indeed, gradient-based methods are challenging to apply to NLP models because the vector representing every word is usually averaged into a single sentence vector. Since a mean operation gradient does not exist, the explainer cannot redistribute the signal back to the original vectors. On the other hand, INTGRAD is immune to this problem because the saliency values are computed as a difference with a baseline value. INTGRAD computes the saliency value of a single word as a difference from the sentence without it. For a fair comparison, we substituted the words with spaces as done for LIME and also for INTGRAD. DEEPLIFT (Shrikumar et al. 2017), presented in Sect. 6, can also be applied to text following the same principle of INTGRAD. For the experiments, we adopt the same preprocessing used for LIME and INTGRAD. L2X (Chen et al. 2018) can produce a SH explanation for text. In particular, for text, the patches adopted are a group of words. Finally, a baseline for text explainers is formed by GRADIENT $\times$ INPUT that uses the black-box gradient of the input w.r.t to the output and multiply these value by the input values.

Sentence highlighting explainers comparison The SM-based explainers for images can be applied to text with minor modifications, which is why it turns out to be the most popular type of explanation also for textual data. As said for explanations on images in Sect. 6, these low feature explanations are helpful to check the model's robustness, not to give a valuable understanding for the final inexpert user. In addition, removing a single word from a sentence is typically not a good measure of the goodness of an explanation since the sentence might lose its meaning. Among the methods tried, INTGRAD is the fastest and most reliable due to its use of the gradients of the black-box. DEEPLIFT also uses the gradients, but its calculation method fails and returns an explanation very similar to the baseline, as reported in Fig. 14. LIME is the only model-agnostic method; however, NLP models tend to be very deep and are generally gradient-based. Thus, LIME results in being very slow, and its performance is clearly inferior to gradient-based methods.

Methods that use gradients such as INTGRAD perform better on the textual data since very deep ML models are usually implemented in the NLP field.

7.2 Attention-based explainers

Attention was first proposed for images in (Xu et al. 2015) to improve the model performance. The authors managed to show through an attention layer which parts of the image most contributed to realize the caption. Formally, the attention layer is indeed a layer to put on top of the model that, for each word, ij of the sentence x generates a positive weight α_{ij}, i.e., the *attention* weight. This value can be interpreted as the probability that a word ij is in the right place to focus on producing the next word in the caption. Attention mechanisms allow models to look over all the information the original sentence holds and learn the context (Wu and Ong 2021; Bahdanau et al. 2015). Therefore, it has caught the interest of XAI researchers, who started using these weights as an explanation. The explanation e of the instance x is composed of the attention values (α), one for each feature x_i. Attention is nowadays a delicate argument, and while it is clear that it augments the performance of models, it is less clear if it helps gain interpretability and the relationship with model outputs (Jain and Wallace 2019). Attention Based Heatmap (Li et al. 2016) is a local, intrinsic, model-specific explainer based on the attention mechanism. It produces a heatmap explanation similar to the one used for SMs by using the weights of the black-box. It can only be applied to attention-based methods, such as BERT, in which the weights α_{ij} of the attention layers are used as a score for every word in the sentence. The higher the score, the more important the word; therefore, the redder the heatmap. Attention Matrix (Cheng et al. 2016) looks at the dependencies between words for producing explanations. It is a self-attention method, sometimes called *intra-attention*. ATTENTION MATRIX relates different positions of a single sequence to compute its internal representation. The attention of a sentence x composed of N words can be understood as an $N \times N$ matrix, where each row and columns represent a word in the input sentence. The values of the matrix are the attention values of every possible combination of the tokens. This matrix is a representation of values pointing from each word to every other word (Vaswani et al. 2017) (see Fig. 15). We can also visualize

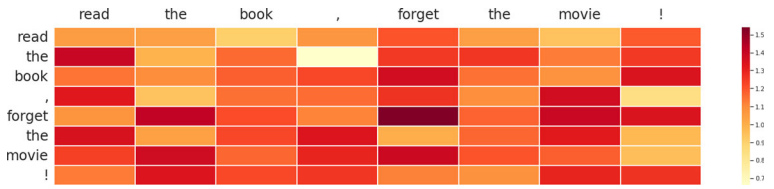

Fig. 15 Attention based heatmap matrix generated from the method presented in (Cheng et al. 2016). The row and the columns of the matrix correspond to the words in the sentence: "Read the book, forget the movie!". Each value of the matrix shows the attention weight α_{ij} of the annotation of the i-th word w.r.t. the j-th

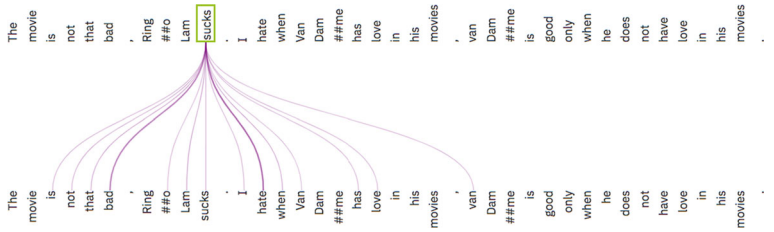

Fig. 16 Attention based representation of BERT for a sentence taken from `imdb` using the visualization of (Hoover et al. 2019). The greater the attention between two words, the bigger the line. Here is selected only the attention related to the word "sucks"

this matrix with a focus on the connection between words (Hoover et al. 2019) as in Fig. 16, where the thickness of the lines is the self-attention value between two tokens.

Attention based explainers comparison Attention is a mechanism good for improving the performance of the model but is not usable as an explanation. As described in (Jain and Wallace 2019), it is unclear what relationship exists between attention weights and model outputs. Learned attention weights are frequently uncorrelated with gradient-based measures of feature importance, and it is possible to identify very different attention distributions that nonetheless yield equivalent predictions. Therefore we might suggest focusing on other types of explanations for text data.

It is unclear if attention can be considered a valid explanation. We suggest focusing on other types of explanations.

7.3 Prototype and counterfactual-based explainers

Counterfactual and prototype explanations are not very common explanations for text data. The richness of meaning in the textual data makes it complex to generate explanations since changing even a single word can deeply alter the meaning of the sentence. In this section, we report the main existing methods that attempt counterfactual or prototype explanations for text data.

PROTOTEX (Das et al. 2022) is a local, intrinsic, model specific NLP classification architecture based on prototype network (Li et al. 2018). PROTOTEX faithfully explains model decisions based on prototype tensors. At inference time, classification decisions are based on the distances between the input text and the prototype tensors,

explained via the training examples most similar to the most influential prototypes. SEDC (Martens and Provost 2014) is a local, post-hoc, model-agnostic explainer able to provide counterfactual instances for documents. The approach proposed is based on a best-first search with pruning. The idea is that, given the words in the document, the algorithm tries to predict the label by removing a single word at a time: if the prediction changes, then it is added as an explanation. Then, also the combinations of words are considered. CAT, Contrastive Attributed explanations for Text (Chemmengath et al. 2022) is a local, post-hoc, model-agnostic explainer for NLP which returns contrastive explanations through a data twist that guarantees semantically meaningful explanations. In particular, CAT rely on a minimal perturbation approach regularized using a BERT language model (Hoover et al. 2019) and an attribute classifier trained on available attributes on the text. POLYJUICE (Wu et al. 2021a) is a local, post-hoc, model-agnostic counterfactual generator that returns a set of realistic textual counterfactuals that can be employed for explanation purposes. In particular, it accounts for returning counterfactuals that are grammatically correct besides being minimal and realistic. The generation makes use of a fill-in-the-blank structure to specify where the perturbation occurs and control codes like negation, delete, insert, shuffle, etc., to specify how it occurs. GYC, Generate Your Counterfactuals (Madaan et al. 2021), is a local, post-hoc, model-agnostic framework to generate diverse counterfactual texts for testing automated decision-making systems. GYC returns plausible, diverse, goal-oriented, and effective explanations through an approach based on constraints that can be specified to guide the generation w.r.t. custom defined class labels, name-entities, topics, etc. Like POLYJUICE, also this approach is not specifically designed for XAI, but it can be easily used for interpretability purposes. XSPELLS (Lampridis et al. 2020) is a local, post-hoc, model-agnostic explainer returning exemplars and counterexamples sentences. In practice, it re-implements ABELE for text data by using LSTM layers in the autoencoder. Exemplars and counter-exemplars are selected using rules extracted from the decision tree learned in the latent space.

