A Survey of Methods for Explaining Black Box Models

RICCARDO GUIDOTTI, ANNA MONREALE, SALVATORE RUGGIERI, and
FRANCO TURINI, KDDLab, University of Pisa, Italy
FOSCA GIANNOTTI, KDDLab, ISTI-CNR, Italy
DINO PEDRESCHI, KDDLab, University of Pisa, Italy

In recent years, many accurate decision support systems have been constructed as black boxes, that is as systems that hide their internal logic to the user. This lack of explanation constitutes both a practical and an ethical issue. The literature reports many approaches aimed at overcoming this crucial weakness, sometimes at the cost of sacrificing accuracy for interpretability. The applications in which black box decision systems can be used are various, and each approach is typically developed to provide a solution for a specific problem and, as a consequence, it explicitly or implicitly delineates its own definition of interpretability and explanation. The aim of this article is to provide a classification of the main problems addressed in the literature with respect to the notion of explanation and the type of black box system. Given a problem definition, a black box type, and a desired explanation, this survey should help the researcher to find the proposals more useful for his own work. The proposed classification of approaches to open black box models should also be useful for putting the many research open questions in perspective.

CCS Concepts: • **Information systems** → **Decision support systems**; **Data analytics**; **Data mining**;

Additional Key Words and Phrases: Open the black box, explanations, interpretability, transparent models

ACM Reference format:
Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv.* 51, 5, Article 93 (August 2018), 42 pages.
https://doi.org/10.1145/3236009

1 INTRODUCTION

The past decade has witnessed the rise of ubiquitous opaque decision systems. These black box systems exploit sophisticated machine-learning models to predict individual information that may also be sensitive. We can consider credit score, insurance risk, health status, as examples. Machine-learning algorithms build predictive models that are able to map user features into a class (outcome or decision) thanks to a learning phase. This learning process is made possible by the digital traces that people leave behind them while performing everyday activities (e.g., movements, purchases,

This work is partially supported by the European Community's H2020 Program under the funding scheme "INFRAIA-1-2014-2015: Research Infrastructures," Grant Agreement No. 654024, *SoBigData*, http://www.sobigdata.eu.
Authors' addresses: R. Guidotti, A. Monreale, S. Ruggieri, and F. Turini, KDDLab, University of Pisa, Largo Bruno Pontecorvo, 3, Pisa, PI, 56127, Italy; emails: {riccardo.guidotti, anna.monreale, salvatore.ruggieri, franco.turini}@di.unipi.it; F. Giannotti, KDDLab, ISTI-CNR, Via Moruzzi, 1, Pisa, PI, 56127, Italy; email: fosca.giannotti@isti.cnr.it; D. Pedreschi, KDDLab, University of Pisa, Largo Bruno Pontecorvo, 3, Pisa, PI, 56127, Italy; email: dino.pedreschi@di.unipi.it.

ACM 0360-0300/2018/08-ART93 $15.00
https://doi.org/10.1145/3236009

comments in social networks, etc.). This enormous amount of data may contain human biases and prejudices. Thus, decision models learned on them may inherit such biases, possibly leading to unfair and wrong decisions.

The European Parliament recently adopted the *General Data Protection Regulation (GDPR)*, which has become law in May 2018. An innovative aspect of the GDPR are the clauses on automated decision-making, including profiling, which for the first time introduce, to some extent, a right of explanation for all individuals to obtain "meaningful explanations of the logic involved" when automated decision making takes place. Despite divergent opinions among legal scholars regarding the real scope of these clauses [36, 74, 126], there is a general agreement on the need for the implementation of such a principle is urgent and that it represents today a huge open scientific challenge. Without an enabling technology capable of explaining the logic of black boxes, the right to an explanation will remain a "dead letter."

By relying on sophisticated machine-learning classification models trained on massive datasets thanks to scalable, high-performance infrastructures, we risk to create and use decision systems that we do not really understand. This impacts not only information on ethics but also on accountability [59], on safety [23], and on industrial liability [53]. Companies increasingly release market services and products by embedding data mining and machine-learning components, often in safety-critical industries such as self-driving cars, robotic assistants, and personalized medicine.

Another inherent risk of these components is the possibility of inadvertently making wrong decisions, learned from artifacts or spurious correlations in the training data, such as recognizing an object in a picture by the properties of the background or lighting, due to a systematic bias in training data collection. How can companies trust their products without understanding and validating the underlying rationale of their machine-learning components? Gartner predicts that "by 2018 half of business ethics violations will occur through the improper use of Big Data analytics." Explanation technologies are an immense help to companies for creating safer, more trustable products, and better managing any possible liability they may have. This is especially important in safety critical applications like self-driving cars and medicine, where a possible wrong decision could even lead to the death of people. The availability of transparent machine-learning technologies would lead to a gain of trust and awareness on the fact that it is always possible to know the reasons of a decision or an event. For example, this kind of guaranty would have been useful in the recent case of the incident that involved a self-driving Uber car that knocked down and killed a pedestrian in Tempe, Arizona on March 2018.[1] In particular, the use of interpretable models in this case would have helped Uber and Waymo in understanding the reason behind the decision and in managing their responsibilities. Likewise, the use of machine-learning models in scientific research, for example, in medicine, biology, socio-economic sciences, requires an explanation not only for trust and acceptance of results but also for the sake of the openness of scientific discovery and the progress of research.

As a consequence, explanation is at the heart of a responsible, open data science, across multiple industry sectors and scientific disciplines. Different scientific communities studied the problem of explaining machine-learning decision models. However, each community addresses the problem from a different perspective and provides a different meaning to *explanation*. Most of the works in the literature come from the machine-learning and data-mining communities. The first one is mostly focused on describing how black boxes work, while the second one is more interested in explaining the decisions even without understanding the details on how the opaque decision systems work in general.

[1] https://www.nytimes.com/2018/03/19/technology/uber-driverless-fatality.html.

Despite the fact that interpretable machine learning has been a topic for quite some time and received recently much attention, today there are many ad hoc scattered results, and a systematic organization and classification of these methodologies is missing. Many questions feed the papers in the literature proposing methodologies for interpreting black box systems [41, 132]: *What does it mean that a model is interpretable or transparent? What is an explanation? When a model or an explanation is comprehensible? Which is the best way to provide an explanation? Which are the problems requiring interpretable models/predictions? What kind of decision data are affected? Which type of data records is more comprehensible? How much are we willing to lose in prediction accuracy to gain any form of interpretability?*

We believe that a clear classification considering all these aspects simultaneously is needed to organize the body of knowledge about research investigating methodologies for opening and understanding the black box. Existing works tend to provide just a general overview of the problem [68] highlighting unanswered questions and problems [28]. However, other works focus on particular aspects like the impact of representation formats on comprehensibility [43] or the interpretability issues in term of advantages and disadvantages of selected predictive models [32]. Consequently, after recognizing four categories of problems and a set of ways to provide an explanation, we have chosen to group the methodologies for opening and understanding black box predictors by considering simultaneously the problem they are facing, the class of solutions proposed for the explanation, the kind of data analyzed and the type of predictor explained.

The rest of the article is organized as follows. First, Section 2 shows which are the motivations for requiring explanation for black box systems by illustrating some real cases. In Section 3, we discuss what interpretability is. In Section 4, we formalize our problem definitions used to categorize the state of the art works. Details of the classification and crucial points distinguishing the various approaches and papers are discussed in Section 5. Sections 6, 7, 8, and 9 present the details of the solutions proposed. Finally, Section 10 summarizes the crucial aspects that emerged from the analysis of the state of the art and discusses which are the open research questions and future research directions.

2 NEEDS FOR INTERPRETABLE MODELS

Which are the real problems requiring interpretable models and explainable predictions? In this section, we briefly report some cases showing how and why black boxes can be dangerous. Indeed, delegating decisions to black boxes without the possibility of an interpretation may be critical, can create discrimination and trust issues.

Training a classifier on historical datasets, reporting human decisions, could lead to the discovery of endemic preconceptions [88]. Moreover, since these rules can be deeply concealed within the trained classifier, we risk to consider, maybe unconsciously, such practices and prejudices as general rules. We are warned about a growing "black box society" [86], governed by "secret algorithms protected by industrial secrecy, legal protections, obfuscation, so that intentional or unintentional discrimination becomes invisible and mitigation becomes impossible."

Automated discrimination is not new and is not necessarily due to "black box" models. A computer program for screening job applicants were used during the 1970s and 1980s in St. George's Hospital Medical School, London, UK. The program used information from applicants' forms, without any reference to ethnicity. However, the program was found to unfairly discriminate against ethnic minorities and women by inferring this information from surnames and place of birth, and lowering their chances of being selected for interview [71]. More recently, the journalists of *propublica.org* have shown that the COMPAS score, a predictive model for the "risk of crime recidivism" (proprietary secret of Northpointe), has a strong ethnic bias. Indeed, according to this score, a black who did not re-offend was classified as high risk twice as much as whites who did not re-offend,

Fig. 1. Examples of possible military tank misclassification depending on the background: sunny (left) and overcast (right).

and white repeat offenders were classified as low risk twice as much as black repeat offenders.[2] Similarly, a study at Princeton [14] shows how text and web corpora contain human biases: names that are associated with black people are found to be significantly more associated with unpleasant terms than with pleasant terms, compared to names associated with whites. As a consequence, the models learned on such text data for opinion or sentiment mining have a possibility of inheriting the prejudices reflected in the data.

Another example is related to Amazon.com. In 2016, the software used to determine the areas of the U.S. to which Amazon would offer free same-day delivery, unintentionally restricted minority neighborhoods from participating in the program (often when every surrounding neighborhood was allowed).[3] With respect to credit bureaus, it is shown in Reference [15] that banks providing credit scoring for millions of individuals, are often discordant: in a study of 500,000 records, 29% of consumers received credit scores that differed by at least 50 points among three major U.S. banks (Experian, TransUnion, and Equifax). Such a difference might mean tens of thousands of dollars over the life of a mortgage. So much variability implies that the three scoring systems either have a very different and undisclosed bias or are highly arbitrary. As an example of bias, we can consider References [32] and [98]. In these works, the authors show how accurate black box classifiers may result from an accidental artifact in the training data. In Reference [32] author discuss the application of a back box classifier in a military context. The military trained a classifier to recognize enemy tanks from friendly tanks. The classifier resulted in a high accuracy on the test set, but when it was used in the field had very poor performance. Later was discovered that friendly photos were taken on sunny days, while enemy photos on overcast days (see Figure 1 for an example). As the authors state, it is not clear if this story is based on a real or hypothetical application of data mining and machine-learning algorithms. Similarly, in Reference [98] it is shown that a classifier trained to recognize wolves and husky dogs were basing its predictions to classify a wolf solely on the presence of snow in the background.

Nowadays, *Deep Neural Networks (DNNs)* have been reaching very good performances on different pattern-recognition tasks, such as visual and text classification, which are easily performed by humans: e.g., saying that a tomato is displaced in a picture or that a text is about a certain topic. Thus, what differences remain between DNNs and humans? Despite the excellent performance of DNNs it seems to be a lot. In Reference [116] it is shown the alteration of an image (e.g., of a tomato) such that the change is undetectable for humans can lead a DNN to tag the image as something else (e.g., labeling a tomato as a dog). In Reference [81] a related result is shown. It is easy to produce images that DNNs believe to be recognizable with 99.99% confidence but that are completely unrecognizable to humans (e.g., labeling white static noise as a tomato). Similarly, in Reference [54] visually indistinguishable training-sets are created using DNNs and linear models. With respect to text, in Reference [67] effective methods to attack DNN text classifiers are presented. Experiments

[2] http://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.
[3] http://www.techinsider.io/how-algorithms-can-be-racist-2016-4.

show that the perturbations introduced in the text are difficult to be perceived by a human but are still able to fool a state-of-the-art DNN to misclassify a text as any desirable class. These results show interesting differences between humans and DNNs and raise reasonable doubts about trusting such black boxes. In Reference [138] it is shown how conventional regularization and small generalization error fail to explain why DNNs generalize well in practice. Specifically, they prove that established state-of-the-art CNN trained for image classification easily fits a random labeling of the training data. This phenomenon occurs even if the true images are replaced by unstructured random noise.

3 INTERPRETABLE, EXPLAINABLE, AND COMPREHENSIBLE MODELS

Before presenting the classification of the problems addressed in the literature with respect to black box predictors, and the corresponding solutions and models categorization, it is crucial to understand the meaning of *black box predictor* and *interpretability*. Thus, in this section, we discuss what an interpretable model is, and we analyze the various dimensions of interpretability as well as the desiderata for an interpretable model. Moreover, we also discuss the meaning of words like *interpretability*, *explainability*, and *comprehensibility*, which are largely used in the literature.

A *black box predictor* is a data-mining and machine-learning obscure model, whose internals are either unknown to the observer or they are known but *uninterpretable* by humans. To *interpret* means to give or provide the meaning or to explain and present in understandable terms some concepts.[4] Therefore, in data mining and machine learning, *interpretability* is defined as the ability to explain or to provide the meaning in understandable terms to a human [28]. These definitions assume implicitly that the concepts expressed in the understandable terms composing an explanation are self-contained and do not need further explanations. Essentially, an explanation is an "interface" between humans and a decision maker that is at the same time both an accurate proxy of the decision maker and comprehensible to humans.

As shown in the previous section, another significant aspect to mention about interpretability is the reason why a system, a service or a method should be interpretable. However, an explanation could be not required if there are no decisions that have to be made on the outcome of the prediction. For example, if we want to know if an image contains a cat or not and this information is not required to take any sort of crucial decision, or there are no consequences for unacceptable results, then we do not need an interpretable model, and we can accept any black box.

Finally, another relevant aspect is the "reason" why an explanation is necessary: an interpretable model can be required either to reveal findings in data that explain the decision, or to explain how the black box itself works. These equally important and complementary reasons demand different analysis methods. To better explain this concept, we report the following example from Reference [39]. For a surgeon who needs to decide where to remove brain tissue it is most important to know the origin of cognitive functions and associated neural processes. However, when communicating with paralyzed patients via brain-computer interfaces, it is most important to accurately extract the neural processes specific to a certain mental state. Therefore, the works presented in this survey may have two flavors: a more applicative nature aimed at explaining why a certain decision have been returned for a particular input, or a more theoretical nature aimed at explaining the logic behind the whole obscure model.

3.1 Dimensions of Interpretability

In the analysis of the interpretability of predictive models, we can identify a set of dimensions to be taken into consideration, and that characterize the interpretability of the model [28].

[4] https://www.merriam-webster.com/.

Global and Local Interpretability: A model may be completely interpretable, i.e., we are able to understand the whole logic of a model and follow the entire reasoning leading to all the different possible outcomes. In this case, we are speaking about *global* interpretability. Instead, we indicate with *local* interpretability the situation in which it is possible to understand only the reasons for a specific decision: only the single prediction/decision is interpretable.

Time Limitation: An important aspect is the time that the user is available or is allowed to spend on understanding an explanation. The user time availability is strictly related to the scenario where the predictive model has to be used. Therefore, in some contexts where the user needs to quickly take the decision (e.g., a disaster is imminent), it is preferable to have an explanation simple to understand. While in contexts where the decision time is not a constraint (e.g., during a procedure to release a loan) one might prefer a more complex and exhaustive explanation.

Nature of User Expertise: Users of a predictive model may have different background knowledge and experience in the task: decision-makers, scientists, compliance and safety engineers, data scientists, and so on. Knowing the user experience in the task is a key aspect of the perception of interpretability of a model. Domain experts may prefer a larger and more sophisticated model over a smaller and sometimes more opaque one.

The works reviewed in the literature only implicitly specify if their proposal is global or local. Just a few of them take into account the nature of user expertise [34, 98, 104], and only few of them provide real experiments about the time required to understand an explanation [61, 129]. Instead, some of the works consider the "complexity" of an explanation through an approximation. For example, they define the model complexity as the model's size (e.g., tree depth, number of rules, etc.) [25, 38, 47, 98]. In the following, we further discuss issues related to the complexity of an explanation.