Prototype and counterfactual explainers comparison Counterfactuals and prototypes explanations for text are very difficult to generate. CAT and SEDC alter single words to change the prediction of the model; the former is more advanced, substituting words with an NLP model to make correct sentences. However, as seen previously, changing a single word hardly alters the meaning of the sentence and, therefore, the black-box prediction. GYC and POLYJUICE are interactive tools in which human intervention is needed, so they can be used for explanation purposes. XSPELL is the most automatic approach, which seems to be also the best one among the others, but it suffers from the same limitations of ABELE. The only prototype explanation method we identified for text is PROTOTEX.

> Prototype and Counterfactual explanations are very difficult to generate for text data due to the meaning of the sentences. Interactive methods such as POLYJUICE and GYC are promising approaches that introduce the human cognitive process that helps in the process of counterfactual generation.

Table 9 Deletion (top) and insertion (bottom) metrics computed on Sentence Highlighting for different datasets

	sst	imdb	yelp
INTGRAD	0.6447 (0.21)	0.647 (0.21)	0.7595 (0.25)
LIME	0.6199 (0.23)	0.648 (0.21)	0.7712 (0.25)
DEEPLIFT	0.6297 (0.23)	0.600 (0.15)	0.7565 (0.31)
GRADIENT X INPUT	0.6287 (0.23)	0.630 (0.16)	0.7590 (0.28)
INTGRAD	0.6107 (0.23)	0.616 (0.16)	0.7625 (0.33)
LIME	0.6337 (0.23)	0.599 (0.17)	0.7513 (0.33)
DEEPLIFT	0.6137 (0.21)	0.645 (0.16)	0.7524 (0.30)
GRADIENT X INPUT	0.5852 (0.22)	0.632 (0.16)	0.7479 (0.31)

The reported value represents the mean of the scores obtained on a subset of 100 instances, and the value on the parenthesis is the standard deviation

7.4 Other types of explainers

In this section, we report on other types of explainers that may not fit into one of the previous sections but are nonetheless relevant to the field. ANCHOR (Ribeiro et al. 2018), presented in Sect. 5, can be adapted to text by using as perturbation the word UNK. It consists of perturbing a sentence by substituting words with UNK (unknown). For example, It shows how "sucks" contributed to the negative prediction of the sentence, but when coupled with "love", the sentence prediction switches to positive. Natural Language Explanation verbalizes explanations in natural human language. Natural language can be generated with complex *deep learning models*, e.g., by training a model with natural language explanations and coupling it with a generative model (Rajani et al. 2019). Besides, it can also be generated using a simple *template-based approach* (Abujabal et al. 2017). NLP is a very complex field, and finding a human-friendly explanation is challenging. Researchers are working in various directions of creating explanations with high concept (Srivastava et al. 2017) and using humans to augment these types of concept (Rajani et al. 2019), as done for Concept Attribution. Another possible development is the use of generative text methods to build a friendly narrative for the final user, which is not always an expert in the field.

> Explanations for classifiers acting on text data are at the very early stages compared to tabular data and images.

7.5 Text explainers quantitative comparison

Similarly to what we did for images, to quantitatively investigate the performance of the SH explainers analyzed, we computed the *deletion* and the *insertion* metric, presented in Sect. 4. For the computation of the *deletion* metric, we removed words in order of importance. In contrast, for the insertion metric, we started with an empty text, and we added words in order of importance. For every substitution we made, we

queried the text to the black-box, obtaining accuracy. The final score is obtained by taking the area under the curve (AUC) (Hand and Till 2001) of accuracy as a function of the percentage of removed words. The results are shown in Fig. 14. We notice that the highlighted words are very different among the various methods. INTGRAD and LIME are the ones who return more meaningful explanations, while DEEPLIFT struggles a lot to diversify from the baseline. We also measured the *deletion/insertion* and reported the results in Table 9. For both metrics, we have very poor performance among all the methods. However, this metric is not ideal with text since removing a single word barely changes the meaning of the sentence. On the other hand, concerning runtime, NLP models are usually very deep and large, resulting in poor performance in terms of runtime. In particular LIME is the slowest since it queries the black-box for prediction multiple times. DEEPLIFT and INTGRAD are faster since they operate on the gradients of the black-box. Finally, AB methods are instant since they only need to extract the weights and do not need to perform any operation.

8 Explainers for other data types

As detailed in this survey, the state-of-the-art is mainly focusing on the explanation of black-box trained on tabular data, images, and texts. However, other types of data are largely available, and many models are built on top of them. In this section, we shortly summarize some explainers, which are pillars for the explanations of black-boxes trained for time series and graph classification.

8.1 Explanations for time series

Due to the tremendous amount of data generated by sensors over time, there is a widespread diffusion of ML models working on *time series* (Theissler 2017). A *time series* $x = \{t_1, t_2, \ldots, t_m\} \in \mathbb{R}^{m \times d}$ is an ordered set of m real-valued observations (or time steps), with dimensionality d. We say that a time series is *univariate* when $d = 1$, while when $d > 1$, we name x a *multivariate time series*. There are areas such as the medical or financial field where temporal data is of particular importance and where black-box ML models are applied to provide support on decision-making for various tasks. For this reason, recently, we have been assisting with the emerging proposal for explainability related to time series (Theissler et al. 2022).

Many explainers for time series can be categorized with the taxonomy introduced in Sect. 3. The most important difference with the other types of data relies on the type of explanation produced. Shapelet is the most characteristic explanation for time series data. They are time series subsequences that are maximally representative of a class. Shapelets are more interpretable, faster, and more accurate than k-Nearest Neighbors (kNN) (Cover and Hart 1967), which is a traditional approach to perform time series classification (Lee et al. 2012). As usual, SMs can be used to highlight which part of the series has contributed the most to the classification. Finally, also the attention methods illustrated for text in Sect. 7.2 can also be applied to time series data. In the following, we briefly illustrate some peculiar explainers for time series classification.

In (Geler et al. 2020), introduced a transparent by design method named Weighted-kNN that extends the classic majority-voting kNN by proposing weighting schemes. By emphasizing the nearer neighbors using a weighting scheme, it is possible to improve the kNN classifier's quality and stability. The nearest neighbors are considered part of the prototypical explanation. LASTS, Local Agnostic Shapelet-based Time Series explainer (LASTS) (Guidotti et al. 2020b), is a variation of ABELE for time series. As explanation LASTS returns exemplars and counterexamples composed of subseries with a shapelet-based rule. An example of a rule is: "if these shapelets are present and these others not, then x is classified as y". DOCTORXAI (Panigutti et al. 2020) is a local post-hoc model-specific explainer acting on sequential data in the medical setting. In particular, it exploits a medical ontology to perturb the data and generate neighbors. DOCTORXAI is designed on healthcare data, but it can theoretically be applied to every type of sequential data with an ontology. For more information on explaining black-box models for time series, we remand the reader to the following survey (Rojat et al. 2021; Theissler et al. 2022).

Time series explainers comparison For time series data, kNN weighting schemes are the most common approach. Shaplet-based explanations are promising, and new approaches using autoencoders or ontologies are being developed to improve time series explanations.

8.2 Explanations for graphs

Graph Neural Networks (GNNs) have become increasingly popular since many real-world data are represented as graphs, such as social networks, chemical molecules, and financial data. A graph can be used for specific tasks like link prediction (Cai and Ji 2020) or node labeling (Calamoneri 2006). However, many advanced GNN operations are also proposed to improve the performance of classification models, for example, in graph convolution models (Kipf and Welling 2017). Compared with image and text domains, the explainability of black-box models working on graphs is less explored, but it is critical for understanding to behavior of GNN.

The types of explanations available for graphs are various. The most common one is node relevance which derives from feature relevance methods. It consists of assigning a value to every node of the graph that represents how much that node has contributed to the prediction. Since graphs are composed of nodes and edges, the same operation can be accomplished with edges. The focus of the explanation is shifted toward the interactions between the nodes instead of the nodes themselves. There are different ways to do this. The most common one is to backpropagate the signal from the prediction to the input graph in LRP style. Other approaches learn a surrogate model to assign values as done by LIME. LRP (for Graphs) (Schwarzenberg et al. 2019) extends the original LRP method for GNN. It decomposes the output prediction score into different node importance scores. The score decomposition rule is developed based on the hidden features and weights. GraphLIME (Huang et al. 2020) extends the LIME for GNN and studies the importance of different node features for node classification tasks. Given a target node in the input graph, GRAPHLIME considers its N-hop neighboring nodes and their predictions as its local neighborhood, where a

reasonable choice of N is the number of layers in the GNNs. Then a nonlinear surrogate model is employed to fit the local neighborhood. Finally, based on the weights of different features in the surrogate model, it selects important features to explain the predictions. XGNN (Yuan et al. 2020a) (eXplainable GNN) is a global, post-hoc, model-specific explainer that proposes graph patterns as explanations to investigate which structure in the graph maximizes a specific prediction. Specifically, they trained a graph generator to generate small graphs, which pattern can be used to explain GNN behavior. The graph generator uses reinforcement learning that starts from an empty graph and, at each step, determines how to add an edge or node and form a new graph. This graph generator is trained based on feedback from trained graph models using a policy gradient algorithm. CF-GNNExplainer (Lucic et al. 2022) is a local, post-hoc, model-specific explainer for GNNs. It aims at finding the minimal perturbation to the graph such that the prediction changes. In particular, CF- GNNEXPLAINER iteratively removes edges from the original adjacency matrix based on a matrix sparsification technique, keeping track of the perturbations that lead to a change in prediction and returning the perturbation with the smallest change w.r.t. the number of edges. For more information on explaining black-box models for graphs, we remand the reader to the following survey (Prado-Romero et al. 2022; Yuan et al. 2020c).

GNN explainers comparison GNN models are becoming more and more common in literature (Wu et al. 2021b), and due to their similarity to neural networks, researchers are trying to adapt existing methods to this type of black-box. GRAPHLIME and GRAPHLRP) are two perfect examples of that, the former adapting LIME while the latter ϵ- LRP. Both methods return scores for nodes, but GRAPHLIME does not output a score for the edges features as it totally ignores the graph structure, which is more critical for graph data. XGNN and CF- GNNEXPLAINER are more aimed at counterfactual explanations. XGNN returns more robust counterfactuals due to the reinforcement learning policy; however, it is much slower than CF- GNNEXPLAINER for the same reason.

9 Explanation toolboxes

A significant number of toolboxes for ML explanation have been proposed during the last few years. In the following, we report the most popular Python toolkits with a brief description of the explanation models they provide.[21]

AIX360 (Arya et al. 2019) contains both intrinsic, post-hoc, local, and global explainers, and it can be used with every kind of input dataset. Regarding local post-hoc explanations, different methods are implemented, such as LIME (Ribeiro et al. 2016), SHAP (Lundberg and Lee 2017), CEM (Dhurandhar et al. 2018), CEM- MAF (Luss et al. 2019) and PROTODASH (Gurumoorthy et al. 2019)). Another interesting method proposed in this toolkit is TED (Hind et al. 2019; Dash et al. 2018), which provides intrinsic local explanations and provides global explanations based on rules. CaptumAI is a

[21] AIX360: https://github.com/Trusted-AI/AIX360, CaptumAI: https://captum.ai/, InterpretML: https://github.com/interpretml/interpret, Alibi https://github.com/SeldonIO/alibi, FAT-Forensics: https://github.com/fat-forensics/fat-forensics, What-If Tool: https://github.com/pair-code/what-if-tool, Shapash: https://github.com/MAIF/shapash.

library built for PyTorch models. CaptumAI divides the available algorithms into three categories: *Primary Attribution*, in which there are methods able to evaluate the contribution of each input feature to the output of a model: INTGRAD (Sundararajan et al. 2017), GRAD- SHAP (Lundberg and Lee 2017), DEEPLIFT (Shrikumar et al. 2017), LIME (Ribeiro et al. 2016), GRAD- CAM (Selvaraju et al. 2020). *Layer Attribution*, in which the focus is on the contribution of each neuron: e.g. GRAD- CAM (Selvaraju et al. 2020) and LAYER- DEEPLIFT (Shrikumar et al. 2017). *Neuron Attribution*, in which is analyzed the contribution of each input feature on the activation of a particular hidden neuron: e.g. NEURON- INTGRAD (Sundararajan et al. 2017), NEURON- GRAD-SHAP (Lundberg and Lee 2017). InterpretML[22] (Nori et al. 2019) contains intrinsic and post-hoc methods for Python and R. InterpretML is particularly interesting due to the *intrinsic* methods it provides: Explainable Boosting Machine (EBM), Decision Tree, and Decision Rule List. These methods offer a user-friendly visualization of the explanations, with several local and global charts. InterpretML also contains the most popular methods, such as LIME and SHAP. DALEX (Lipovetsky 2022) is an R and Python package that provides post-hoc and model-agnostic explainers that allow local and global explanations. It is tailored for tabular data and can produce different kinds of visualization plots. Alibi provides intrinsic and post-hoc models. It can be used with *any type of input dataset* and *both for classification and regression* tasks. Alibi provides a set of counterfactual explanations, such as CEM, and, interestingly, an implementation of ANCHOR (Ribeiro et al. 2018). Regarding global explanation methods, Alibi contains ALE (Accumulated Local Effects) (Apley and Zhu 2016), which is a method based on partial dependence plots (Guidotti et al. 2019c). FAT-Forensics takes into account *fairness, accountability and transparency*. Regarding intrinsic explainability, it provides methods to assess explainability under three perspectives: data, models, and predictions. For *accountability*, it offers a set of techniques that *assesses privacy, security, and robustness*. For *fairness*, it contains methods for *bias detection*. What-If Tool is a toolkit providing a visual interface from which it is possible to play without coding. Moreover, it can work directly with ML models built on *Cloud AI Platform* (https://cloud.google.com/ai-platform). It contains a variety of approaches to get feature attribution values such as SHAP (Lundberg and Lee 2017), INTGRAD (Sundararajan et al. 2017), and SMOOTHGRAD (Selvaraju et al. 2020). Shapash is a Python library that aims to make machine learning interpretable and understandable by everyone. It provides several types of interpretable visualization that display explicit labels that everyone can understand. Shapash offers different types of interactive visualization, from feature importance graphs to contributions ones.