3.2 Desiderata of an Interpretable Model

An interpretable model is required to provide an explanation. Thus, to realize an interpretable model, it is necessary to take into account the following list of desiderata, which are mentioned by a set of papers in the state of art [5, 28, 32, 45]:

- *Interpretability*: to which extent the model and/or its predictions are human understandable. The most addressed discussion is related to how the interpretability can be measured. In Reference [32] a component for measuring the interpretability is the *complexity* of the predictive model in terms of the model size. According to the literature, we refer to interpretability also with the name *comprehensibility*.
- *Accuracy*: to which extent the model accurately predicts unseen instances. The accuracy of a model can be measured using evaluation measures like the accuracy score, the F1-score [118], and so on. Producing an interpretable model maintaining competitive levels of accuracy is the most common target among the papers in the literature.
- *Fidelity*: to which extent the model is able to accurately *imitate* a black-box predictor. The fidelity captures how much is good an interpretable model in the mimic of the behavior of a black-box. Similarly to the accuracy, the fidelity is measured in terms of accuracy score, F1-score, and so on, but with respect to the outcome of the black box.

Moreover, besides these features are strictly related to interpretability, yet according to [5, 28, 32, 45] a data-mining and machine-learning model should have other important desiderata. Some of these desiderata are related to ethical aspects such as *fairness* and *privacy*. The first principle requires that the model guarantees the protection of groups against (direct or indirect) discrimination [100]; while the second one requires that the model does not reveal sensitive information about people [4]. The level of interpretability of a model together with the standards of privacy and

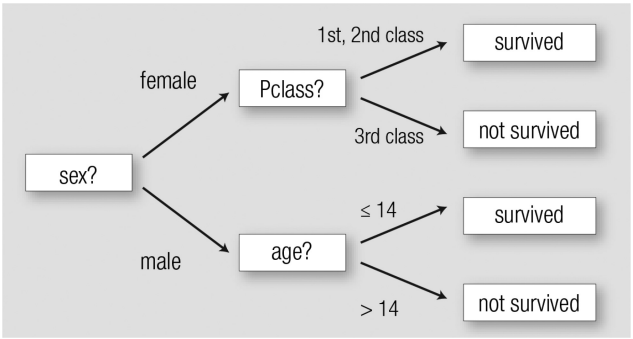

Fig. 2. Example of decision tree classifier.

non-discrimination that are guaranteed may impact on how much human users trust that model. The degree of trust on a model increases if the model is built by respecting constraints of *monotonicity* given by the users [77, 87, 123]. A predictor respecting the *monotonicity* principle is, for example, a predictor where the increase of the values of a numerical attribute tends to either increase or decrease in a monotonic way the probability of a record of being member of a class [32]. Another property that influences the trust level of a model is *usability*: people tend to trust more models providing information that assist them to accomplish a task with awareness. In this line, an interactive and queryable explanation results to be more usable than a textual and fixed explanation.

Furthermore, data mining and machine-learning models should also have other ordinary important required features such as *reliability*, *robustness*, *causality*, *scalability*, and *generality*. This means that a model should have the ability to maintain certain levels of performance independently from small variations of the parameters or of the input data (*reliability*/*robustness*) and that controlled changes in the input due to a perturbation affect the model behavior (*causality*). Moreover, since we are in the "Big Data era," it is opportune to have models able to *scale* to large input data with large input spaces. Finally, since often in different application scenarios one might use the same model with different data, it is preferable to have portable models that do not require special training regimes or restrictions (*generality*).

3.3 Recognized Interpretable Models

In the state of the art a small set of existing interpretable models is recognized: *decision tree*, *rules*, *linear models* [32, 43, 98]. These models are considered easily understandable and interpretable for humans.

A decision system based on a *decision tree* exploits a graph structured like a tree and composed of internal nodes representing tests on features or attributes (e.g., whether a variable has a value lower than, equals to or grater than a threshold, see Figure 2) and leaf nodes representing a class label. Each branch represents a possible outcome [92]. The paths from the root to the leaves represent the classification rules. Indeed, a decision tree can be linearized into a set of decision rules with the *if-then* form [31, 90, 91]:

$$\text{if } condition_1 \wedge condition_2 \wedge condition_3, \text{ then } outcome.$$

Here, the outcome corresponds to the class label of a leaf node while the conjunctions of conditions in the if clause correspond to the different conditions in the path from the root to that leaf node.

More generally, a *decision rule* is a function that maps an observation to an appropriate action. Decision rules can be extracted by generating the so-called *classification rules*, i.e., association rules that in the consequence have the class label [3]. The most common rules are *if-then rules* where the

if clause is a combination of conditions on the input variables. In particular, it may be formed by conjunctions, negations and disjunctions. However, methods for rule extraction typically take into consideration only rules with conjunctions. Other types of rules are: *m-of-n rules*, where given a set of n conditions, if m of them are verified, then the consequence of the rule is considered true [79]; *list of rules*, where given an ordered set of rules is considered true, the consequent of the first rule that is verified [135]; *falling rule lists* consists of a list of if-then rules ordered with respect to the probability of a specific outcome, and the order identifies the example to be classified by that rule [127]; *decision sets*, where an unordered set of classification rules is provided such that the rules are not connected by else statements, but each rule is an independent classifier that can assign its label without regard for any other rules [61].

The interpretation of rules and decision trees is different with respect to different aspects [32]. Decision trees are widely adopted for their graphical representation, while rules have a textual representation. The main difference is that textual representation does not provide immediately information about the more relevant attributes of a rule. However, the hierarchical position of the features in a tree gives this kind of clue.

Attributes' relative importance could be added to rules by means of positional information. Specifically, rule conditions are shown by following the order in which the rule extraction algorithm added them to the rule. Even though the representation of rules causes some difficulties in understanding the whole model, it enables the study of single rules representing partial parts of the whole knowledge ("local patterns"), which are composable. Also in a decision tree, the analysis of each path separately from the leaf node to the root, enables users to focus on such local patterns. However, if the tree is very deep in this case it is a much more complex task.

A further crucial difference between rules and decision trees is that in a decision tree each record is classified by only one leaf node, i.e., the class predicted are represented in a mutually exclusive and exhaustive way by the set of leaves and their paths to the root node. However, a certain record can satisfy the antecedent of rules having as consequent a different class for that record. Indeed, rule-based classifiers have the disadvantage of requiring an additional approach for resolving such situations of conflicting outcome [133]. Many rule-based classifiers deal with this issue by returning an ordered rule list, instead of an unordered rule set. In this way it is returned the outcome corresponding to the first rule matching the test record and ignoring the other rules in the list. We notice that ordered rule lists may be harder to interpret than classical rules. In fact, in this model a given rule cannot be considered independently from the precedent rules in the list [133]. Another widely used approach consists in considering the top-k rules satisfying the test record where the ordering is given by a certain weight (e.g., Laplace accuracy). Then, the outcome of the rules with the average highest weight among the top-k is returned as predicted class [135].

Another set of approaches adopted to provide explanations are *linear models* [29, 39, 98]. This can be done by considering and visualizing the *features importance*, i.e., both the sign and the magnitude of the contribution of the attributes for a given prediction (see Figure 3). If the contribution of an attribute-value is positive, then it contributes by increasing the model's output. Instead, if the sign is negative then the attribute-value decreases the output of the model. If an attribute-value has an higher contribution than another, then it means that it has an higher influence on the prediction of the model. The produced contributions summarize the performance of the model, thus the difference between the predictions of the model and expected predictions, providing the opportunity of quantifying the changes of the model prediction for each test record. In particular, it is possible to identify the attributes leading to this change and for each attribute how much it contributed to the change. An intrinsic problem that linear models have when used for explanation is that when the model does not optimally fit the training data, it may use spurious features to optimize the error, and these features may be very hard to interpret for a human.

Fig. 3. Example of linear model and the returned features importance.

We point out that, in general, when an explanation for a prediction is provided, besides the explanation (satisfied rules, branch of the tree, features importance, etc.), it is often useful to analyze also instances that are exceptions with respect to the "boundaries" provided by the explanation, or with very few differences with respect to the prototypes returned as explanation. For example, instances covered by the rule body but with an outcome label different from the class of the outcome predicted. Even though this sort of *exception analysis* is hardly performed, it can be more informative than the direct explanation, and it can also provide clues about the application domain [37, 85].

Finally, as last remark, we underline that all the aforementioned techniques for providing explanations are effectively interpretable only when they have human-reasonable sizes. Indeed, the goodness of the explanation could be invalidated by its size and complexity. For example, when the linear model is high-dimensional, the explanation may be overwhelming. Moreover, if a too large set of rules, or a too deep and wide tree are returned they could not be humanly manageable even though they are perfectly capturing the internal logic of the black box for the classification.

3.4 Explanations and Interpretable Models Complexity

In the literature, very little space is dedicated to a crucial aspect: the model complexity. The evaluation of the model complexity is generally tied to the model comprehensibility, and this is a very hard task to address. As a consequence, this evaluation is generally estimated with a rough approximation related to the *size* of the model. Moreover, complexity is often used as an opposed term to interpretability. Before analyzing the various notions of model complexity in the literature, we point out that, concerning the problem of black box explanation, the complexity is only related to the model and not to the training data that is generally unknown.

In Reference [38] the complexity is identified by the number of regions, i.e., the parts of the model, for which the boundaries are defined. In Reference [98] as complexity for linear models is adopted the number of non-zero weights, while for decision trees the depth of the tree. In Reference [25] the complexity of a rule (and thus of an explanation) is measured by the length of the rule condition, defined as the number of attribute-value pairs in the condition. Given two rules with similar frequency and accuracy, the rule with the smaller length may be preferred as it is more interpretable. Similarly, in the case of lists of rules the complexity is typically measured considering the total number of attribute-value pairs in the whole set of rules. However, this could not be a suitable way for measuring the model complexity, since in an ordered rule list different test records need distinct numbers of rules to be evaluated [32]. In this kind of model, a more honest measure could be the average number of conditions evaluated to classify a set of test records [84]. However, this is more a "measure of the explanation" of a list of rules, rather than a "measure of the complexity" of rule list's itself.

In the decision tree literature, the problems of over-fitting the training data and of "trading accuracy for simplicity" [10] have been addressed by a class of post-processing algorithms called

simplification methods [93]. The most well-known simplification method is decision tree *pruning*, which relies on an error estimation function to compare the errors of the two alternatives: to prune a subtree or not [101]. Such methods are mainly intended for improving model generality.

Differently from the not flexible representation of decision trees where the prediction of a single record is mutually exhaustive and exclusive, rules characterization contains only significant clauses. That is, a record can satisfy more than a rule also with different outcome labels, and there could be records that are not covered by any rule. Moreover, rules do not capture insignificant clauses, while decision trees can also have insignificant branches. This happens because rule-based classifier generally select one *attribute-value* while expanding a rule, whereas decision tree algorithms usually select one *attribute* while expanding the tree [32]. Considering these aspects to estimate the complexity is very difficult. Consequently, even though a model equivalence exists, the estimation of the fact that a different representation for the same model (or explanation) is more complex than another can be very subjective with respect to the interpreter.

3.5 Interpretable Data for Interpretable Models

The *types of data* used for classification may have diverse nature. Different types of data present a different level of interpretability for a human. The majority of data mining and machine-learning techniques work on data organized in *tables* that algorithms may handle as matrices. The advantage of this type of data is that it is both easily managed by these algorithms, without requiring specific transformations, and enough simple to be interpreted by humans [43]. Whereas the disadvantage of tables is that the interpretation of the represented information requires understanding also the meta-data that allow us to associate a meaning to values in the tables.

Other forms of data that are more understandable than tables are *images* and *texts*. They indeed represent the most common and natural way people use to communicate in everyday life, and they do not require the understanding of any meta-structure useful for deriving a specific meaning. However, the processing of these data for predictive models requires their transformation into vectors that make them easier to be processed by algorithms. Indeed, on images and texts, the state of art techniques typically apply predictive models based on support vector machine, neural networks or deep neural networks that are usually hard to be interpreted. As a consequence, certain recognized interpretable models cannot be directly employed for this type of data to obtain an interpretable model or a human understandable explanation. Transformations using equivalences, approximations or heuristics are required in such a way that images and texts can be employed by prediction systems and used for providing the interpretation of the model and/or the prediction at the same time.

Finally, there exist other forms of data such as sequences, spatio-temporal data and complex networks that may be used by data mining and machine-learning algorithms. However, to the best of our knowledge, in the literature there is no work addressing the black box model explanation for data different from images, texts, and tabular data. The only exceptions are [39, 113] that using EEG data provide heatmaps with data point's relevance for the decision's outcome.[5]

4 OPEN THE BLACK BOX PROBLEMS

An accurate analysis and review of the literature led to the identification of different categories of problems. At a very high level, we can distinguish between *reverse engineering* and *design* of explanations. In the first case, given the decision records produced by a black box decision maker the problem consists in reconstructing an explanation for it. The original dataset upon which the black box is trained is generally not known in real life. Reverse engineering is exploited to build

[5]In practice, being a sort of time series EEG data can be viewed as a particular type of tabular data.

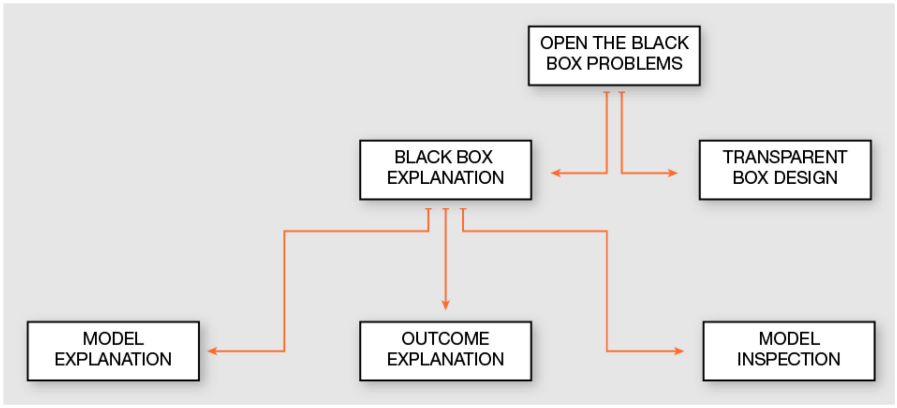

Fig. 4. Open the black box problems taxonomy. The *Open the Black Box Problems* for understanding how a black box works can be separated from one side as the problem of *explaining* how the decision system returned certain outcomes (*Black Box Explanation*) and on the other side as the problem of directly designing a *transparent* classifier that solves the same classification problem (*Transparent Box Design*). Moreover, the Black Box Explanation problem can be further divided among *Model Explanation* when the explanation involves the whole logic of the obscure classifier, *Outcome Explanation* when the target is to understand the reasons for the decisions on a given object, and *Model Inspection* when the target to understand how internally the black box behaves changing the input.

the explanations by most of the works presented in this survey. Details about reverse engineering approaches (also known as *post hoc interpretability* in the literature) are discussed at the end of this section. In the second case, given a dataset of training decision records the task consists in developing an interpretable predictor model together with its explanations. Through a deep analysis of the state of the art, we are able to further refine the first category into three different problems: *model explanation*, *outcome explanation*, and *model inspection*. We name the first macro-category *black box explanation problem* and the second one *transparent box design problem*. Figure 4 depicts a tree-structured diagram representing our categorization of the open the black box problems.

Recalling the concept of the "motivation" for having an explanation discussed in the previous section, the model explanation problem is aimed at understanding the overall logic behind the black box, while the outcome explanation problem is more related to the correlation between the data of a record and the outcome decision. Finally, the model inspection problem is somehow in the middle and depends on the motivation of the specific paper under analysis. All these problems can be seen as specific cases of the general classification problems with the common target of providing an interpretable and accurate predictive model. Details of the formalization are provided in the following sections.

Other important variants are generally not treated in the literature making the problem of discovering an explanation increasingly difficult: *(i)* Is it allowed to query the black box at will to obtain new decision examples, or only a fixed dataset of decision records is available? *(ii)* Is the complete set of features used by the decision model known, or instead only part of these features is known? Said in other terms, there are any co-founding factors? In this survey, we do not address these issues as in the literature there is not sufficient material.