10 Conclusion

In this paper, we have presented a survey of the latest advances in XAI methods, following a categorization based on the data type and explanation type. A subset of widely adopted explainers has been benchmarked with a quantitative and qualitative comparison. Among those explainers, LIME and SHAP are probably the most widely

[22] https://github.com/interpretml/interpret.

used methods but also among the most unstable, with LIME much less stable than SHAP. We observed that local explainers using surrogate models like LIME suffer from a lack of stability due to the high level of randomness in the neighborhood generation approach. The solid theoretical background of SHAP accounts for more faithful explanations. The explanations returned by LIME and SHAP are based on (individual) features importance, an approach that is often hardly understandable, especially for tabular data. As a recent trend, rule-based explanations and counterfactual explanations are gaining attention since their logic formalization supports a deeper understanding of the model's internal decisions. Unfortunately, metrics to compare such types of models are still missing in the literature. For image data, there is a very wide variety of methods that provide explanations through different implementations of saliency maps. According to our experiments, the best approach is the masking method introduced by RISE. Gradient methods like INTGRAD, ϵ- LRP and DEEPLIFT are very close runners-up in our ranking, while segmentation methods like LIME and XRAI produce less convincing results. Concept Attribution methods are extremely promising but also hard to validate, as they require additional knowledge about involved concepts. Prototypes and counter-exemplars have received increasing attention in recent years but suffer from a lack of a validation framework and are difficult to use in specific contexts of application. Finally, we noticed that there are very limited explanation techniques for text data. One of the pioneers is Sentence Highlight, that, similarly to feature importance for tabular data, gives weight to the portion of the input that contributed, positively or negatively, to the outcome. This strategy, customized for text data, is not mature yet.

Across the different data types, different approaches tend to use similar strategies. This is also evident if we look at the internals of these algorithms. For example, several methods exploit the generation of a synthetic neighborhood around an instance to reconstruct the local distribution of data around the point to investigate. This approach is prevalent when trying to produce counterfactual images. Another widespread approach is to use the gradients of the black-box, if available. The best results are obtained by combining the use of the gradients with a masking methodology.

In recent years the contributions in the Explainable AI area have constantly been growing, particularly explainers for machine learning models and, to a minor extent, novel *explainable-by-design* methods. To date, limited attention is devoted to the comparison of the explainers proposed, mostly using measures of the intrinsic, standalone quality of an explainer, such as the metrics considered in this paper. A more holistic validation approach is needed, that brings the user into the picture to assess whether a proposed explanation is meaningful to the user's purpose and whether the interaction with the AI system increases the user's trust. This emerging approach requires the careful design of user studies that evaluate the combined human-AI ecosystem, considering the cognitive model of the user. As an early example along this line, a user study on XAI-assisted clinical decision-making revealed that XAI-based advice yields higher (expert) user trust, compared to an AI-based advise (without explanation) (Panigutti et al. 2022). Experiments like this also typically reveal the dissatisfaction of users with the proposed explanation interface, a crucial aspect that suggests how AI-assisted decision-making systems should be *co-designed* in a synergistic process where all relevant stakeholders are involved from the beginning. While some preliminary results are emerging (Guidotti 2021; Jeyakumar et al. 2020; Hase and Bansal 2020), there

is still a long way to go towards explainable machine learning paradigms based on the interaction and collaboration with the user(s), so that the decision makers and the ML model can co-evolve together, exploiting their complementary strengths and becoming progressively more effective and trustworthy. This is ultimately a key goal of Human-centered AI: enhancing the capacity of humans to solve hard problems and make wise decisions. We hope that the current developments of explainable AI may represent a basis on which the novel paradigm may eventually flourish.

Acknowledgements This work has been partially supported by the European Community Horizon 2020 program under the funding schemes: G.A. 871042 *SoBigData++* (http://www.sobigdata.eu), G.A. 952026 *HumanE AI Net* (https://www.humane-ai.eu/), ERC-2018-ADG G.A. 834756 *XAI: Science and technology for the eXplanation of AI decision making* (https://xai-project.eu/index.html), G.A. 952215 *TAILOR* (https://tailor-network.eu/), and G.A. CHIST-ERA-19-XAI-010 *SAI*, by MUR (N. not yet available), FWF (N. I 5205), EPSRC (N. EP/V055712/1), NCN (N. 2020/02/Y/ST6/00064), ETAg (N. SLTAT21096), BNSF (N. KP-06-AOO2/5) (https://www.sai-project.eu/). And the NextGenerationEU program under the funding schemes PNRR-PE-AI scheme (M4C2, investment 1.3, line on AI) FAIR (Future Artificial Intelligence Research), and "SoBigData.it—Strengthening the Italian RI for Social Mining and Big Data Analytics"—Prot. IR0000013. H2020 Research Infrastructures (Grant No. 654024).

Author Contributions FB, FN: Conceptualization, methodology, software, validation, investigation, resources, visualization, writing—original draft, writing—review and editing. RG: Conceptualization, methodology, resources, writing—original draft, writing—review and editing, visualization, supervision. SR: Conceptualization, investigation, writing—review and editing, visualization, supervision. FG, DP: Conceptualization, investigation, writing—review, project administration, supervision.

Funding Open access funding provided by Scuola Normale Superiore within the CRUI-CARE Agreement.

Data availability The datasets adopted in this work are open source and available at https://archive.ics.uci.edu/ml/datasets.php for `adultn` and `german`, https://www.image-net.org/ for `imagenet`, http://yann.lecun.com/exdb/mnist/ for `mnist`, https://www.cs.toronto.edu/kriz/cifar.html for `cifar`, https://nlp.stanford.edu/sentiment/code.html for `sst`, and www.kaggle.com/datasets for `imdb` and `yelp`.

Code Availability The code is open source and can be downloaded at https://github.com/kdd-lab/XAI-Survey.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication The authors declare that they all provide consent for publication.

References

Abujabal A, Roy RS, Yahya M, et al (2017) QUINT: interpretable question answering over knowledge bases. In: Proceedings of the 2017 conference on empirical methods in natural language processing, EMNLP 2017, Copenhagen, Denmark—system demonstrations

Adadi A, Berrada M (2018) Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access

Adebayo J, Gilmer J, Muelly M, et al (2018) Sanity checks for saliency maps. In: Advances in neural information processing systems 31: annual conference on neural information processing systems 2018, NeurIPS 2018, Montréal, Canada

Adebayo J, Muelly M, Liccardi I, et al (2020) Debugging tests for model explanations. In: Advances in neural information processing systems 33: annual conference on neural information processing systems 2020, NeurIPS 2020, virtual

Agarwal R, Melnick L, Frosst N, et al (2021) Neural additive models: Interpretable machine learning with neural nets. In: Advances in neural information processing systems 34: annual conference on neural information processing systems 2021, NeurIPS 2021, virtual

Aggarwal CC, Zhai C (2012) A survey of text classification algorithms. In: Mining text data. Springer, pp 163–222

Albini E, Rago A, Baroni P, et al (2020) Relation-based counterfactual explanations for bayesian network classifiers. In: Proceedings of the twenty-ninth international joint conference on artificial intelligence, IJCAI 2020

Alvarez-Melis D, Jaakkola TS (2018) Towards robust interpretability with self-explaining neural networks. In: Advances in neural information processing systems 31: annual conference on neural information processing systems 2018, NeurIPS 2018, Montréal, Canada

Anjomshoae S, Najjar A, Calvaresi D, et al (2019) Explainable agents and robots: Results from a systematic literature review. In: Proceedings of the 18th international conference on autonomous agents and multiagent systems, AAMAS '19, Montreal, QC, Canada

Anjomshoae S, Kampik T, Främling K (2020) Py-ciu: a python library for explaining machine learning predictions using contextual importance and utility. In: IJCAI-PRICAI 2020 workshop on explainable artificial intelligence (XAI)

Apley DW, Zhu J (2016) Visualizing the effects of predictor variables in black box supervised learning models. arXiv preprint arXiv:1612.08468

Arras L, Montavon G, Müller K, et al (2017) Explaining recurrent neural network predictions in sentiment analysis. In: Proceedings of the 8th workshop on computational approaches to subjectivity, sentiment and social media analysis, WASSA@EMNLP 2017, Copenhagen, Denmark

Arrieta AB, Rodríguez ND, Ser JD, et al (2020) Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf Fus

Artelt A, Hammer B (2019) On the computation of counterfactual explanations—a survey. arXiv preprint arXiv:1911.07749

Arya V, Bellamy RKE, Chen P, et al (2019) One explanation does not fit all: A toolkit and taxonomy of AI explainability techniques. arXiv preprint arXiv:1909.03012

Bach S, Binder A, Montavon G et al (2015) On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS One 10(7):e0130140

Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: 3rd International conference on learning representations, ICLR 2015, San Diego, CA, USA, conference track proceedings

Bien J, Tibshirani R (2011) Prototype selection for interpretable classification. Ann Appl Stat 2403–2424

Blanco-Justicia A, Domingo-Ferrer J, Martínez S, et al (2020) Machine learning explainability via microaggregation and shallow decision trees. Knowl Based Syst

Boz O (2002) Extracting decision trees from trained neural networks. In: Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining, Edmonton, Alberta, Canada

Bramhall S, Horn H, Tieu M et al (2020) Qlime-a quadratic local interpretable model-agnostic explanation approach. SMU Data Sci Rev 3(1):4

Byrne RM (2019) Counterfactuals in explainable artificial intelligence (XAI): evidence from human reasoning. In: IJCAI, pp 6276–6282

Byrne RM, Johnson-Laird P (2020) If and or: real and counterfactual possibilities in their truth and probability. J Exp Psychol Learn Mem Cogn 46(4):760
Cai L, Ji S (2020) A multi-scale approach for graph link prediction. In: The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, the thirty-second innovative applications of artificial intelligence conference, IAAI 2020, the tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020, New York, NY, USA
Calamoneri T (2006) The L(h, k)-labelling problem: a survey and annotated bibliography. Comput J
Carvalho DV, Pereira EM, Cardoso JS (2019) Machine learning interpretability: a survey on methods and metrics. Electronics 8(8):832
Chattopadhay A, Sarkar A, Howlader P, et al (2018) Grad-cam++: generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE winter conference on applications of computer vision (WACV), IEEE
Chemmengath SA, Azad AP, Luss R, et al (2022) Let the CAT out of the bag: Contrastive attributed explanations for text. In: Proceedings of the 2022 conference on empirical methods in natural language processing, EMNLP 2022, Abu Dhabi, United Arab Emirates
Chen J, Song L, Wainwright MJ, et al (2018) Learning to explain: an information-theoretic perspective on model interpretation. In: Proceedings of the 35th international conference on machine learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden
Chen C, Li O, Tao D, et al (2019) This looks like that: deep learning for interpretable image recognition. In: Advances in neural information processing systems 32: annual conference on neural information processing systems 2019, NeurIPS 2019, Vancouver, BC, Canada
Cheng J, Dong L, Lapata M (2016) Long short-term memory-networks for machine reading. In: Proceedings of the 2016 conference on empirical methods in natural language processing, EMNLP 2016, Austin, Texas, USA
Chipman H, George E, McCulloh R (1998) Making sense of a forest of trees. Comput Sci Stat
Chouldechova A (2017) Fair prediction with disparate impact: a study of bias in recidivism prediction instruments. Big Data
Chowdhary K (2020) Natural language processing. In: Fundamentals of artificial intelligence. Springer, pp 603–649
Chowdhury T, Rahimi R, Allan J (2022) Equi-explanation maps: concise and informative global summary explanations. In: 2022 ACM conference on fairness, accountability, and transparency, FAccT '22
Cover TM, Hart PE (1967) Nearest neighbor pattern classification. IEEE Trans Inf Theory
Craven MW, Shavlik JW (1995) Extracting tree-structured representations of trained networks. In: Advances in neural information processing systems 8, NIPS, Denver, CO, USA
Danilevsky M, Qian K, Aharonov R, et al (2020) A survey of the state of explainable AI for natural language processing. In: Proceedings of the 1st conference of the Asia-Pacific chapter of the association for computational linguistics and the 10th international joint conference on natural language processing, AACL/IJCNLP 2020, Suzhou, China
Das A, Gupta C, Kovatchev V, et al (2022) Prototex: explaining model decisions with prototype tensors. In: Proceedings of the 60th annual meeting of the association for computational linguistics (vol. 1: long papers), ACL 2022, Dublin, Ireland
Dash S, Günlük O, Wei D (2018) Boolean decision rules via column generation. In: Advances in neural information processing systems 31: annual conference on neural information processing systems 2018, NeurIPS 2018, Montréal, Canada
Desai S, Ramaswamy HG (2020) Ablation-cam: visual explanations for deep convolutional network via gradient-free localization. In: IEEE winter conference on applications of computer vision, WACV 2020, Snowmass Village, CO, USA
Dhurandhar A, Chen P, Luss R, et al (2018) Explanations based on the missing: towards contrastive explanations with pertinent negatives. In: Advances in neural information processing systems 31: annual conference on neural information processing systems 2018, NeurIPS 2018, Montréal, Canada
Doersch C (2016) Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908
Domingos PM (1998) Knowledge discovery via multiple models. Intell Data Anal 2(1–4):187–202
Donnelly J, Barnett AJ, Chen C (2022) Deformable protopnet: an interpretable image classifier using deformable prototypes. In: CVPR. IEEE, pp 10255–10265
Doshi-Velez F, Kim B (2017) Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608

Došilović FK, Brčić M, Hlupić N (2018) Explainable artificial intelligence: a survey. In: 2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO), IEEE, pp 0210–0215

ElShawi R, Sherif Y, Al-Mallah M, et al (2019) Ilime: local and global interpretable model-agnostic explainer of black-box decision. In: European conference on advances in databases and information systems. Springer, pp 53–68

Erion GG, Janizek JD, Sturmfels P, et al (2019) Learning explainable models using attribution priors. arXiv preprint arXiv:1906.10670

Fong R, Patrick M, Vedaldi A (2019) Understanding deep networks via extremal perturbations and smooth masks. In: 2019 IEEE/CVF international conference on computer vision, ICCV 2019, Seoul, Korea (South)

Freitas AA (2013) Comprehensible classification models: a position paper. SIGKDD Explor 15(1):1–10

Friedman J, Popescu BE (2008) Predictive learning via rule ensembles. Ann Appl Stat 2:916–954

Geler Z, Kurbalija V, Ivanovic M, et al (2020) Weighted KNN and constrained elastic distances for time-series classification. Expert Syst Appl

Ghorbani A, Wexler J, Zou JY, et al (2019) Towards automatic concept-based explanations. In: Advances in neural information processing systems 32: annual conference on neural information processing systems 2019, NeurIPS 2019, Vancouver, BC, Canada