4.1 Problem Formulation

In the following, we generalize the classification problem (see Figure 5). A *predictor*, also named model or classifier, is a function $b : \mathcal{X}^{(m)} \rightarrow \mathcal{Y}$, which maps data instances (tuples) x from a feature

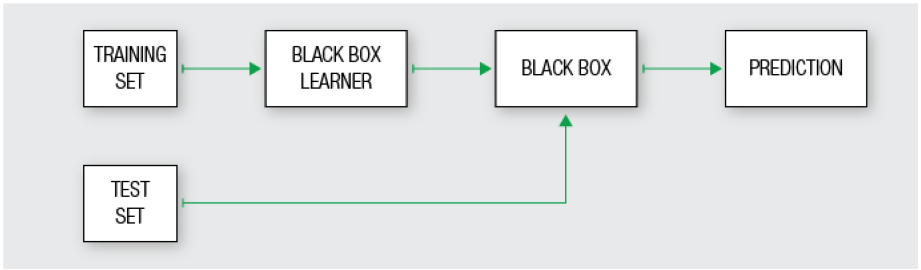

Fig. 5. Classification problem: given a *train* D_{train} and a *test* D_{test} set the black box model is learned on the train (*black box learner*) generating the black box predictor b that can applied on the test set D_{test} to obtain the prediction $Y = \bigcup_{x \in D_{test}} b(x)$.

space $\mathcal{X}^{(m)}$ with m input features to a decision y in a target space $\mathcal{Y}$. We write $b(x) = y$ to denote the decision y predicted by b, and $b(X) = Y$ as a shorthand for $\{b(x) \mid x \in X\} = Y$. An instance x consists of a set of m attribute-value pairs (a_i, v_i), where a_i is a feature (or attribute) and v_i is a value from the domain of a_i. The domain of a feature can be continuous or categorical. The target space $\mathcal{Y}$ (with dimensionality equals to one) contains the different labels (classes or outcomes) and also in this case the domain can be continuous or categorical. Note that, in case of ordinal classification labels in Y have an order. A predictor b can be a machine-learning model, a domain-expert rule-based system, or any combination of algorithmic and human knowledge processing. In the following, we denote by b a *black box* predictor, whose internals are either unknown to the observer or they are known but uninterpretable by humans. Examples include neural networks, SVMs, ensemble classifiers, or a composition of data-mining and hard-coded expert systems. Instead, we denote with c an *interpretable* predictor, whose internal processing yielding a decision $c(x)$ can be given a symbolic interpretation comprehensible by a human, i.e., for which a global or a local explanation is available. Examples of such predictors include rule-based classifiers, decision trees, decision sets, and rational functions.

In supervised learning [118], a training dataset D_{train} is used for training a predictor b, and a test dataset D_{test} is used for evaluating the performance of b. In the black box explanations' problems, treated in this survey, D_{train} is usually unknown. Given $D_{test} = \{X, \hat{Y}\}$, the evaluation consists of observing for each pair of data record and target value $(x, \hat{y}) \in D_{test}$ the matches between $\hat{y}$ and $b(x) = y$. The *accuracy* is the percentage of matches over the size of the test dataset.

The predictive performances of both the black box b and the interpretable predictor c can be evaluated through the *accuracy* measure over the test dataset. For the interpretable predictor c, a second measure is the *fidelity*, which evaluates how well c mimicks the black box predictor b on the test dataset. Formally, fidelity is the percentage of matches $c(x) = b(x)$, for $(x, \hat{y}) \in D_{test}$, over the size of the test dataset. Note that the *fidelity* score can be interpreted as the accuracy of the interpretable predictor c with respect to predictions of the black box b. The measures *accuracy* and *fidelity* can be easily extended to more refined ones such as precision, recall, F1-score [118].

Model Explanation

The *black box explanation problem* consists in providing a global explanation of the black box model through an interpretable and transparent model. This model should be both able to mimic the behavior of the black box and it should also be understandable by humans. In other words, the interpretable model approximating the black box must be globally interpretable. We formalize this problem by assuming that the interpretable global predictor is derived from the black box and some dataset of instances X. The dataset X provided by the user is a sampling of the domain $\mathcal{X}^{(m)}$, and it

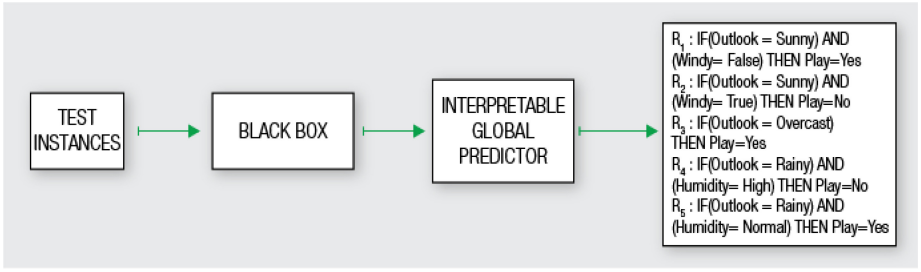

Fig. 6. Model explanation problem example. Starting from test instances in X, first query the black box, and then extract an interpretable global predictor from $\{X, b(X)\}$ in the form of a decision rule classifier.

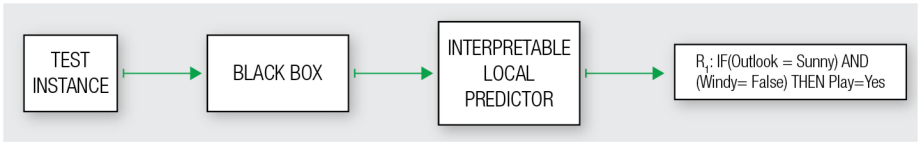

Fig. 7. Outcome explanation problem example. For a test instance x, the black box decision $b(x)$ is explained by building an interpretable local predictor c_l, e.g., a decision rule classifier. The local explanation $\varepsilon_l(c_l, x)$ is the specific rule used to classify x.

may include actual class values (hence allowing to evaluate accuracy of the interpretable model). As extreme possibilities, X can be empty or X can precisely be the training dataset used to learn b. The process of extracting the interpretable predictor may further expand X, e.g., by random perturbation or random sampling. For such test instances, only the predictions of the black box are known—see Figure 6. Therefore, we define the model explanation problem as follows.

Definition 4.1 (Model Explanation Problem). Given a black box predictor b and a set of instances X, the *model explanation problem* consists in finding an explanation $E \in \mathcal{E}$, belonging to a human-interpretable domain $\mathcal{E}$, through an interpretable global predictor $c_g = f(b, X)$ derived from the black box b and the instances X using some process $f(\cdot, \cdot)$. An explanation $E \in \mathcal{E}$ is obtained through c_g, if $E = \varepsilon_g(c_g, X)$ for some explanation logic $\varepsilon_g(\cdot, \cdot)$, which reasons over c_g and X.

A large set of papers reviewed in this survey describe various designs for the function f to solve the explanation problem, including many approaches in the expansion of the dataset X for the purpose of training the interpretable model c_g. Such models can include decision trees or decision rule classifiers [43]. The domain of explanations $\mathcal{E}$ in such cases consists of decision trees and sets of rules, respectively. An example is shown in Figure 6. The definition above also accounts for an explanation logic $\varepsilon_g(\cdot, \cdot)$, which further processes the interpretable global predictor, e.g., to provide an abstraction of c_g tailored at users with different background knowledge. As an example, the explanation logic may condense a decision tree to extract a feature importance vector, summarizing the contribution of features to the decisions of the global model (see Section 6.3).

Outcome Explanation

Given a black box and an input instance, the *outcome explanation problem* consists in providing an explanation for the outcome of the black box on that instance. It is not required to explain the whole logic underlying the black box but only the reason for the prediction on a specific input instance. We formalize this problem by assuming that first an interpretable local model c_l is build from the black box b and the instance x, and then an explanation is derived from c_l—see Figure 7.

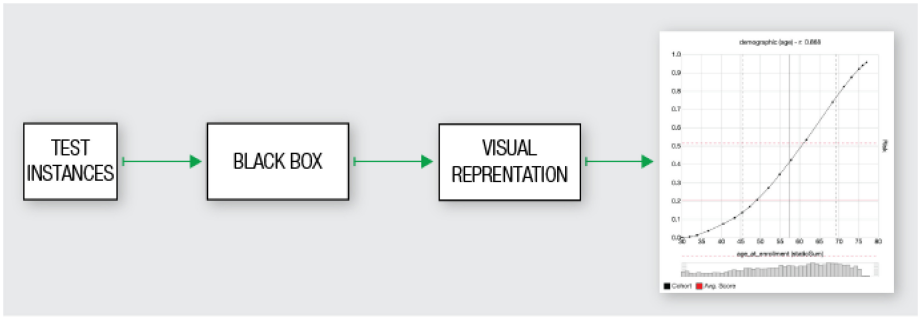

Fig. 8. Model inspection problem example. Query the black box on test instances X, and then extract a sensitivity analysis plot.

Definition 4.2 (Outcome Explanation Problem). Given a black box predictor b and an instance x, the *outcome explanation problem* consists in finding an explanation $e \in \mathcal{E}$, belonging to a human-interpretable domain $\mathcal{E}$, through an interpretable local predictor $c_l = f(b, x)$ derived from the black box b and the instance x using some process $f(\cdot, \cdot)$. An explanation $e \in \mathcal{E}$ is obtained through c_l, if $e = \varepsilon_l(c_l, x)$ for some explanation logic $\varepsilon_l(\cdot, \cdot)$, which reasons over c_l and x.

As an example, the local predictor c_l can be a decision tree built from a neighborhood of x, and an explanation e can be the path of the decision tree followed by attribute values in x [32]. Another example is shown in Figure 7. We will survey recent works adopting very diversified approaches that instantiate the outcome explanation problem.

Model Inspection Problem

The *model inspection problem* consists in providing a representation (visual or textual) for understanding some specific property of the black box model or of its predictions. Example properties of interest include sensitivity to attribute changes, and identification of components of the black box (e.g., neurons in DNN) responsible for specific decisions. We define this problem as follows.

Definition 4.3 (Model Inspection Problem). Given a black box b and a set of instances X, the *model inspection problem* consists in providing a (visual or textual) representation $r = f(b, X)$ of some property of b using some process $f(\cdot, \cdot)$.

For example, the function f may be based on sensitivity analysis that, by observing the changes occurring in the predictions when varying the input of b, returns a set of visualizations (e.g., partial dependence plots [55] or variable effect characteristic curve [19]), highlighting the feature importance for the predictions. See the example in Figure 8. The key element differentiating the inspection problem from the model explanation problem is that the latter requires the extraction of an interpretable global predictor, while the former concentrates the analysis of specific properties of the black box without requiring a global understanding of it. This is readily checked by contrasting Figures 8 and 6.

Transparent Box Design Problem

The *transparent box design problem* consists in directly providing a model that is locally or globally interpretable.

Definition 4.4 (Transparent Box Design Problem). Given a training dataset $D = \{X, \hat{Y}\}$, the *transparent box design problem* consists in learning a locally or globally interpretable predictor c from

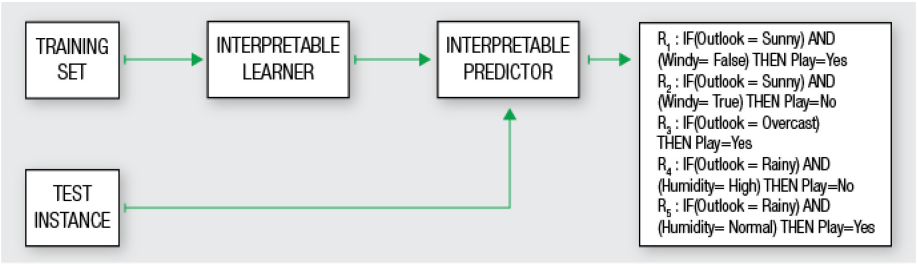

Fig. 9. Transparent box design problem example. A decision rule classifier learned from a training dataset is globally interpretable predictor. Moreover, the rule that applies on a given test instance is a local explanation of the predictor's decision.

D. For a locally interpretable predictor c, there exists a local explanator logic ε_l to derive an explanation $\varepsilon_l(c, x)$ of the decision $c(x)$ for an instance x. For a globally interpretable predictor c, there exists a global explanator logic ε_g to derive an explanation $\varepsilon_g(c, X)$.

An example interpretable predictor c is a decision tree classifier. A local explanation for an instance x can be derived as a decision rule with conclusion $c(x)$ and with premise the conditions from the tree path followed according to the attributed values of x. A global explanation can be the decision tree itself, or, as discussed in the case of model explanation, a feature importance vector condensed from the decision tree. See Figure 9 as example.

In summary, according to our problem definitions, when stating that a method is able to *open the black box*, we are referring to one of the following statements: *(i)* it explains the model, *(ii)* it explains the outcome, *(iii)* it can inspect the black box internally, *(iv)* it provides a transparent solution.

5 PROBLEM AND EXPLANATOR-BASED CLASSIFICATION

In this survey, we propose a classification based on the type of problem faced and on the explanator adopted to open the black box. In particular, in our classification, we take into account the following features:

- the type of *problem* faced (according to the definitions in Section 4);
- the type of *explanator* adopted to open the black box;
- the type of *black box model* that the explanator is able to open;
- the type of *data* used as input by the black box model.

In each section, we group together all the papers with the same problem definition, while the subsections correspond to the different solutions adopted. In turn, in each subsection, we group the papers that try to explain the same type of black box. Finally, we keep the type of data used by the black box as a feature that is specified for each work analyzed.

We organize the sections discussing the different problems as follows. In Section 6, we analyze the papers presenting approaches to solve the *model explanation problem*. These approaches provide a globally interpretable predictor that is able to mimic the black box. Section 7 reviews the methods solving the *outcome explanation problem*: the predictor returned is locally interpretable and provides an explanation only for a given record. In Section 8, we discuss the papers proposing methodologies for *inspecting black boxes*, i.e., not providing a comprehensible predictor but a visualization tool for studying how the black box work internally, and what can happen when a certain input is provided. Finally, in Section 9, we report the papers designing a *transparent* predictor to

overcome the "obscure" limitation of black boxes. These approaches try to provide a global or local interpretable model without sacrificing the accuracy of a black box learned to solve the same task.

We further categorize in the subsections the various methods with respect to the type of interpretable explanator:

- *Decision Tree (DT) or Single Tree.* It is commonly recognized that decision tree is one of the more interpretable and easily understandable models, primarily for global, but also for local, explanations. Indeed, a very widespread technique for opening the black box is the so-called "single-tree approximation."
- *Decion Rules (DR) or Rule Based Explanator.* Decision rules are among the more human understandable techniques. There exist various types of rules (illustrated in Section 3.3). They are used to explain the model, the outcome and also for the transparent design. We remark the existence of techniques for transforming a tree into a set of rules.
- *Features Importance (FI).* A very simple but effective solution acting as either global or local explanation consists in returning as explanation the weight and magnitude of the features used by the black box. Generally the feature importance is provided by using the values of the coefficients of linear models used as interpretable models.
- *Saliency Mask (SM).* An efficient way of pointing out what causes a certain outcome, especially when images or texts are treated, consists in using "masks" visually highlighting the determining aspects of the record analyzed. They are generally used to explain deep neural networks and can be viewed as a visual representation of FI.
- *Sensitivity Analysis (SA).* It consists of evaluating the uncertainty in the outcome of a black box with respect to different sources of uncertainty in its inputs. It is generally used to develop visual tools for model inspection.
- *Partial Dependence Plot (PDP).* These plots help in visualizing and understanding the relationship between the outcome of a black box and the input in a reduced feature space.
- *Prototype Selection (PS).* This explanator consists in returning, together with the outcome, an example very similar to the classified record, to make clear which criteria the prediction was returned. A prototype is an object that is representative of a set of similar instances and is part of the observed points, or it is an artifact summarizing a subset of them with similar characteristics.
- *Activation Maximization (AM).* The inspection of neural networks and deep neural network can be carried out also by observing which are the fundamental neurons activated with respect to particular input records, i.e., to look for input patterns that maximize the activation of a certain neuron in a certain layer. AM can be viewed also as the generation of an input image that maximizes the output activation (also called adversarial generation).

In the following, we list all the black boxes opened in the reviewed papers.

- *Neural Network (NN).* Inspired by biological neural networks, artificial neural networks are formed by a set of connected neurons. Each link between neurons can transmit a signal. The receiving neuron can process the signal and then transmit to downstream neurons connected to it. Typically, neurons are organized in layers. Different layers perform different transformations on their inputs. Signals travel from the input layer, to the output layer, passing through the hidden layer(s) in the middle multiple times. Neurons and connections may also have a weight that varies as learning proceeds, which can increase or decrease the strength of the signal.
- *Tree Ensemble (TE).* Ensemble methods combine more than one learning algorithm to improve the predictive power of any of the single learning algorithms that they combines.

Random forests, boosted trees and tree bagging are examples of tree ensembles. They combine the predictions of different decision trees each one trained on an independent subset (with respect to features and records) of the input data.

- *Support Vector Machine (SVM).* Support Vector Machines utilize a subset of the training data, called support vectors, to represent the decision boundary. A SVM is a classifier that, using a set of available different kernels, searches for hyperplanes with the largest margin for the decision boundary.
- *Deep Neural Network (DNN).* A DNN is a NN that can model complex non-linear relationship with multiple hidden layers. A DNN architecture is formed by a composition of models expressed as a layered combination of basic units. In DNNs the data typically flows from the input to the output layer without looping back. A largely used DNN are Recurrent Neural Networks (RNNs). A peculiar component of RNNs are Long Short-Term Memory (LSTM) nodes, which are particularly effective for language modeling. However, in image processing Convolutional Neural Networks (CNNs) are typically used. We distingish between NN and DNN solely on the fact that DNN can be deeper than NN and that can use a more sophisticated node architecture (e.g., RNN, CNN).
- *Non-Linear Models (NLM).* The function used to model the observations is a nonlinear combination of the model parameters and depends on one or mode independent variables.

Recently, *agnostic* approaches for explaining black boxes are being developed. An *Agnostic Explanator (AGN)* is a comprehensible predictor that is not tied to a particular type of black box, explanation or data type. In other words, in theory, an agnostic predictor can explain indifferently a neural network or a tree ensemble using a single tree or a set of rules. Since only a few approaches in the literature describe themselves to be *fully* agnostic, and since the principal task is to explain a black box predictor, in this article, if not differently specified, with the term *agnostic*, we refer only to the approaches defined to explain any type of black box, i.e., black box agnostic.

The types of data used as input of black boxes analyzed in this survey are the following:

- *Tabular (TAB).* With tabular data, we indicate any classical dataset in which every record shares the same set of features and each feature is either numerical, categorical or boolean.
- *Image (IMG).* Many black boxes work with labeled images. These images can be treated as they are by the black box or can be preprocessed (e.g, re-sized to have all the same dimensions).
- *Text (TXT).* As language modeling is one of the tasks most widely assessed nowadays together with image recognition, labeled datasets of text are generally used for tasks like spam detection or topic classification.

In data mining and machine learning classification problems other types of data are also used like sequences, networks, mobility trajectories, and so on. However, with the exceptions of [39, 113] that uses EEG data that can be roughly categorized as tabular data, types of data different from tabular, images and text are not generally used as input for the methods of the papers proposing a solution for opening a black box system.

Tables 2, 3, 4, and 5 list the methods for opening and explaining black boxes and summarizes the various fundamental features and characteristics listed so far for each of the four recognized problems model explanation, outcome explanation, model inspection, and transparent box design, respectively. Moreover, they also provide additional information that we believe could be useful for the reader. The columns *Examples*, *Code*, and *Dataset* indicate if any kind of example of explanation is shown in the paper and if the source code and the dataset used in the experiments are publicly available, respectively. The columns *General* and *Random* are discussed in the following section.

Table 1. Legend of Tables 2, 3, 4, and 5

Column	Description
Problem	Model Explanation, Outcome Explanation, Model Inspection, Transparent Design
Explanator	DT–Decision Tree, DR–Decision Rules, FI–Features Importance, SM–Saliency Masks, SA–Sensitivity Analysis, PDP–Partial Dependence Plot, AM–Activation Maximization, PS–Prototype Selection
Black Box	NN–Neural Network, TE–Tree Ensemble, SVM–Support Vector Machines, DNN–Deep Neural Network, AGN–AGNostic black box, NLM–Non Linear Models
Data Type	TAB–TABular, IMG–IMaGe, TXT–TeXT, ANY–ANY type of data
General	Indicates if an explanatory approach can be generalized for every black box, i.e., it does not consider peculiarities of the black box to produce the explanation
Random	Indicates if any kind of random perturbation of the dataset is performed
Examples	Indicates if example of explanations are shown in the paper
Code	Indicates if the source code is available
Dataset	Indicates if the datasets used in the experiments are available

In the following are described the features reported and the abbreviations adopted.

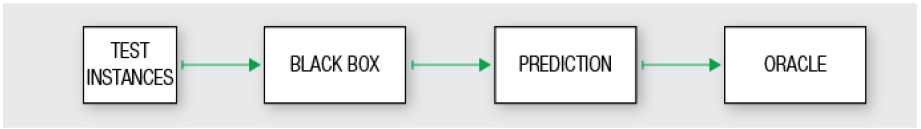

Fig. 10. Reverse engineering approach: the learned black box predictor b is queried with a test dataset $D = \{X, Y\}$ to produce an oracle $\hat{Y}$, which associate to each record $x \in X$, a label that is not real but assigned by the black box.

We point out that Tables 2, 3, 4, and 5 report the main references only, while existing extensions or derived works are discussed in the survey. Table 1 reports the legend for the aforementioned tables, i.e., expanded acronyms and the meaning of the columns. Moreover, to provide the reader a useful survey, the tables can help in finding a particular set of papers with determined characteristics; Appendix A provides Tables 6, 7, and 8, in which are reported the list of the papers with respect to each problem, explanator, and black box, respectively.

Reverse Engineering: A Common Approach for Understanding the Black Box

Before proceeding in the detailed analysis and classification of papers proposing method f for understanding black boxes b, we present in this section the most widely used approach to solve the three black box model explanation problems. We refer to this approach as *reverse engineering*, because the black box predictor b is queried with a certain test dataset to create an *oracle* dataset that in turn will be used to train the comprehensible predictor (see Figure 10). The name "reverse engineering" comes from the fact that we can only observe the input and output of the black box.

With respect to the black box model and outcome explanation problems, the possibility of action tied with this approach relies on the choice of adopting a particular type of comprehensible predictor, and in the possibility of querying the black box with input records created in a controlled way and/or by using *random perturbations* of the initial train or test dataset. Regarding the random perturbations of the input used to feed the black box, it is important to recall that recent studies

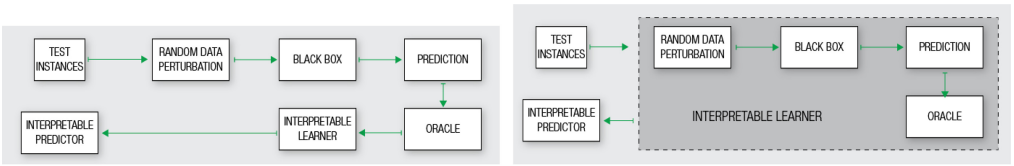

Fig. 11. *(Left)* Generalizable reverse engineering approach: internal peculiarities of the black box b are not exploited to build the comprehensible predictor c. *(Right)* Not Generalizable reverse engineering approach: the comprehensible predictor c is the result of a procedure involving internal characteristics of the black box b.

discovered that DNN built for classification problems on texts and images can be easily fooled (see Section 2). Changes in an image that are imperceptible to humans can lead a DNN to label the record as something else. Thus, according to these discoveries, the methods treating images or text, in theory, should not be enabled to use completely random perturbations of their input. However, this is not always the case in practice [98].

Such a reverse engineering approach can be classified as *generalizable* or not (or pedagocial vs. decompositional as described in Reference [76]). We say that an approach is generalizable when a purely reverse engineering procedure is followed, i.e., the black box is only queried with different input records to obtain an oracle used for learning the comprehensible predictor (see Figure 11(left)). In other words, internal peculiarities of the black box are not exploited to build the comprehensible predictor. Thus, if an approach is generalizable, even though it is presented to explain a particular type of black box, in reality, it can be used to interpret any kind of black box predictor. That is, it is an agnostic approach for interpreting black boxes. However, we say that an approach is not generalizable if it can be used to open only that particular type of black box for which it was designed for (see Figure 11(right)). For example, if an approach is designed to interpret random forest and internally use a concept of distance between trees, then such an approach cannot be used to explain predictions of a NN. A not generalizable approach cannot be black box agnostic.

In Tables 6, 7, and 8, we keep track of these aspects with the two features *General* and *Random*. With *General*, we indicate if an explanatory approach can be generalized for every black box, while with *Random*, we indicate if any kind of random perturbation or permutation of the original dataset is used by the explanatory approach.

In light of these concepts, as the reader will discover below, a further classification not explicitly indicated emerges from the analysis of these papers. This fact can be at the same time a strong point or a weakness of the current state of the art. Indeed, we highlight that the works for opening the black box are realized for two cases. The first (larger) group contains approaches proposed to tackle a particular problem (e.g., medical cases) or to explain a particular type of black box, that is, the solutions are specific for the problem instance. The second group contains general purpose solutions that try to be general as much as possible and propose agnostic and generalizable solutions.

6 SOLVING THE MODEL EXPLANATION PROBLEM

In this section, we review the methods for opening the black box facing the *model explanation problem* (see Section 4.1). The proposed methods provide *globally* interpretable models that are able to mimic the behavior of black boxes and that are also understandable by humans. We recognized different groups of approaches. In Section 6.1, we analyze the proposals using a decision tree as explanator, while in Section 6.2 those using rules. Section 6.3 describes the methods that are designed to work with any type of black box. Finally, Section 6.4 contains the remaining ones.

Table 2. Summary of Methods for Opening Black Boxes Solving the *Model Explanation* Problem

Name	Ref.	Authors	Year	Explanator	Black Box	Data Type	General	Random	Examples	Code	Dataset
Trepan	[22]	Craven et al.	1996	DT	NN	TAB	✓				✓
–	[57]	Krishnan et al.	1999	DT	NN	TAB	✓		✓		✓
DecText	[12]	Boz	2002	DT	NN	TAB	✓	✓			✓
GPDT	[46]	Johansson et al.	2009	DT	NN	TAB	✓	✓	✓		✓
Tree Metrics	[17]	Chipman et al.	1998	DT	TE	TAB					✓
CCM	[26]	Domingos et al.	1998	DT	TE	TAB	✓	✓			✓
–	[34]	Gibbons et al.	2013	DT	TE	TAB	✓	✓			
STA	[140]	Zhou et al.	2016	DT	TE	TAB		✓			
CDT	[104]	Schetinin et al.	2007	DT	TE	TAB			✓		
–	[38]	Hara et al.	2016	DT	TE	TAB		✓	✓		✓
TSP	[117]	Tan et al.	2016	DT	TE	TAB					✓
Conj Rules	[21]	Craven et al.	1994	DR	NN	TAB		✓			
G-REX	[44]	Johansson et al.	2003	DR	NN	TAB	✓	✓	✓		
REFNE	[141]	Zhou et al.	2003	DR	NN	TAB	✓	✓	✓		✓
RxREN	[6]	Augasta et al.	2012	DR	NN	TAB		✓	✓		✓
SVM+P	[82]	Nunez et al.	2002	DR	SVM	TAB			✓		✓
–	[33]	Fung et al.	2005	DR	SVM	TAB			✓		✓
inTrees	[25]	Deng	2014	DR	TE	TAB			✓		✓
–	[70]	Lou et al.	2013	FI	AGN	TAB	✓		✓	✓	✓
GoldenEye	[40]	Henelius et al.	2014	FI	AGN	TAB	✓	✓	✓	✓	✓
PALM	[58]	Krishnan et al.	2017	DT	AGN	ANY	✓		✓		✓
FIRM	[142]	Zien et al.	2009	FI	AGN	TAB	✓	✓	✓	✓	✓
MFI	[124]	Vidovic et al.	2016	FI	AGN	TAB	✓	✓	✓		✓
–	[121]	Tolomei et al.	2017	FI	TE	TAB			✓		✓
POIMs	[111]	Sonnenburg et al.	2007	FI	SVM	TAB			✓	✓	✓

Table 2 summarizes and categorizes these contributions according to the features described in Section 5.

6.1 Explanation via Single-Tree Approximation

The following set of works address the *model explanation problem* implementing in different ways the function f. However, all these works adopt a *decision tree* as comprehensible global predictor c_g, and consequently represent the explanation ε_g with the decision tree itself. We point out that all the methods presented in this section work on tabular data.

6.1.1 Explanation of Neural Networks. The following papers describe the implementation of functions f that are able to interpret a black box b consisting in a *Neural Network (NN)* [118] with a comprehensible global predictor c_g consisting in a decision tree. In these works, the NNs are considered black-boxes, i.e., the only interface permitted is presenting an input X to the neural network b and obtaining the outcome Y. The final goal is to comprehend how the neural networks behave by submitting to it a large set of instances and analyzing their different predictions.

Single-tree approximations for NNs were first presented in 1996 by Craven et al. [22]. The comprehensible representations of the neural network b is returned by *Trepan*, which is the implementation of function f. Trepan queries the neural network b to induce a decision tree

c_g approximating the concepts represented by the networks by maximizing the gain ratio [118] together with an estimation of the current model fidelity. Another advantage of Trepan with respect to common tree classifiers like ID3 or C4.5 [118] is that, thanks to the black box b, it can use as many instances as desired for each split, so that also the node splits near to the bottom of the tree are realized using a considerable amount of data.

In Reference [57], Krishnan et al. present a three step method f. The first step generates a sort of "prototype" for each target class in Y by using genetic programming to query the trained neural network b. The input features dataset X is exploited for constraining the prototypes. The second step selects the best prototypes for inducing the learning of the decision tree c_g in the third step. This approach leads to get more understandable and smaller decision trees starting from smaller data sets.

In Reference [12], Boz describes *DecText* that uses a decision tree c_g to explain a neural network b. The overall procedure recalls Trepan [22] with the innovation of four splitting methods aimed at finding the most relevant features during the tree construction. Moreover, since one of the main purposes of the tree is to maximize the fidelity while keeping the model "simple," a fidelity-based pruning strategy to reduce the tree size is defined. A set of random instances are generated. Then, starting from the bottom of the tree, for each internal node a leaf is created with the majority label using the labeling of the random instances. If the fidelity of the new tree overtakes the old one, then the maximum fidelity and the tree are updated.

In Reference [46], Johansson et al. use *Genetic Programming* to evolve Decision Trees (the comprehensible global predictor c_g), to mimic the behavior of a neural network ensemble b. The dataset D used by genetic programming (implementing function f) consists of a lot of different combinations of the original data and oracle data labeled by b. The paper shows that trees based only on original training data have the worst performance in terms of accuracy in the test data, while the trees evolved using both the oracle guide and the original data produce significantly more accurate trees c_g.

We underline that, even though these approaches are developed to explain neural networks, since peculiarities of the neural networks are not used by f, which uses b only as an oracle, these approaches can be potentially adopted as agnostic explanators, i.e., they can be used to open any kind of black box and represent it with a single tree.

6.1.2 Explanation of Tree Ensembles. Richer collections of trees provide higher performance and less uncertainty in the prediction. However, it is generally difficult to make sense of the resultant forests. The following papers describe functions f for approximating a black box model b consisting in *Tree Ensembles (TE)* [118] (e.g., random forests) with a global comprehensible predictor c_g in the form of a single decision tree, and explanation ε_g as a the decision tree itself.

Unlike previous works, the tree ensembles are not only viewed as black boxes but also some of their internal features are used to derive the global comprehensible model c_g. For example, in Reference [17], Chipman et al. observe that although hundreds of distinct trees are identified by *random forests*, in practice, many of them generally differ only by few nodes. In addition, some trees may differ only in the topology, but use the same partitioning of the feature space $\mathcal{X}$. The paper proposes several measures of dissimilarity for trees. Such measures are used to summarize forest of trees through clustering and finally use archetypes of the associated clusters as model explanation. Here, f corresponds to the clustering procedure, and the global comprehensible predictor c_g is the set of tree archetypes minimizing the distance among all the trees in each cluster. In this approach, f does not extend the input dataset D with random data.