Gilpin LH, Bau D, Yuan BZ, et al (2018) Explaining explanations: an overview of interpretability of machine learning. In: 5th IEEE international conference on data science and advanced analytics, DSAA 2018, Turin, Italy

Gleicher M (2016) A framework for considering comprehensibility in modeling. Big Data 4(2):75–88

Goebel R, Chander A, Holzinger K, et al (2018) Explainable AI: the new 42? In: Machine learning and knowledge extraction—second IFIP TC 5, TC 8/WG 8.4, 8.9, TC 12/WG 12.9 international cross-domain conference, CD-MAKE 2018, Hamburg, Germany, Proceedings

Goyal Y, Shalit U, Kim B (2019) Explaining classifiers with causal concept effect (cace). arXiv preprint arXiv:1907.07165

Guidotti R (2021) Evaluating local explanation methods on ground truth. Artif Intell

Guidotti R (2022) Counterfactual explanations and how to find them: literature review and benchmarking. DAMI, pp 1–55

Guidotti R, Monreale A, Giannotti F, et al (2019a) Factual and counterfactual explanations for black box decision making. IEEE Intell Syst

Guidotti R, Monreale A, Matwin S, et al (2019b) Black box explanation by learning image exemplars in the latent feature space. In: Machine learning and knowledge discovery in databases—European conference, ECML PKDD 2019, Würzburg, Germany, proceedings, part I

Guidotti R, Monreale A, Ruggieri S, et al (2019c) A survey of methods for explaining black box models. ACM Comput Surv

Guidotti R, Monreale A, Matwin S, et al (2020a) Explaining image classifiers generating exemplars and counter-exemplars from latent representations. In: The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, the thirty-second innovative applications of artificial intelligence conference, IAAI 2020, the tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020, New York, NY, USA

Guidotti R, Monreale A, Spinnato F, et al (2020b) Explaining any time series classifier. In: 2nd IEEE international conference on cognitive machine intelligence, CogMI 2020, Atlanta, GA, USA

Gurumoorthy KS, Dhurandhar A, Cecchi GA, et al (2019) Efficient data representation by selecting prototypes with importance weights. In: 2019 IEEE international conference on data mining, ICDM 2019, Beijing, China

Hand DJ, Till RJ (2001) A simple generalisation of the area under the ROC curve for multiple class classification problems. Mach Learn

Hartmann Y, Liu H, Lahrberg S, et al (2022) Interpretable high-level features for human activity recognition. In: Proceedings of the 15th international joint conference on biomedical engineering systems and technologies, BIOSTEC 2022, vol. 4: BIOSIGNALS, Online Streaming

Hase P, Bansal M (2020) Evaluating explainable AI: which algorithmic explanations help users predict model behavior? In: Proceedings of the 58th annual meeting of the association for computational linguistics, ACL 2020, Online

Hastie TJ, Tibshirani RJ (1990) Generalized additive models, vol 43. CRC Press

Hind M, Wei D, Campbell M, et al (2019) TED: teaching AI to explain its decisions. In: Proceedings of the 2019 AAAI/ACM conference on AI, ethics, and society, AIES 2019, Honolulu, HI, USA
Hoover B, Strobelt H, Gehrmann S (2019) exbert: a visual analysis tool to explore learned representations in transformers models. arXiv preprint arXiv:1910.05276
Huang Q, Yamada M, Tian Y, et al (2020) Graphlime: local interpretable model explanations for graph neural networks. arXiv preprint arXiv:2001.06216
Hvilshøj F, Iosifidis A, Assent I (2021) ECINN: efficient counterfactuals from invertible neural networks. In: BMVC. BMVA Press, p 43
Jain S, Wallace BC (2019) Attention is not explanation. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, NAACL-HLT 2019, Minneapolis, MN, USA, vol. 1 (long and short papers)
Jeyakumar JV, Noor J, Cheng Y, et al (2020) How can I explain this to you? An empirical study of deep neural network explanation methods. In: Advances in neural information processing systems 33: annual conference on neural information processing systems 2020, NeurIPS 2020, virtual
Kamakshi V, Gupta U, Krishnan NC (2021) PACE: posthoc architecture-agnostic concept extractor for explaining CNNs. In: International joint conference on neural networks, IJCNN 2021, Shenzhen, China
Kanamori K, Takagi T, Kobayashi K, et al (2020) DACE: distribution-aware counterfactual explanation by mixed-integer linear optimization. In: Proceedings of the twenty-ninth international joint conference on artificial intelligence, IJCAI 2020
Kapishnikov A, Bolukbasi T, Viégas FB, et al (2019) XRAI: better attributions through regions. In: 2019 IEEE/CVF international conference on computer vision, ICCV 2019, Seoul, Korea (South)
Karimi A, Barthe G, Balle B, et al (2020a) Model-agnostic counterfactual explanations for consequential decisions. In: The 23rd international conference on artificial intelligence and statistics, AISTATS 2020, Online [Palermo, Sicily, Italy]
Karimi A, Barthe G, Schölkopf B, et al (2020b) A survey of algorithmic recourse: definitions, formulations, solutions, and prospects. arXiv preprint arXiv:2010.04050
Katehakis Jr MN, Veinott AF (1987) The multi-armed bandit problem: decomposition and computation. Math Oper Res
Kenny EM, Keane MT (2021) On generating plausible counterfactual and semi-factual explanations for deep learning. In: AAAI. AAAI Press, pp 11575–11585
Kim B, Chacha CM, Shah JA (2015) Inferring team task plans from human meetings: a generative modeling approach with logic-based prior. J Artif Intell Res
Kim B, Koyejo O, Khanna R (2016) Examples are not enough, learn to criticize! criticism for interpretability. In: Advances in neural information processing systems 29: annual conference on neural information processing systems 2016, Barcelona, Spain
Kim B, Wattenberg M, Gilmer J, et al (2018) Interpretability beyond feature attribution: quantitative testing with concept activation vectors (TCAV). In: Proceedings of the 35th international conference on machine learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden
Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: 5th International conference on learning representations, ICLR 2017, Toulon, France, conference track proceedings
Koh PW, Liang P (2017) Understanding black-box predictions via influence functions. In: Proceedings of the 34th international conference on machine learning, ICML 2017, Sydney, NSW, Australia
Kurenkov A (2020) Lessons from the pulse model and discussion. The gradient
Lakkaraju H, Bach SH, Leskovec J (2016) Interpretable decision sets: a joint framework for description and prediction. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco, CA, USA
Lampridis O, Guidotti R, Ruggieri S (2020) Explaining sentiment classification with synthetic exemplars and counter-exemplars. In: Discovery science—23rd international conference, DS 2020, Thessaloniki, Greece, Proceedings
Lang O, Gandelsman Y, Yarom M, et al (2021) Explaining in style: training a GAN to explain a classifier in stylespace. In: ICCV. IEEE, pp 673–682
Lapuschkin S, Wäldchen S, Binder A, et al (2019) Unmasking clever hans predictors and assessing what machines really learn. arXiv preprint arXiv:1902.10178
Lee Y, Wei C, Cheng T, et al (2012) Nearest-neighbor-based approach to time-series classification. Decis Support Syst