However, random data enrichment and model combination are the basis for the *Combined Multiple Model (CCM)* procedure f presented in Reference [26]. Given the tree ensemble black box b, it

first modifies n times the input dataset D and learns a set of n black boxes b_i $\forall i = 1, \ldots, n$, and then it randomly generates data record x, which are labeled using a *combination* (e.g. bagging) of the n black boxes b_i, i.e., $C_{b_1,\ldots,b_n}(x) = \hat{y}$. In this way, the training dataset $D = D \cup \{x, \hat{y}\}$ is increased. Finally, it builds the global comprehensible model c_g as a decision tree (C4.5 [92]) on the enriched dataset D. Since it is not exploiting particular features of the tree ensemble b, also this approach can be generalized with respect to the black box b. In line with Reference [26], the authors of Reference [34] generate a very large artificial dataset D using the prediction of the random forest b, then explain b by training a decision tree c_g on this artificial dataset to mime the behavior of the random forest. Finally, they improve the comprehensibility of c_g by cutting the decision tree with respect to a human understandable depth (i.e., from 6 to 11 nodes of depth). Reference [140] proposes *Single-Tree Approximation (STA)*, an extension of Reference [34], which empowers the construction of the final decision tree c_g by using test hypothesis to understand which are the best splits observing the Gini indexes on the trees of the random forest b.

Schetinin et al. in Reference [104] present an approach for the probabilistic interpretation of the black box b *Bayesian decision trees ensembles* [13] through a quantitative evaluation of uncertainty of a *Confident Decision Tree* (CDT) c_g. The methodology f for interpreting b is summarized as follows: *(i)* the classification confidence for each tree in the ensemble is calculated using the training data D, *(ii)* the decision tree c_g that covers the maximal number of correct training examples is selected, keeping minimal the amount of misclassifications on the remaining examples by subsequentially refining the training dataset D. Similar to Reference [17], this explanation method f does not extend the input dataset D with random data and cannot be generalized to other black boxes but can be used only with Bayesian decision tree ensembles.

In Reference [38], Hara et al. reinterpret *Additive Tree Models (ATM)* (the black box b) using a probabilistic generative model interpretable by humans. An interpretable ATM has a sufficiently small number of regions. Their aim is to reduce the number of regions in an ATM while minimizing the model error. To satisfy these requirements, they propose a post processing f that first learns an ATM b generating a number of regions, and then, it mimics b using a simpler model (the comprehensible global predictor c_g) where the number of regions is fixed as small, e.g., ten. To obtain c_g an Expectation Maximization algorithm is adopted [118] minimizing the Kullback-Leibler divergence from the ensemble b.

The authors of Reference [117] propose the *Tree Space Prototype (TSP)* f for interpretating tree ensembles b by finding tree prototypes (the comprehensible global predictor c_g) in the tree space. The main contributions for f are: *(i)* the definition of the *random forest proximity* between trees, and *(ii)* the procedure to extract the tree prototypes used for classification.

6.2 Explanation via Rule Extraction

Another commonly used interpretable and easily understandable model is the *set of rules*. When a set of rules describing the logic behind the black box model is returned the interpretability is provided at a global level. In the following, we present a set of reference works solving the *model explanation problem* by implementing in different ways function f, and by adopting any kind of *decision rules* as comprehensible global predictor c_g. Hence, the global explanation ε_g changes accordingly to the type of rules extracted by c_g. Also, all the methods presented in this section work on tabular data.

6.2.1 Explanation of Neural Networks. The following papers describe the implementation of functions f, which are able to interpret a black box b consisting of a *neural network* [118]. A specific survey on techniques extracting rules from neural networks is Reference [5]. It provides an overview of mechanisms designed to *(i)* insert knowledge into neural networks (knowledge

initialization), *(ii)* extract rules from trained NNs (rule extraction), and *(iii)* use NNs to refine existing rules (rule refinement). The approaches presented in Reference [5] are strongly dependent on the black box b and on the specific type of decision rules c_g. Thus, they are not generalizable and can not be employed to solve other instances of the problem. The survey Reference [5] classifies the methods according to the following criteria:

- Expressive power of the extracted rules.
- Translucency: that is decompositional, pedagogical, and eclectic properties.
- Portability of the rule extraction technique.
- Quality of the rules extracted (e.g., accuracy, fidelity, consistency).
- Algorithmic complexity.

A typical paper analyzed in Reference [5] is Reference [21], where Craven et al. present a method f to explain the behavior of a neural network b by transforming rule extraction (which is a search problem) into a learning problem. The original training data D and a randomized extension of it are provided as input to the black box b. If the input $x \in D$ with outcome $\hat{y}$ is not covered by the set of rules, then a *conjunctive* (or m-of-n) rule is formed from $\{x, \hat{y}\}$ considering all the possible antecedents. The procedure ends when all the target classes have been processed.

In Reference [47] Johansson et al. exploit *G-REX* [44], an algorithm for rule extraction, as function f to explain a neural network b. They use the classical reverse engineering schema where random permutations of the original dataset D are annotated by b, and such dataset is used as input by G-REX, which correspond with c_g in this case. In particular, G-REX extracts rules by exploiting genetic programming as a key concept. In subsequent works, the authors show that the proposed methodology f can be also employed to interpret trees ensembles. Reference [45] extends G-REX for handling regression problems by generating regression trees, and classification problems by generating fuzzy rules.

In Reference [141], the authors present *REFNE*, an approach f to explain neural network ensembles b. REFNE uses ensembles for generating instances and then, extracts symbolic rules c_g from those instances. REFNE avoids useless discretizations of continuous attributes, by applying a particular discretization leading to discretize different continuous attributes using different intervals. Moreover, REFNE can also be used as a rule learning approach, i.e., it solves the transparent box design problem (see Section 4.1). Also, in Reference [6], Augasta et al. propose *RxREN* a rule extraction algorithm c_g, which returns the explanation of a trained NN b. The method f works as follows. First, it prunes the insignificant input neurons from trained NNs and identifies the data range necessary to classify the given test instance with a specific class. Second, using a reverse engineering technique, through RxREN generates the classification rules for each class label exploiting the data ranges previously identified, and improve the initial set of rules by a process that prunes and updates the rules.

6.2.2 Explanation of Support Vector Machines. In the following are shown methods f for explaining *Support Vector Machine* (SVM) [118] still returning a comprehensible global predictor c_g consisting in a rule-based classifier.

The authors of Reference [82] propose the *SVM+Prototypes (SVM+P)* procedure f for rule extraction c_g from support vector machines b. It works as follows: it first determines the decision function by means of a SVM, then a clustering algorithm is used to find out a prototype vector for each class. By using geometric methods, these points are joined with the support vectors for defining ellipsoids in the input space that can be transformed into if-then rules.

Fung et al., in Reference [33], describe as function f an algorithm based on constraint programming for converting linear SVM b (and other hyperplane-based linear classifiers) into a set of non

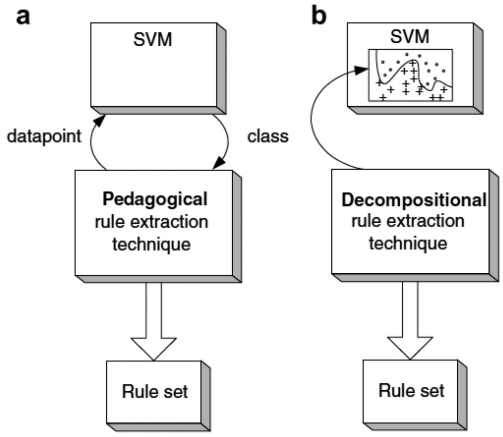

Fig. 12. From Reference [76]: pedagogical *(a)* and decompositional *(b)* rule extraction techniques.

overlapping and interpretable rules c_g. These rules are asymptotically equivalent to the original linear SVM. Each iteration of the algorithm for extracting the rules is designed to solve a constrained optimization problem having a low computational cost. We underline that this black box explanation solution f is not generalizable and can be employed only for Linear SVM-like black boxes.

In Reference [76] the authors propose a qualitative comparison of the explanations returned by techniques for extraction of rules from SVM black boxes (e.g., SVM+P [82], Fung method [33]) against the redefining of methods designed for explaining neural networks, i.e., C4.5 [118], Trepan [22], and G-REX [44]. How we anticipated in the previous section, the authors delineate the existence of two type of approaches to extract rules: *pedagogical* and *decompositional* (see Figure 12). Pedagogical techniques f directly extract rules that relate the inputs and outputs of the predictor (e.g., Reference [22, 44]), while decompositional approaches are closely intertwined with the internal structure of the SVM (e.g., Reference [33, 82]). We recall that, in Table 2, we identify with the term generalizable the pedagogical approaches.

6.2.3 Explanation of Tree Ensembles. Finally, in Reference [25], Deng proposes the *inTrees* framework f to explain black boxes b defined as *Tree Ensembles (TE)* by returning a set of decision rules c_g. InTrees extracts, measures, prunes and selects rules from tree ensembles, and calculates frequent variable interactions. The set of black boxes b that inTrees can explain is represented by any kind of tree ensemble like random forests, regularized random forests and boosted trees. InTrees can be used for both classification and regression problems. The technique described by InTrees is also known as *Simplified Tree Ensamble Learner (STEL)*: It extracts the most supported and simplest rules form the trees ensemble.

6.3 Agnostic Explanator

Recent approaches for interpretation are *agnostic (AGN)* with respect to the black box to be explained. In this section, we present a set of works solving the *model explanation problem* by implementing function f such that any type of black box b can be explained. These approaches do not return a specific comprehensible global predictor c_g, thus the type of explanation ε_g change with respect to f and c_g. By definition all these approaches are generalizable.

Probably the first attempt of an agnostic solution was proposed in Reference [69]. Lou et al. propose a method f, which exploits Generalized Additive Models (GAMs) and it is able to interpret regression splines (linear and logistics), single trees and tree ensembles (bagged trees, boosted

trees, boosted bagged trees and random forests). GAMs are presented as the gold standard for intelligibility when only univariate terms are considered. Indeed, the explanation ε_c is returned as the importance of the contribution of the individual features in b together with their *shape function*, such that the impact of each predictor can be quantified. A shape function is the plot of a function capturing the linearities and nonlinearities together with its shape. It works on tabular data. A refinement of the GAM approach is proposed by the same authors in Reference [70]. A case study on health care showing the application of the GAM the refinement is presented in Reference [16]. In particular, this approach is used for the prediction of the pneumonia risk and hospital 30-day readmission.

In Reference [40], the authors present an iterative algorithm f, named *GoldenEye*, which is based on data randomization (within class permutation, data permutation, etc.) and on finding groups of attributes whose interactions have an impact on the predictive power. The attributes and the dependencies among the grouped attributes represent the global explanation ε_g.

In Reference [58], *Partition Aware Local Model (PALM)* is presented to implement f. PALM is a method able to learn and summarize the structure of the training dataset to help the machine-learning debugging. PALM mimes a black box b using a meta-model for partitioning the training dataset, and a set of sub-models for approximating and miming the patterns within each partition. As meta-model it uses a decision tree (c_g) so that the user can examine its structure and determine if the rules detected follow the intuition or not, and link efficiently problematic test records to the responsible train data. The sub-models linked to the leaves of the tree can be arbitrarily complex models able to catch elaborate local patterns, but yet interpretable by humans. Thus, with respect to the final sub-models PALM is not only black box agnostic but also explanator agnostic. Moreover, PALM is also data agnostic; i.e., it can work on any kind of data.

In Reference [142] is introduced *FIRM* (Feature Importance Ranking Measure) for implementing function f as an extension of the POIMs [111]. FIRM takes the underlying correlation structure of the features into account and it is able to discover the most relevant ones. The model explanation ε_g is provided through a vector of features importance. FIRM is general as it can be applied to a very broad family of classifiers. In summary, the FIRM score depends on the conditional expected value for a feature and the importance of the feature is measured as the variability of the conditional expected score.

A further extension of FIRM is proposed in Reference [124]. Vidovic et al. propose the *Measure of Feature Importance (MFI)*. As FIRM, MFI can be applied to any classifier. MFI is intrinsically non-linear and can detect features that by itself are inconspicuous and only impact the prediction function through their interaction with other features. Moreover, MFI can be used for solving both the black box model explanation and the outcome explanation.

6.4 Explanation via Other Approaches

In Reference [121], a solution for the *model explanation problem* is presented. It adopts an approach that cannot be classified as one of the previous. The proposed approach f uses the internals of a random forest model b to produce recommendations on the transformation of true negative examples into positively predicted examples. These recommendations, which are strictly related to the feature importance, corresponds to the comprehensible global predictor c_g. In particular, the function f aims at transforming a negative instance into a positive instance by analyzing the path on the trees in the forest predicting such instance as positive or negative. The explanation of b is provided by means of the helpfulness of the features in the paths adopted for changing the instance outcome from negative to positive.

Another approach hard to classify is Reference [111], based on References [142] and [124]. The authors explain SVM black boxes b through the concept of *POIMs: Positional Oligomiter Importance*

Table 3. Summary of Methods for Opening Black Boxes Solving the *Outcome Explanation* Problem

Name	Ref.	Authors	Year	Explanator	Black Box	Data Type	General	Random	Examples	Code	Dataset
–	[134]	Xu et al.	2015	SM	DNN	IMG			✓	✓	✓
–	[30]	Fong et al.	2017	SM	DNN	IMG			✓		
CAM	[139]	Zhou et al.	2016	SM	DNN	IMG			✓	✓	✓
Grad-CAM	[106]	Selvaraju et al.	2016	SM	DNN	IMG			✓	✓	✓
–	[109]	Simonian et al.	2013	SM	DNN	IMG			✓		✓
PWD	[7]	Bach et al.	2015	SM	DNN	IMG			✓		✓
–	[113]	Sturm et al.	2016	SM	DNN	IMG			✓		✓
DTD	[78]	Montavon et al.	2017	SM	DNN	IMG			✓		✓
DeapLIFT	[107]	Shrikumar et al.	2017	FI	DNN	ANY			✓	✓	
CP	[64]	Landecker et al.	2013	SM	NN	IMG			✓		
–	[143]	Zintgraf et al.	2017	SM	DNN	IMG			✓	✓	✓
VBP	[11]	Bojarski et al.	2016	SM	DNN	IMG			✓		✓
–	[65]	Lei et al.	2016	SM	DNN	TXT			✓		✓
ExplainD	[89]	Poulin et al.	2006	FI	SVM	TAB		✓	✓		
–	[29]	Strumbelj et al.	2010	FI	AGN	TAB	✓	✓	✓		✓
LIME	[98]	Ribeiro et al.	2016	FI	AGN	ANY	✓	✓	✓	✓	✓
MES	[122]	Turner et al.	2016	DR	AGN	ANY	✓		✓		✓
Anchors	[99]	Ribeiro et al.	2018	DR	AGN	ANY	✓	✓	✓	✓	✓
–	[110]	Singh et al.	2016	DT	AGN	TAB	✓	✓	✓		✓
LORE	[37]	Guidotti et al.	2018	DR	AGN	TAB	✓	✓	✓	✓	✓
MFI	[124]	Vidovic et al.	2016	FI	AGN	TAB	✓	✓	✓		✓
–	[39]	Haufe et al.	2014	FI	NLM	TAB			✓		

Matrices (i.e., the global explanation function provided ε_g). The problem faced in Reference [111] is specific for the classification of DNA sequences and k-mers. The POIMs f is a scoring system that assigns a score to each k-mer and it can be used to rank and visualize the k-mer scoring system.

7 SOLVING THE OUTCOME EXPLANATION PROBLEM

In this section, we review the methods solving the *outcome explanation problem* (see Section 4.1). These methods provide a *locally interpretable* model that is able to explain the prediction of the black box in understandable terms for humans for a specific instance or record. This category of approaches using a local point of view with respect to the prediction is becoming the most studied in recent years. Section 7.1 describes the methods providing the salient parts of the record for which a prediction is required using *Deep Neural Networks (DNNs)*, while Section 7.2 analyzes the methods that are able to provide a local explanation for any type of black box. Table 3 summarizes and categorizes these works.

7.1 Explanation of Deep Neural Network via Saliency Masks

In the following works the opened black box b is a DNN and the explanation is provided by using a *Saliency Mask (SM)* as comprehensible local predictor c_l, i.e., a subset of the original record, which is mainly responsible for the prediction. For example, as a salient mask, we can consider the part of an image or a sentence in a text. A saliency image summarizes where a DNN looks into an image for recognizing their predictions. The function f to extract the local explanation ε_l is always

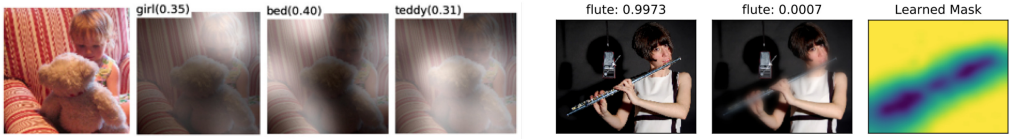

Fig. 13. Saliency masks for explanation of deep neural network. *(Left)* From [134] the elements of the image highlighted. *(Right)* From Reference [30] the mask and the level of accuracy on the image considering and not considering the learned mask.

not generalizable and often strictly tied with the particular type of network, i.e., convolutional, recursive, and so on. We point out that some of the papers described in this section could also be categorized as methods for solving the model inspection problem and thus presented in Section 8 and vice-versa. Another aspect that is worth to highlight is the slight difference between saliency masks and features importance. Indeed, they can be seen as two different ways of building the same explanation: in both cases a value that expresses the importance of a feature/area is provided.

Reference [134] introduces an *attention-based model* f, which automatically identifies the contents of an image. The black box is a neural network that consists of a combination of a *Convolutional NN (CNN)* for the features extraction and a *Recursive NN (RNN)* containing Long Short Term Memory (LSTM), nodes producing the image caption by generating a single word for each iteration. The explanation ε_l of the prediction is provided through a visualization of the attention (area of an image, see Figure 13(left)) for each word in the caption. A similar result is obtained by Fong et al. in Reference [30]. In this work, the authors propose a framework f of explanations c_l as meta-predictors. In their view, an explanation ε_l, and thus a meta-predictor, is a rule that predicts the response of a black box b to certain inputs. Moreover, they use *saliency maps* as explanations for black boxes to highlight the salient part of the images (see Figure 13(right)).

Similarly, another set of works produce saliency masks incorporating network activations into their visualizations. This kind of approaches f are named *Class Activation Mapping (CAM)*. In Reference [139], global average pooling in CNN (the black box b) is used for generating the CAM. A CAM (the local explanation ε_l) for a particular outcome label indicates the discriminative active region that identifies that label. Reference [106] defines its relaxed generalization Grad-CAM, which visualizes the linear combination of a late layer's activations and label-specific weights (or gradients for Reference [139]). All these approaches invoke different versions of *back propagation and/or activation*, which results in aesthetically pleasing, heuristic explanations of image saliency. Their solution is not black box agnostic limited to NN, but it requires specific architectural modifications [139] or access to intermediate layers [106]. However, in Reference [109] is proposed a method f for visualizing saliency mask ε_l specialized for CNN black boxes b.

Another set of works base their explanation on saliency masks ε_l and on using a technique f, which involves the backpropagtion from the output to the input layer. This technique is called *Layer-wise Relevance Propagation (LRP)* and consists in assigning a relevance score fore each layer backpropagating the effect of a decision on a certain image up to the input level. Thus, LRP can also be seen as a way for obtaining the features importance, then visualized through saliency masks. The first method exploiting this technique is the *Pixel-wise Decomposition method (PWD)* described in Reference [7]. In this work, the saliency masks are named *heatmaps*. PWD is presented as an explanator of nonlinear classifiers and in particular of neural networks. Input images are encoded in f using the bag of (visual) word features and the contribution of every pixel is shown as an heatmap that shows the focus of the black box b for taking a certain decision. In Reference [113], LRP is specifically adopted for working on DNN trained to classify EEG analysis data. An evolution of LRP is the *Deep Taylor Decomposition (DTD)* method illustrated in Reference [78]. It tries

to overcome a functional approach, where the explanation results form the local analysis of the prediction function [109], and where the explanation is obtained by running a backward pass in that graph [7, 137]. Also DTD is used for interpreting multilayer neural networks by composing the network decision into contributions (*relevance propagation*) of its input elements. In [103] it is presented a comparison between explanations provided with LRP and more traditional deconvolution and sensitivity analysis. Another work following this line of research is Reference [107], which focuses on the activation differences. It is worth to mention that perhaps the first step toward LRP was Reference [64], with the *Contribution Propagation (CP)* method f. Similar to LRP, *Contribution Propagation*, inspired by Reference [89], calculates the contributions backwards from the output level to the input in such a way that for each instance, CP explains how important each part of the record was for that decision.

In Reference [143] the authors present a probabilistic methodology f for explaining classification decisions made by DNN. The method can be used to produce a saliency map for each instance and also for each node of the neural network (in this sense it solves the model inspection problem). In particular, the saliency mask highlights the parts of either the image (i.e., the features) of the input that constitute most evidence for (or against) the activation of the given output or internal node. This approach uses a technique that exploits the difference analysis of activation maximization on the network nodes producing a relevance vector with the relative importance of the input features.

Another approach f that can be used to solve both the outcome explanation problem and the model inspection in case of image classification with CNN is *VisualBackProp (VBP)* [11]. VBP visualizes which sets of pixels ε_l of the input image contribute most to the prediction. The method uses the intuition that the feature maps contain less and less irrelevant information to the prediction decision when moving deeper into the network. VBP was initially developed as a debugging tool for CNN-based systems for steering self-driving cars. This makes VBP a valuable inspection tool that can be easily used during both training and inference. VBP obtains visualization similar to the LRP approach while achieving orders of magnitude speed-ups.

Finally, with respect to texts, in Reference [65] the authors develop an approach f that incorporates *rationales* as part of the learning process of b. A rationale is a simple subset of words representing a short and coherent piece of text (e.g., phrases), and alone must be sufficient for the prediction of the original text. A rational is the local explanator ε_l and provides the saliency of the text analyzed, i.e., indicates the reason for a certain outcome.

7.2 Agnostic Explanator

In this section, we present the *agnostic* solutions proposed for the *outcome explanation problem* implementing function f such that any type of black box b can be explained. All these approaches are generalizable by definition and return a comprehensible local predictor c_l. Thus, in some cases they can also be employed for diversified data types.

A first attempt of describing an agnostic method f for explaining black box systems b is described in Reference [89] with *ExplainD (Explain Decision)*. This framework f is specially designed for Naive Bayes, SVM and Linear Regression black boxes, but it is in principle generalizable to every black box. ExplainD uses the concept of *additive models* to weight the importance of the features of the input dataset. It provides a graphical explanation of the decision process by visualizing the feature importance for the decisions, the capability to speculate on the effect of changes to the data, and the capability, wherever possible, to drill down and audit the source of the evidence.

Then, in Reference [29], Strumbelj et al. describe a method f for explaining individual predictions of any type of black box b based on Naive Bayes models. The proposed approach exploits notions from *coalitional game theory*, and explains the predictions utilizing the contribution of the

value of different individual features ε_l (see Figure 3). The method is agnostic with respect to the black box used and is tested only on tabular data.

In Reference [98] is presented the *Local Interpretable Model-agnostic Explanations (LIME)* approach f, which does not depend on the type of data, nor on the type of black box b to be opened, nor on a particular type of comprehensible local predictor c_l or explanation ε_l. The main intuition of LIME is that the explanation may be derived locally from the records generated randomly in the neighborhood of the record to be explained, and weighted according to their proximity to it. In their experiments, the authors adopt linear models as comprehensible local predictor c_l returning the importance of the features as explanation ε_l. As black box b, the following classifiers are tested: decision trees, logistic regression, nearest neighbors, SVM, and random forest. A weak point of this approach is the required transformation of any type of data in a binary format that is claimed to be human interpretable: Moreover, in practice the explanation is only provided through linear models and their features importance. References [97] and [96] propose some extensions of LIME with an analysis of particular aspects and cases not overcoming the aforementioned limitations.

A similar approach is presented in Reference [122], where Turner et al. design the *Model Explanation System (MES)* f that augments black box predictions with explanations by using a Monte Carlo algorithm. In practice, they derive a scoring system for finding the best explanation based on formal requirements and consider that the explanations ε_l are simple logical statements, i.e., decision rules. The authors test logistic regression and SVMs as black box b.

An extension of LIME using decision rules as local interpretable classifier c_l is presented in Reference [99]. The *Anchor* f uses a bandit algorithm that randomly constructs the anchors with the highest coverage and respecting a user-specified precision threshold. An anchor explanation is a decision rule that sufficiently tie a prediction locally such that changes to the rest of the features values do not matter, i.e., similar instances covered by the same anchor have the same prediction outcome. Anchor is applied on tabular, images and textual datasets. Reference [110] is an antecedent of Anchor for tabular data only. It adopts a simulated annealing approach that randomly grows, shrinks, or replaces nodes in an expression tree (the comprehensible local predictor c_l). It was meant to return black box decision in forms of "programs."

A recent proposal that overcomes both LIME and Anchor in terms of performance and clarity of the explanations is *LORE (LOcal Rule-based Explanations)* [37]. LORE implements function f by learning a local interpretable predictor c_l on a synthetic neighborhood generated through a genetic algorithm approach. Then, it derives from the logic of c_l, represented by a decision tree, an explanation e consisting of: a decision rule explaining the reasons of the decision, and a set of counterfactual rules, suggesting the changes in the instance's features that lead to a different outcome.

7.3 Explanation via Other Approaches

In Reference [39] is presented a solution for the *outcome explanation problem* not easily classifiable in one of the previous categories. Haufe et al. propose an approach f for determining the origin of neural processes in time or space. Function f transforms nonlinear models (NLM) in terms of multivariate classifiers (the black box b of this work) into linear interpretable models (the comprehensible linear predictor c_l) from which is possible to interpret the features ε_l for a specific prediction. In particular, the f described in Reference [39] enables the neurophysiological interpretation of the parameters of nonlinear models.

8 SOLVING THE MODEL INSPECTION PROBLEM

In this section, we review the methods for opening the black box facing the *model inspection problem* (see Section 4.1). Given a black box solving a classification problem, the inspection problem consists in providing a representation for understanding either how the black box model works or

Table 4. Summary of Methods for Opening Black Boxes Solving the *Model Inspection* Problem

Name	Ref.	Authors	Year	Explanator	Black Box	Data Type	General	Random	Examples	Code	Dataset
NID	[83]	Olden et al.	2002	SA	NN	TAB			✓		
GDP	[8]	Baehrens	2010	SA	AGN	TAB	✓		✓		✓
QII	[24]	Datta et al	2016	SA	AGN	TAB	✓		✓		✓
IG	[115]	Sundararajan	2017	SA	DNN	ANY			✓		✓
VEC	[18]	Cortez et al.	2011	SA	AGN	TAB	✓		✓		✓
VIN	[42]	Hooker	2004	PDP	AGN	TAB	✓		✓		✓
ICE	[35]	Goldstein et al.	2015	PDP	AGN	TAB	✓		✓	✓	✓
Prospector	[55]	Krause et al.	2016	PDP	AGN	TAB	✓		✓		✓
Auditing	[2]	Adler et al.	2016	PDP	AGN	TAB	✓		✓	✓	✓
OPIA	[1]	Adebayo et al.	2016	PDP	AGN	TAB	✓		✓		
–	[136]	Yosinski et al.	2015	AM	DNN	IMG			✓		✓
IP	[108]	Shwartz et al.	2017	AM	DNN	TAB			✓		
–	[137]	Zeiler et al.	2014	AM	DNN	IMG		✓		✓	
–	[112]	Springenberg et al.	2014	AM	DNN	IMG			✓		✓
DGN-AM	[80]	Nguyen et al.	2016	AM	DNN	IMG			✓	✓	✓
–	[72]	Mahendran et al.	2016	AM	DNN	IMG			✓	✓	✓
–	[95]	Radford	2017	AM	DNN	TXT			✓		
–	[143]	Zintgraf et al.	2017	SM	DNN	IMG			✓	✓	✓
VBP	[11]	Bojarski et al.	2016	SM	DNN	IMG			✓		✓
TreeView	[119]	Thiagarajan et al.	2016	DT	DNN	TAB			✓		✓

why the black box returns certain predictions more likely than others. In Reference [105], Seifet et al. provide a survey of visualizations of DNNs by defining a classification scheme describing visualization goals and methods. They found that most papers use pixel displays to show *neuron activations*. As in the previous sections, in the following, we propose a classification based on the type of technique f used to provide the visual explanation of how the black box works. Most papers in this section try to inspect NNs and DNNs. Table 4 summarizes and categorizes these works according to the features described in Section 5.

8.1 Inspection via Sensitivity Analysis

In this section, we review the works solving the *black box inspection problem* by implementing function f using *Sensitivity Analysis (SA)* for the visual representation r. Sensitivity analysis studies the correlation between the uncertainty in the output of a predictor and that one in its inputs [102]. The following methods mostly work on tabular datasets. We emphasize that besides model inspection, sensitivity analysis is also used for outcome explanation (e.g., References [78, 103]).

Sensitivity analysis for "illuminating" the black box was first proposed by Olden in Reference [83], where a visual method for understanding the mechanism of NN is described. In particular, they propose to assess the importance of axon connections and the contribution of input variables by means of sensitivity analysis and *Neural Interpretation Diagram (NID)* to remove not significant connections and improve the network interpretability.

In Reference [8] the authors propose a procedure based on *Gaussian Process Classification (GDP)*, which allows explaining the decisions of any classification method through an explanation vector.

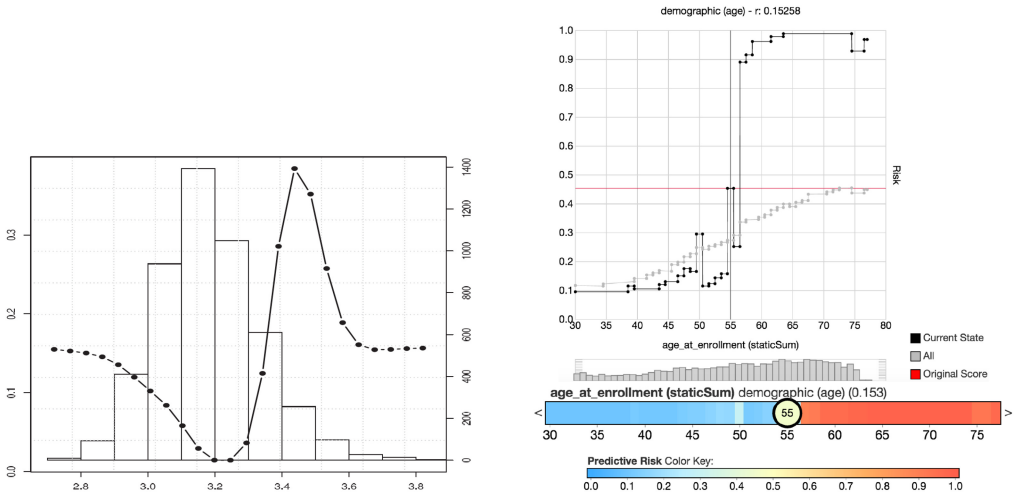

Fig. 14. *(Left)* From Reference [18], VEC curve and histogram for the pH input feature (x-axis) and the respective high-quality wine probability outcome (left of y-axis) and frequency (right of y-axis). *(Right)* From Reference [55], Age at enrollment shown as line plot (top) and partial dependence bar (middle). Color denotes the predicted risk of the outcome.

That is, the procedure f is black box agnostic. The explanation vectors r are visualized to highlight the features that were most influential for the decision of a particular instance. Thus, we are dealing with an inspection for outcome explanation ε_1.