Letham B, Rudin C, McCormick TH, et al (2015) Interpretable classifiers using rules and Bayesian analysis: building a better stroke prediction model. arXiv preprint arXiv:1511.01644
Ley D, Mishra S, Magazzeni D (2022) Global counterfactual explanations: investigations, implementations and improvements. In: ICLR 2022 workshop on PAIR∧2Struct: privacy, accountability, interpretability, robustness, reasoning on structured data. https://openreview.net/forum?id=Btbgp0dOWZ9
Li J, Monroe W, Jurafsky D (2016) Understanding neural networks through representation erasure. arXiv preprint arXiv:1612.08220
Li O, Liu H, Chen C, et al (2018) Deep learning for case-based reasoning through prototypes: a neural network that explains its predictions. In: Proceedings of the thirty-second AAAI conference on artificial intelligence, (AAAI-18), the 30th innovative applications of artificial intelligence (IAAI-18), and the 8th AAAI symposium on educational advances in artificial intelligence (EAAI-18), New Orleans, Louisiana, USA
Li H, Tian Y, Mueller K, et al (2019) Beyond saliency: understanding convolutional neural networks from saliency prediction on layer-wise relevance propagation. Image Vis Comput
Lipovetsky S (2022) Explanatory model analysis: Explore, explain and examine predictive models, by Przemyslaw Biecek, Tomasz Burzykowski, Boca Raton, FL, Chapman and Hall/CRC, Taylor & Francis Group, 2021, xiii + 311 pp., $ 79.96 (hbk), ISBN 978-0-367-13559-1. Technometrics
Looveren AV, Klaise J (2021) Interpretable counterfactual explanations guided by prototypes. In: Machine learning and knowledge discovery in databases. Research track—European conference, ECML PKDD 2021, Bilbao, Spain, proceedings, part II
Lucic A, Haned H, de Rijke M (2020) Why does my model fail?: Contrastive local explanations for retail forecasting. In: FAT* '20: conference on fairness, accountability, and transparency, Barcelona, Spain
Lucic A, ter Hoeve MA, Tolomei G, et al (2022) Cf-gnnexplainer: counterfactual explanations for graph neural networks. In: International conference on artificial intelligence and statistics, AISTATS 2022, virtual event
Lundberg SM, Lee S (2017) A unified approach to interpreting model predictions. In: Advances in neural information processing systems 30: annual conference on neural information processing systems 2017, Long Beach, CA, USA
Luss R, Chen P, Dhurandhar A, et al (2019) Generating contrastive explanations with monotonic attribute functions. arXiv preprint arXiv:1905.12698
Luss R, Chen P, Dhurandhar A, et al (2021) Leveraging latent features for local explanations. In: KDD '21: the 27th ACM SIGKDD conference on knowledge discovery and data mining, virtual event, Singapore
Madaan N, Padhi I, Panwar N, et al (2021) Generate your counterfactuals: towards controlled counterfactual generation for text. In: Thirty-fifth AAAI conference on artificial intelligence, AAAI 2021, thirty-third conference on innovative applications of artificial intelligence, IAAI 2021, the eleventh symposium on educational advances in artificial intelligence, EAAI 2021, virtual event
Martens D, Provost FJ (2014) Explaining data-driven document classifications. MIS Q
Martens D, Baesens B, Gestel TV, et al (2007) Comprehensible credit scoring models using rule extraction from support vector machines. Eur J Oper Res
Miller T (2019) Explanation in artificial intelligence: insights from the social sciences. Artif Intell
Ming Y, Qu H, Bertini E (2019) Rulematrix: visualizing and understanding classifiers with rules. IEEE Trans Vis Comput Graph
Mollas I, Bassiliades N, Tsoumakas G (2019) Lionets: local interpretation of neural networks through penultimate layer decoding. In: Machine learning and knowledge discovery in databases—international workshops of ECML PKDD 2019, Würzburg, Germany, proceedings, part I
Molnar C (2022) Model-agnostic interpretable machine learning. PhD thesis, Ludwig Maximilian University of Munich, Germany
Mothilal RK, Sharma A, Tan C (2020) Explaining machine learning classifiers through diverse counterfactual explanations. In: FAT* '20: conference on fairness, accountability, and transparency, Barcelona, Spain
Muhammad MB, Yeasin M (2020) Eigen-cam: Class activation map using principal components. In: 2020 International joint conference on neural networks, IJCNN 2020, Glasgow, UK
Murdoch WJ, Singh C, Kumbier K et al (2019) Definitions, methods, and applications in interpretable machine learning. Proc Natl Acad Sci 116(44):22071–22080
Nauta M, van Bree R, Seifert C (2021) Neural prototype trees for interpretable fine-grained image recognition. In: CVPR. Computer vision foundation/IEEE, pp 14933–14943

Nori H, Jenkins S, Koch P, et al (2019) Interpretml: a unified framework for machine learning interpretability. arXiv preprint arXiv:1909.09223

Pan D, Li X, Zhu D (2021) Explaining deep neural network models with adversarial gradient integration. In: Proceedings of the thirtieth international joint conference on artificial intelligence, IJCAI 2021, virtual event/Montreal, Canada

Panigutti C, Perotti A, Pedreschi D (2020) Doctor XAI: an ontology-based approach to black-box sequential data classification explanations. In: FAT* '20: conference on fairness, accountability, and transparency, Barcelona, Spain

Panigutti C, Beretta A, Giannotti F, et al (2022) Understanding the impact of explanations on advice-taking: a user study for ai-based clinical decision support systems. In: CHI '22: CHI conference on human factors in computing systems, New Orleans, LA, USA

Pasquale F (2015) The black box society: the secret algorithms that control money and information. Harvard University Press

Pawelczyk M, Broelemann K, Kasneci G (2020) Learning model-agnostic counterfactual explanations for tabular data. In: WWW '20: the web conference 2020, Taipei, Taiwan

Peltola T (2018) Local interpretable model-agnostic explanations of Bayesian predictive models via Kullback–Leibler projections. arXiv preprint arXiv:1810.02678

Petsiuk V, Das A, Saenko K (2018) RISE: randomized input sampling for explanation of black-box models. In: British machine vision conference 2018, BMVC 2018, Newcastle, UK

Pezeshkpour P, Tian Y, Singh S (2019) Investigating robustness and interpretability of link prediction via adversarial modifications. In: 1st Conference on automated knowledge base construction, AKBC 2019, Amherst, MA, USA

Plumb G, Molitor D, Talwalkar A (2018) Model agnostic supervised local explanations. In: Advances in neural information processing systems 31: annual conference on neural information processing systems 2018, NeurIPS 2018, Montréal, Canada

Poyiadzi R, Sokol K, Santos-Rodríguez R, et al (2020) FACE: feasible and actionable counterfactual explanations. In: AIES '20: AAAI/ACM conference on AI, ethics, and society, New York, NY, USA

Prado-Romero MA, Prenkaj B, Stilo G, et al (2022) A survey on graph counterfactual explanations: definitions, methods, evaluation. arXiv preprint arXiv:2210.12089

Puri I, Dhurandhar A, Pedapati T, et al (2021) Cofrnets: interpretable neural architecture inspired by continued fractions. In: Advances in neural information processing systems 34: annual conference on neural information processing systems 2021, NeurIPS 2021, virtual

Rajani NF, McCann B, Xiong C, et al (2019) Explain yourself! Leveraging language models for commonsense reasoning. In: Proceedings of the 57th conference of the association for computational linguistics, ACL 2019, Florence, Italy, vol 1: long papers

Renard X, Woloszko N, Aigrain J, et al (2019) Concept tree: high-level representation of variables for more interpretable surrogate decision trees. arXiv preprint arXiv:1906.01297

Ribeiro MT, Singh S, Guestrin C (2016) "Why should I trust you?": Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco, CA, USA

Ribeiro MT, Singh S, Guestrin C (2018) Anchors: High-precision model-agnostic explanations. In: Proceedings of the thirty-second AAAI conference on artificial intelligence, (AAAI-18), the 30th innovative applications of artificial intelligence (IAAI-18), and the 8th AAAI symposium on educational advances in artificial intelligence (EAAI-18), New Orleans, Louisiana, USA

Robnik-Šikonja M, Kononenko I (2008) Explaining classifications for individual instances. IEEE Trans Knowl Data Eng 20(5)

Rojat T, Puget R, Filliat D, et al (2021) Explainable artificial intelligence (XAI) on timeseries data: a survey. arXiv preprint arXiv:2104.00950

Rudin C (2019) Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat Mach Intell

Samek W, Montavon G, Vedaldi A, et al (eds) (2019) Explainable AI: interpreting, explaining and visualizing deep learning, lecture notes in computer science, vol 11700. Springer

Schwab P, Karlen W (2019) Cxplain: causal explanations for model interpretation under uncertainty. In: Advances in neural information processing systems 32: annual conference on neural information processing systems 2019, NeurIPS 2019, Vancouver, BC, Canada

Schwarzenberg R, Hübner M, Harbecke D, et al (2019) Layerwise relevance visualization in convolutional text graph classifiers. In: Proceedings of the thirteenth workshop on graph-based methods for natural language processing, TextGraphs@EMNLP 2019, Hong Kong

Selvaraju RR, Cogswell M, Das A, et al (2020) Grad-cam: Visual explanations from deep networks via gradient-based localization. Int J Comput Vis

Setzu M, Guidotti R, Monreale A, et al (2019) Global explanations with local scoring. In: Machine learning and knowledge discovery in databases—international workshops of ECML PKDD 2019, Würzburg, Germany, proceedings, part I

Setzu M, Guidotti R, Monreale A, et al (2021) Glocalx—from local to global explanations of black box AI models. Artif Intell

Shankaranarayana SM, Runje D (2019) ALIME: autoencoder based approach for local interpretability. In: Intelligent data engineering and automated learning—IDEAL 2019—20th international conference, Manchester, UK, proceedings, part I

Shen W, Wei Z, Huang S, et al (2021) Interpretable compositional convolutional neural networks. In: Proceedings of the thirtieth international joint conference on artificial intelligence, IJCAI 2021, virtual event/Montreal, Canada

Shi S, Zhang X, Fan W (2020) A modified perturbed sampling method for local interpretable model-agnostic explanation. arXiv preprint arXiv:2002.07434

Shrikumar A, Greenside P, Kundaje A (2017) Learning important features through propagating activation differences. In: Proceedings of the 34th international conference on machine learning, ICML 2017, Sydney, NSW

Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: 3rd International conference on learning representations, ICLR 2015, San Diego, CA, USA, Conference track proceedings

Smilkov D, Thorat N, Kim B, et al (2017) Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825

Snyder H (2019) Literature review as a research methodology: an overview and guidelines. J Bus Res 104:333–339. https://doi.org/10.1016/j.jbusres.2019.07.039

Srivastava S, Labutov I, Mitchell TM (2017) Joint concept learning and semantic parsing from natural language explanations. In: Proceedings of the 2017 conference on empirical methods in natural language processing, EMNLP 2017, Copenhagen, Denmark

Suissa-Peleg A, Haehn D, Knowles-Barley S, et al (2016) Automatic neural reconstruction from petavoxel of electron microscopy data. Microsc Microanal

Sundararajan M, Taly A, Yan Q (2017) Axiomatic attribution for deep networks. In: Proceedings of the 34th international conference on machine learning, ICML 2017, Sydney, NSW, Australia

Tan S, Soloviev M, Hooker G, et al (2020) Tree space prototypes: another look at making tree ensembles interpretable. In: FODS '20: ACM-IMS foundations of data science conference, virtual event, USA

Theissler A (2017) Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection. Knowl Based Syst

Theissler A, Spinnato F, Schlegel U, et al (2022) Explainable AI for time series classification: a review, taxonomy and research directions. IEEE Access

Tjoa E, Guan C (2019) A survey on explainable artificial intelligence (XAI): towards medical XAI. arXiv preprint arXiv:1907.07374

Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. In: Advances in neural information processing systems 30: annual conference on neural information processing systems 2017, Long Beach, CA, USA

Verma S, Dickerson JP, Hines K (2020) Counterfactual explanations for machine learning: a review. arXiv preprint arXiv:2010.10596

Vermeire T, Brughmans D, Goethals S et al (2022) Explainable image classification with evidence counterfactual. Pattern Anal Appl 25(2):315–335

Wachter S, Mittelstadt BD, Russell C (2017) Counterfactual explanations without opening the black box: automated decisions and the GDPR. arXiv preprint arXiv:1711.00399

Wang H, Wang Z, Du M, et al (2020) Score-cam: score-weighted visual explanations for convolutional neural networks. In: 2020 IEEE/CVF conference on computer vision and pattern recognition, CVPR workshops 2020, Seattle, WA, USA

Williams JJ, Kim J, Rafferty AN, et al (2016) AXIS: generating explanations at scale with learnersourcing and machine learning. In: Proceedings of the third ACM conference on learning @ Scale, L@S 2016, Edinburgh, Scotland, UK

Wu Z, Ong DC (2021) Context-guided BERT for targeted aspect-based sentiment analysis. In: Thirty-fifth AAAI conference on artificial intelligence, AAAI 2021, thirty-third conference on innovative applications of artificial intelligence, IAAI 2021, the eleventh symposium on educational advances in artificial intelligence, EAAI 2021, virtual event

Wu T, Ribeiro MT, Heer J, et al (2021a) Polyjuice: generating counterfactuals for explaining, evaluating, and improving models. In: Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing, ACL/IJCNLP 2021, (vol 1: long papers), virtual event

Wu Z, Pan S, Chen F, et al (2021b) A comprehensive survey on graph neural networks. IEEE Trans Neural Networks Learn Syst

Xu K, Ba J, Kiros R, et al (2015) Show, attend and tell: neural image caption generation with visual attention. In: Proceedings of the 32nd international conference on machine learning, ICML 2015, Lille, France

Yang M, Kim B (2019) BIM: towards quantitative evaluation of interpretability methods with ground truth. arXiv preprint arXiv:1907.09701

Yang H, Rudin C, Seltzer MI (2017) Scalable Bayesian rule lists. In: Proceedings of the 34th international conference on machine learning, ICML 2017, Sydney, NSW, Australia

Yeh C, Kim B, Arik SÖ, et al (2020) On completeness-aware concept-based explanations in deep neural networks. In: Advances in neural information processing systems 33: annual conference on neural information processing systems 2020, NeurIPS 2020, virtual

Yuan H, Tang J, Hu X, et al (2020a) XGNN: towards model-level explanations of graph neural networks. In: KDD '20: the 26th ACM SIGKDD conference on knowledge discovery and data mining, virtual event, CA, USA, August (2020)

Yuan H, Yu H, Gui S, et al (2020b) Explainability in graph neural networks: a taxonomic survey. arXiv preprint arXiv:2012.15445

Yuan H, Yu H, Gui S, et al (2020c) Explainability in graph neural networks: a taxonomic survey. arXiv preprint arXiv:2012.15445

Zafar MR, Khan NM (2019) DLIME: a deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems. arXiv preprint arXiv:1906.10263

Zhang Y, Chen X (2020) Explainable recommendation: a survey and new perspectives. Found Trends Inf Retr

Zhang H, Torres F, Sicre R, et al (2023) Opti-cam: optimizing saliency maps for interpretability. CoRR arXiv:2301.07002

Zhou Y, Hooker G (2016) Interpreting models via single tree approximation. arXiv preprint arXiv:1610.09036