In Reference [24], Datta et al. introduce a set of *Quantitative Input Influence (QII)* measures f capturing how much inputs influence the outputs of black box predictors. These measures provide a foundation for transparency reports r of black box predictors. In practice, the output consists in the feature importance for outcome predictions.

Reference [115] studies the problem of attributing the prediction of a DNN (the black box b) to its input features. Two fundamental axioms are identified: sensitivity and implementation invariance. These axioms guide the design of an attribution method f, called *Integrated Gradients (IG)*, that requires no modification to the original network. Differently from the previous work, this approach is tested on different types of data.

Finally, Cortez in References [18, 19] uses sensitivity analysis based and visualization techniques f to explain black boxes b. The sensitivity measures are variables calculated as the range, gradient, variance of the prediction. Then, the visualizations r realized are barplots for the features importance, and *Variable Effect Characteristic curve (VEC)* [20] plotting the input values (x-axis) versus the (average) outcome responses (see Figure 14(left)).

8.2 Inspection via Partial Dependence

In this section, we report a set of approaches solving the *model inspection problem* by implementing a function f that returns a *Partial Dependence Plot (PDP)*. The partial dependence plot r returned by f is a tool for visualizing the relationship between the response variable and predictor variables in a reduced feature space. All the approaches presented in this section are black box agnostic and are tested on tabular datasets.

In Reference [42], the authors present an approach f aimed at evaluating the importance of non-additive interactions between any set of features. The implementation uses the *Variable Interaction Network (VIN)* visualization generated from the use of ANOVA statistical methodology

(a technique to calculate partial dependence plots). VIN allows to visualize in r the importance of the features together with their interdependencies.

Goldstein et al. provide in Reference [35] a technique f that extends classical PDP named *Individual Conditional Expectation (ICE)* to visualize the model approximated by a black box b that helps in visualizing the average partial relationship between the outcome and some features. ICE plots r improves PDP by highlighting the variation in the fitted values.

In Reference [55], Krause et al. introduce random perturbations on the black box b input values to understand to which extent every feature impact the prediction through a visual inspection r using the PDPs f. The main idea of *Prospector* is to observe how the output varies by varying the input changing one variable at a time. It provides an effective way to understand which are the most important features for a valuable interpretation (see Figure 14(right)).

In Reference [2], the authors propose a method f for *auditing* (i.e., inspecting) black box predictors b, studying to which extent existing models benefit of specific features in the data. This method does not assume any knowledge on the models behavior. In particular, the method f focuses on *indirect influence* and visualizes in r the global inspection through an obscurity vs. accuracy plot (the features are obscured one after the other).

Yet, the dependence of a black box b on its input features is relatively quantified by the procedure f proposed in Reference [1], where the authors present an iterative procedure based on *Orthogonal Projection of Input Attributes (OPIA)*.

8.3 Inspection via Activation Maximization

The works we present in this section solve the *black box inspection problem* by implementing function f using *Activation Maximization (AM)* for finding the activation neurons and the instances (generally images r) characterizing the decision. We remark the very soft difference between these works and those of Section 7.1. Indeed, activation maximization can be used also to find out the pixels and areas on which the black box b focused for taking the decision (e.g., References [11, 143]).

In Reference [136] are proposed two tools for visualizing and interpreting DNNs and for understanding what computations DNNs perform at intermediate layers and which neurons are activated. These tools visualize in r the activations of each layer of a trained CNN during the processing of images or videos. Moreover, they visualize the features of the different layers by regularized optimization in image space. Yosinski et al. found that by analyzing the live activations and observing as they change in correspondence of different inputs, helps to generate an explanation on the DNNs behaviour.

Shwartz-Ziv et al. in Reference [108] proposes f that shows the effectiveness of the *Information Plane* visualization of DNNs r, by highlighting that the empirical error minimization of each stochastic gradient descent phase epoch is always followed by a slow representation compression. This is a very useful result that can be exploited for explaining DNNs.

Similarly to the works presented in 7.1, in Reference [137], Zeiler et al. backtrack the network computations to identify which image patches are responsible for certain neural activations. Simonyan in Reference [109], demonstrated that Zeiler's method can be interpreted as a sensitivity analysis of the network input/output relation.

Explaining the outcome of black boxes is not the main topic of Reference [112], as the paper is more focused in showing that homogeneous and not complicated CNN can reach the state of art performance. However, the intermediate steps of CNN are analyzed introducing a new variant of the "deconvolution approach" for visualizing in r the features learned by b.

The method f proposed in Reference [80] by Nguyen et al. for understanding the inner workings of CNN is based on activation maximization. In practice, it is aimed to retrieve in a visual representation r what each neuron has learned to detect by synthesizing an input image

that highly activates the neuron. This work proposes a way to improve the qualitative activation maximization using a *Deep Generator Network (DGN)*. Thus, DGN-AM generates synthetic images that look almost real and reveals the features learned by each neuron in an interpretable way.

In Reference [72], the authors analyze the visual information contained in DNN internal representations trying to reconstruct the input image from its encoding. They provide a general framework f that is able to invert image representations. As a side effect this method allows to show that several layers in CNN retain photographically accurate information about the image. Moreover, in Reference [73] this work is extended by comparing various family of methods analyzed for understanding "representations." Activation maximization is one of these methods. This task is assessed using the result of the natural pre-images reconstructed by Reference [72]. Moreover, these natural pre-images can be helpful in studying for what black box models are discriminative outside the domain of the images used to train them. In Reference [72], natural pre-images are extracted using an approach based on regularized energy minimization.

Finally, it is worth mentioning that in Reference [95] is presented the discovery that a *single* neuron unit of a DNN can perform alone *sentiment analysis* after the training of the network reaching the same level of performance of strong baselines.

8.4 Inspection via Tree Visualization

We present here a solution for the *black box inspection problem* that adopts an approach f that can be categorized as none of the previous ones. Reference [119] shows the extraction of a visual interpretation r of a DNN using a decision tree. The method *TreeView* f works as follows. Given the black box b as a DNN, it first decomposes the feature space into K (user defined) overlapping factors. Then, it builds a meta feature for each of the K clusters and a random forest that predicts the cluster labels. Finally, it shows a surrogate decision tree from the forest as an approximation of the black box.

9 SOLVING THE TRANSPARENT BOX DESIGN PROBLEM

In this section, we review the approaches designed to solve the classification problem using a transparent method that is locally or globally interpretable on its own, i.e., solving the *transparent box design problem* (see Section 4.1). Table 5 summarizes and categorizes these papers according to the features described in Section 5. We mention that the issue of designing explanations for models that output a ranking of possible values has been considered in the research area of *recommender systems*. For a survey of such approaches, we refer the reader to Reference [120].

9.1 Explanation via Rule Extraction

In this section, we present the most relevant state of the art works solving the *transparent box design problem* by means of comprehensible predictors c based on *rules*. In these cases, c_g is a comprehensible global predictor providing the whole set of rules leading to any possible decision: a *global explanator* ε_g is made available by c_g. All the methods presented in this section work on tabular data.

In Reference [135], the authors propose the approach f named *CPAR (Classification based on Predictive Association Rules),* combining the positive aspects of both associative classification and traditional rule-based classification. Indeed, following the basic idea of FOIL [94], CPAR does not generate a large set of candidates, as in associative classification, and applies a greedy approach for generating rules c_g directly from training data.

In Reference [127], Wang and Rudin propose a method f to extract falling rule lists c_g (see Section 3.3) instead of classical rules. The falling rule lists extraction method f relies on a Bayesian framework.

Table 5. Summary of Methods for Opening Black Boxes Solving the *Transparent Box Desing* Problem

Name	Ref.	Authors	Year	Explanator	Black Box	Data Type	General	Random	Examples	Code	Dataset
CPAR	[135]	Yin et al.	2003	DR	–	TAB					✓
FRL	[127]	Wang et al.	2015	DR	–	TAB			✓	✓	✓
BRL	[66]	Letham et al.	2015	DR	–	TAB			✓		
TLBR	[114]	Su et al.	2015	DR	–	TAB			✓		✓
IDS	[61]	Lakkaraju et al.	2016	DR	–	TAB			✓		
Rule Set	[130]	Wang et al.	2016	DR	–	TAB			✓	✓	✓
1Rule	[75]	Malioutov et al.	2017	DR	–	TAB			✓		✓
PS	[9]	Bien et al.	2011	PS	–	ANY			✓		✓
BCM	[51]	Kim et al.	2014	PS	–	ANY			✓		✓
OT-SpAMs	[128]	Wang et al.	2015	DT	–	TAB			✓	✓	✓

In Reference [66], the authors tackle the problem to build a system for medical scoring that is interpretable and characterized by high accuracy. To this end, they propose *Bayesian Rule Lists (BRL)* f to extract the comprehensible global predictor c_g as a *decision list*. A decision list consists of a series of if-then statements discretizing the whole feature space into a set of simple and directly interpretable decision statements.

A Bayesian approach is followed also in Reference [114]. The authors propose algorithms f for learning *Two-Level Boolean Rules (TLBR)* in Conjunctive Normal Form or Disjunctive Normal Form c_g. Two formulations are proposed. The first one is an integer program whose objective function combines the total number of errors and the total number of features used in the rule. The second formulation replaces the 0-1 classification error with the Hamming distance from the current two-level rule to the closest rule that correctly classifies a sample. In Reference [62], the authors propose a method f exploiting a two-level boolean rule predictor to solve the model explanation, i.e., the transparent approach is used in the reverse engineering approach to explain the black box.

Yet another type of rule is exploited in Reference [61]. Here, Lakkaraju et al. propose a framework f for generating prediction models, which are both interpretable and accuratem, by extracting *Interpretable Decision Sets (IDS)* c_g, i.e., independent if-then rules. Since each rule is independently applicable, decision sets are simple, succinct, and easily to be interpreted. In particular, this approach can learn accurate, short, and non-overlapping rules covering the whole feature space.

Rule Sets are adopted in Reference [130] as comprehensible global predictor c_g. The authors present a Bayesian framework f for learning Rule Sets. A set of parameters is provided to the user to encourage the model to have a desired size and shape to conform with a domain-specific definition of interpretability. A Rule Set consists of a small number of short rules where an instance is classified as positive if it satisfies at least one of the rules. The rule set provides reasons for predictions, and also descriptions of a particular class.

Finally, in Reference [75] an approach f is designed to learn both sparse *conjunctive* and *disjunctive* clause rules from training data through a linear programming solution. The optimization formulation leads the resulting rule-based global predictor c_g (*1Rule*) to automatically balance accuracy and interpretability.

9.2 Explanation via Prototype Selection

In this section, we present the design of a set of approaches f for solving the *transparent box design problem* returning a comprehensible predictor c_g equipped with a human understandable global

explanator function ε_g. A prototype, also referred to with the name artifact or archetype, is an object that is representative of a set of similar instances. A prototype can be an instance x part of the training set $D = \{X, Y\}$, or it can lie anywhere in the space $\mathcal{X}^m \times \mathcal{Y}$ of the dataset D. Having only prototypes among the observed points is desirable for interpretability, but it can also improve the classification error. As an example of a prototype, we can consider the record minimizing the sum of the distances with all the other points of a set (like in K-Medoids) or the record generated averaging the value of the features of a set of points (like in K-Means) [118]. Different definitions and requirements to find a prototype are specified in each work.

In Reference [9], Bien et al. design the transparent *Prototype Selection (PS)* approach f that first seeks for the best prototype (two strategies are proposed), and then assigns the points in D to the label corresponding to the prototype. In particular, they face the problem of recognizing hand written digits. In this approach, every instance can be described by more than one prototype, and more than a prototype can refer to the same label (e.g., there can be more than one prototype for digit zero, more than one for digit one, etc.). The comprehensible predictor c_g provides a global explanation in which every instance must have a prototype corresponding to its label in its neighborhood; no instances should have a prototype with a different label in its neighborhood, and there should be as few prototypes as possible.

Kim et al. in References [51, 52] design the *Bayesian Case Model (BCM)* comprehensible predictor c_l able to learn prototypes by clustering the data and to learn subspaces. Each prototype is the representative sample of a given cluster, while the subspaces are sets of features that are important in identifying the cluster prototype. That is, the global explanator ε_g returns a set of prototypes together with their fundamental features. Possible drawbacks of this approach are the high number of parameters (e.g., number of clusters) and various types of probability distributions that are assumed to be correct for each type of data. Reference [49] proposes an extension of BCM that exploits humans interaction to improve the prototypes. Finally, in Reference [50], the approach is further expanded to include criticisms, where a criticism is an instance that does not fit the model very well, i.e., a counter-example part of the cluster of a prototype.

With respect to prototypes and DNN, [72] already analyzed in Section 8.3 proposes a method for changing the image representations to use only information from the original image and from a generic natural image prior. This task is mainly related to image reconstruction rather than black box explanation, but it is realized with the aim of understanding the example to which the DNN b is related to producing a certain prediction by realizing a sort of artificial image prototype. Therefore it is worth to highlight that there is a significant amount of work in understanding the representation of DNN by means of artifact images [48, 125, 131].

We conclude this section presenting how Reference [30] deals with artifacts in DNNs. Finding a single representative prototype by perturbation, deletion, preservation, and similar approaches has the risk of triggering artifacts of the black box. As discussed in Section 8.4, NN and DNN are known to be affected by surprising artifacts. For example, Reference [60] shows that a nearly-invisible image perturbation can lead a NN to classify an object for another; Reference [81] constructs abstract synthetic images that are classified arbitrarily; Reference [72] finds deconstructed versions of an image that are indistinguishable from the viewpoint of the DNN from the original image, and also with respect to texts [67] inserts typos and random sentences in real texts that are classified arbitrarily. These examples demonstrate that it is possible to find particular inputs that can drive the DNN to generate nonsensical or unexpected outputs. While not all artifacts look "unnatural," nevertheless, they form a subset of images that are sampled with negligible probability when the network is normally operated. In our opinion, two guidelines should be followed to avoid such artifacts in generating explanations for DNNs, and for every black box in general. The first one is

that powerful explanations should, just like any predictor, generalize as much as possible. Second, the artifacts should not be representative of natural perturbations.

9.3 Explanation via Other Approaches

We present here a solution for the *transparent box design problem* that cannot be easily categorized with the previous groups. In Reference [128], Wang et al. propose a method f named *OT-SpAMs* based on oblique tree sparse additive models for obtaining a global interpretable predictor c_g as a decision tree. *OT-SpAMs* divides the feature space into regions using a sparse oblique tree splitting and assigns local sparse additive experts (leaf of the tree) to individual regions. Basically, *OT-SpAMs* passes from complicated trees/linear models to an explainable tree ε_g.

10 CONCLUSION

In this article, we have presented a comprehensive overview of methods proposed in the literature for explaining decision systems based on opaque and obscure machine-learning models. First, we have identified the different components of the family of the explanation problems. In particular, we have provided a formal definition of each problem belonging to that family capturing for each one the proper peculiarity. We have named these black box problems: *model explanation problem*, *outcome explanation problem*, *model inspection problem*, and *transparent box design problem*. Then, we have proposed a classification of methods studied in the literature that take into account the following dimensions: the specific explanation problem addressed, the type of explanator adopted, the black box model opened, and the type of data used as input by the black box model.

As shown in this article, a considerable amount of work has already been done in different scientific communities and especially in the machine-learning and data-mining communities. The first one is mostly focused on describing how the black boxes work, while the second one is more interested into explaining the decisions even without understanding the details on how the opaque decision systems work in general.

The analysis of the literature has led to the conclusion that despite many approaches have been proposed to explain black boxes, some important scientific questions still remain unanswered. One of the most important open problems is that, until now, there is no agreement on what an *explanation* is. Indeed, some works provide as explanation a set of rules, others a decision tree, others a prototype (especially in the context of images). It is evident that the research activity in this field is not providing yet a sufficient level of importance in the study of a general and common formalism for defining an explanation, identifying which are the *properties* that an explanation should guarantee, e.g., soundness, completeness, compactness and comprehensibility. Concerning this last property, there is no work that seriously addresses the problem of quantifying the grade of comprehensibility of an explanation for humans, although it is of fundamental importance. The study of measures able to capture this aspect is challenging, because it also consider aspects like the expertise of the user or the amount of time available to understand the explanation. The definition of a (mathematical) formalism for explanations and of tools for measuring how much an explanation is comprehensible for humans would improve the practical applicability of most of the approaches presented in this article.

Moreover, there are other open research questions related to black boxes and explanations that are starting to be treated by the scientific community and that deserve attention and more investigation. We discuss them in the following.

A common assumption of all categories of works presented in this article is that the features used by the black box decision system are completely known. However, a black box might use additional information besides that explicitly asked to the user. For example, it might link the user's information with different data sources for augmenting the data to be exploited for the prediction.

Therefore, an important aspect to be investigated is to understand how an explanation might also be derived in cases where black box systems make decisions in presence of *latent features*. An interesting starting point for this research direction is the framework proposed in Reference [63] for the evaluation of the prediction models performances on labeled data where the decision of decision-makers is taken in the presence of unobserved features. In principle, latent and unobserved features can be inferred by visible input features. As a consequence, the evidence assigned to these latent features can be redirected to the input one. For example, latent features can be the hidden neurons of a neural networks. Using a backward pass, the explanation can be propagated back to the input variables.

Last, a further interesting point is the fact that explanations are important on their own and predictors might be learned directly from explanations. A starting study of this aspect is Reference [56], which presents a software agent learned to simulate the Mario Bros. game only utilizing explanations rather than the logs of previous plays.

REFERENCES

[1] Julius Adebayo and Lalana Kagal. 2016. Iterative orthogonal feature projection for diagnosing bias in black-box models. *arXiv preprint arXiv:1611.04967*.

[2] Philip Adler, Casey Falk, Sorelle A. Friedler, Gabriel Rybeck, Carlos Scheidegger, Brandon Smith, and Suresh Venkatasubramanian. 2016. Auditing black-box models for indirect influence. In *Proceedings of the IEEE 16th International Conference on Data Mining (ICDM'16)*. IEEE, Springer, 1–10.

[3] Rakesh Agrawal, Ramakrishnan Srikant et al. 1994. Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94)*, Vol. 1215. 487–499.

[4] Yousra Abdul Alsahib S. Aldeen, Mazleena Salleh, and Mohammad Abdur Razzaque. 2015. A comprehensive review on privacy preserving data mining. *SpringerPlus* 4, 1 (2015), 694.

[5] Robert Andrews, Joachim Diederich, and Alan B. Tickle. 1995. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowl.-Based Syst.* 8, 6 (1995), 373–389.

[6] M. Gethsiyal Augasta and T. Kathirvalavakumar. 2012. Reverse engineering the neural networks for rule extraction in classification problems. *Neural Process. Lett.* 35, 2 (2012), 131–150.

[7] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS One* 10, 7 (2015), e0130140.

[8] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert MÃžller. 2010. How to explain individual classification decisions. *J. Mach. Learn. Res.* 11(June 2010), 1803–1831.

[9] Jacob Bien and Robert Tibshirani. 2011. Prototype selection for interpretable classification. *Ann. Appl. Stat.* 5, 4 (2011), 2403–2424.

[10] Marko Bohanec and Ivan Bratko. 1994. Trading accuracy for simplicity in decision trees. *Mach. Learn.* 15, 3 (1994), 223–250.

[11] Mariusz Bojarski, Anna Choromanska, Krzysztof Choromanski, Bernhard Firner, Larry Jackel, Urs Muller, and Karol Zieba. 2016. VisualBackProp: Visualizing CNNs for autonomous driving. *CoRR, Vol. abs/1611.05418* (2016).

[12] Olcay Boz. 2002. Extracting decision trees from trained neural networks. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 456–461.

[13] Leo Breiman, Jerome Friedman, Charles J. Stone, and Richard A. Olshen. 1984. *Classification and Regression Trees*. CRC Press.

[14] Aylin Caliskan-Islam, Joanna J. Bryson, and Arvind Narayanan. 2016. Semantics derived automatically from language corpora necessarily contain human biases. *arXiv preprint arXiv:1608.07187* (2016).

[15] Carolyn Carter, Elizabeth Renuart, Margot Saunders, and Chi Chi Wu. 2006. The credit card market and regulation: In need of repair. *NC Bank. Inst.* 10 (2006), 23.

[16] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1721–1730.

[17] H. A. Chipman, E. I. George, and R. E. McCulloh. 1998. Making sense of a forest of trees. In *Proceedings of the 30th Symposium on the Interface*, S. Weisberg (Ed.). Fairfax Station, VA: Interface Foundation of North America, 84–92.

[18] Paulo Cortez and Mark J. Embrechts. 2011. Opening black box data mining models using sensitivity analysis. In *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM'11)*. IEEE, 341–348.

[19] Paulo Cortez and Mark J. Embrechts. 2013. Using sensitivity analysis and visualization techniques to open black box data mining models. *Info. Sci.* 225 (2013), 1–17.
[20] Paulo Cortez, Juliana Teixeira, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. 2009. Using data mining for wine quality assessment. In *Discovery Science*, Vol. 5808. Springer, 66–79.
[21] Mark Craven and Jude W. Shavlik. 1994. Using sampling and queries to extract rules from trained neural networks. In *Proceedings of the International Conference on Machine Learning (ICML'94)*. 37–45.
[22] Mark Craven and Jude W. Shavlik. 1996. Extracting tree-structured representations of trained networks. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*. 24–30.
[23] David Danks and Alex John London. 2017. Regulating autonomous systems: Beyond standards. *IEEE Intell. Syst.* 32, 1 (2017), 88–91.
[24] Anupam Datta, Shayak Sen, and Yair Zick. 2016. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *Proceedings of the IEEE Symposium on Security and Privacy (SP'16)*. IEEE, 598–617.
[25] Houtao Deng. 2014. Interpreting tree ensembles with intrees. *arXiv preprint arXiv:1408.5456* (2014).
[26] Pedro Domingos. 1998. Knowledge discovery via multiple models. *Intell. Data Anal.* 2, 1–4 (1998), 187–202.
[27] Pedro Domingos. 1998. Occam's two razors: The sharp and the blunt. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD'98)*. 37–43.
[28] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv:1702.08608v2*.
[29] Strumbelj Erik and Igor Kononenko. 2010. An efficient explanation of individual classifications using game theory. *J. Mach. Learn. Res.* 11(Jan. 2010), 1–18.
[30] Ruth Fong and Andrea Vedaldi. 2017. Interpretable explanations of black boxes by meaningful perturbation. *arXiv preprint arXiv:1704.03296* (2017).
[31] Eibe Frank and Ian H. Witten. 1998. Generating accurate rule sets without global optimization. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML'98)*. 144–151.
[32] Alex A. Freitas. 2014. Comprehensible classification models: A position paper. *ACM SIGKDD Explor. Newslett.* 15, 1 (2014), 1–10.
[33] Glenn Fung, Sathyakama Sandilya, and R. Bharat Rao. 2005. Rule extraction from linear support vector machines. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. ACM, 32–40.
[34] Robert D. Gibbons, Giles Hooker, Matthew D. Finkelman, David J. Weiss, Paul A. Pilkonis, Ellen Frank, Tara Moore, and David J. Kupfer. 2013. The CAD-MDD: A computerized adaptive diagnostic screening tool for depression. *J. Clin. Psych.* 74, 7 (2013), 669.
[35] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. 2015. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *J. Comput. Graph. Stat.* 24, 1 (2015), 44–65.
[36] Bryce Goodman and Seth Flaxman. 2016. EU regulations on algorithmic decision-making and a "right to explanation." In *Proceedings of the ICML Workshop on Human Interpretability in Machine Learning (WHI'16)*. Retrieved from http://arxiv. org/abs/1606.08813 v1.
[37] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. 2018. Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820* (2018).
[38] Satoshi Hara and Kohei Hayashi. 2016. Making tree ensembles interpretable. *arXiv preprint arXiv:1606.05390* (2016).
[39] Stefan Haufe, Frank Meinecke, Kai Görgen, Sven Dähne, John-Dylan Haynes, Benjamin Blankertz, and Felix Bießmann. 2014. On the interpretation of weight vectors of linear models in multivariate neuroimaging. *Neuroimage* 87 (2014), 96–110.
[40] Andreas Henelius, Kai Puolamäki, Henrik Boström, Lars Asker, and Panagiotis Papapetrou. 2014. A peek into the black box: Exploring classifiers by randomization. *Data Min. Knowl. Discov.* 28, 5–6 (2014), 1503–1529.
[41] Jake M. Hofman, Amit Sharma, and Duncan J. Watts. 2017. Prediction and explanation in social systems. *Science* 355, 6324 (2017), 486–488.
[42] Giles Hooker. 2004. Discovering additive structure in black box functions. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 575–580.
[43] Johan Huysmans, Karel Dejaeger, Christophe Mues, Jan Vanthienen, and Bart Baesens. 2011. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decis. Supp. Syst.* 51, 1 (2011), 141–154.
[44] U. Johansson, R. König, and L. Niklasson. 2003. Rule extraction from trained neural networks using genetic programming. In *Proceedings of the 13th International Conference on Artificial Neural Networks*. 13–16.
[45] Ulf Johansson, Rikard König, and Lars Niklasson. 2004. The truth is in there-rule extraction from opaque models using genetic programming. In *Proceedings of the FLAIRS Conference*. 658–663.

[46] Ulf Johansson and Lars Niklasson. 2009. Evolving decision trees using oracle guides. In *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM'09)*. IEEE, 238–244.
[47] Ulf Johansson, Lars Niklasson, and Rikard König. 2004. Accuracy vs. comprehensibility in data mining models. In *Proceedings of the 7th International Conference on Information Fusion*, Vol. 1. 295–300.
[48] Hiroharu Kato and Tatsuya Harada. 2014. Image reconstruction from bag-of-visual-words. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 955–962.
[49] Been Kim, Elena Glassman, Brittney Johnson, and Julie Shah. 2015. iBCM: Interactive Bayesian case model empowering humans via intuitive interaction. Technical Report: MIT-CSAIL-TR-2015-010.
[50] Been Kim, Oluwasanmi O. Koyejo, and Rajiv Khanna. 2016. Examples are not enough, learn to criticize! criticism for interpretability. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*. 2280–2288.
[51] Been Kim, Cynthia Rudin, and Julie A. Shah. 2014. The Bayesian case model: A generative approach for case-based reasoning and prototype classification. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*. 1952–1960.
[52] Been Kim, Julie A. Shah, and Finale Doshi-Velez. 2015. Mind the gap: A generative approach to interpretable feature selection and extraction. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*. 2260–2268.
[53] John K. C. Kingston. 2016. Artificial intelligence and legal liability. In *Proceedings of the Specialist Group on Artificial Intelligence Conference (SGAI'16)*. Springer, 269–279.
[54] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730* (2017).
[55] Josua Krause, Adam Perer, and Kenney Ng. 2016. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, 5686–5697.
[56] Samantha Krening, Brent Harrison, Karen M. Feigh, Charles Lee Isbell, Mark Riedl, and Andrea Thomaz. 2017. Learning from explanations using sentiment and advice in RL. *IEEE Trans. Cogn. Dev. Syst.* 9, 1 (2017), 44–55.
[57] R. Krishnan, G. Sivakumar, and P. Bhattacharya. 1999. Extracting decision trees from trained neural networks. *Pattern Recogn.* 32, 12 (1999).
[58] Sanjay Krishnan and Eugene Wu. 2017. PALM: Machine learning explanations for iterative debugging. In *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*. ACM, 4.
[59] Joshua A. Kroll, Joanna Huey, Solon Barocas, Edward W. Felten, Joel R. Reidenberg, David G. Robinson, and Harlan Yu. 2017. Accountable algorithms. *U. Penn. Law Rev.* 165 (2017), 633–705.
[60] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533* (2016).
[61] Himabindu Lakkaraju, Stephen H. Bach, and Jure Leskovec. 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*. ACM, 1675–1684.
[62] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. 2017. Interpretable & explorable approximations of black box models. *arXiv preprint arXiv:1707.01154* (2017).
[63] Himabindu Lakkaraju, Jon Kleinberg, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. 2017. The selective labels problem: Evaluating algorithmic predictions in the presence of unobservables. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 275–284.
[64] Will Landecker, Michael D. Thomure, Luís M. A. Bettencourt, Melanie Mitchell, Garrett T. Kenyon, and Steven P. Brumby. 2013. Interpreting individual classifications of hierarchical networks. In *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM'13)*. IEEE, 32–38.
[65] Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155* (2016).
[66] Benjamin Letham, Cynthia Rudin, Tyler H. McCormick, David Madigan et al. 2015. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. *Ann. Appl. Stat.* 9, 3 (2015), 1350–1371.
[67] Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2017. Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006* (2017).
[68] Zachary C. Lipton. 2016. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490* (2016).
[69] Yin Lou, Rich Caruana, and Johannes Gehrke. 2012. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 150–158.
[70] Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. 2013. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 623–631.
[71] Stella Lowry and Gordon Macpherson. 1988. A blot on the profession. *Brit. Med. J. Clin. Res.* 296, 6623 (1988), 657.

[72] Aravindh Mahendran and Andrea Vedaldi. 2015. Understanding deep image representations by inverting them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5188–5196.
[73] Aravindh Mahendran and Andrea Vedaldi. 2016. Visualizing deep convolutional neural networks using natural pre-images. *Int. J. Comput. Vis.* 120, 3 (2016), 233–255.
[74] Gianclaudio Malgieri and Giovanni Comandé. 2017. Why a right to legibility of automated decision-making exists in the general data protection regulation. *Int. Data Priv. Law* 7, 4 (2017), 243–265.
[75] Dmitry M. Malioutov, Kush R. Varshney, Amin Emad, and Sanjeeb Dash. 2017. Learning interpretable classification rules with boolean compressed sensing. In *Transparent Data Mining for Big and Small Data*. Springer, 95–121.
[76] David Martens, Bart Baesens, Tony Van Gestel, and Jan Vanthienen. 2007. Comprehensible credit scoring models using rule extraction from support vector machines. *Eur. J. Operat. Res.* 183, 3 (2007), 1466–1476.
[77] David Martens, Jan Vanthienen, Wouter Verbeke, and Bart Baesens. 2011. Performance of classification models from a user perspective. *Decis. Support Syst.* 51, 4 (2011), 782–793.
[78] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. 2017. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recogn.* 65 (2017), 211–222.
[79] Patrick M. Murphy and Michael J. Pazzani. 1991. ID2-of-3: Constructive induction of M-of-N concepts for discriminators in decision trees. In *Proceedings of the 8th International Workshop on Machine Learning*. 183–187.
[80] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. 2016. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Proceedings of the Conference on Advances in Neural Information Processing Systems*. 3387–3395.
[81] Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 427–436.
[82] Haydemar Núñez, Cecilio Angulo, and Andreu Català. 2002. Rule extraction from support vector machines. In *Proceedings of the European Symposium on Artificial Neural Networks (ESANN'02)*. 107–112.
[83] Julian D. Olden and Donald A. Jackson. 2002. Illuminating the "black box": A randomization approach for understanding variable contributions in artificial neural networks. *Ecol. Model.* 154, 1 (2002), 135–150.
[84] Fernando E. B. Otero and Alex A. Freitas. 2013. Improving the interpretability of classification rules discovered by an ant colony algorithm. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*. ACM, 73–80.
[85] Gisele L. Pappa, Anthony J. Baines, and Alex A. Freitas. 2005. Predicting post-synaptic activity in proteins with data mining. *Bioinformatics* 21, suppl. 2 (2005), ii19–ii25.
[86] Frank Pasquale. 2015. *The Black Box Society: The Secret Algorithms that Control Money and Information*. Harvard University Press.
[87] Michael J. Pazzani, S. Mani, William R. Shankle et al. 2001. Acceptance of rules generated by machine learning among medical experts. *Methods Info. Med.* 40, 5 (2001), 380–385.
[88] Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. 2008. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 560–568.
[89] Brett Poulin, Roman Eisner, Duane Szafron, Paul Lu, Russell Greiner, David S. Wishart, Alona Fyshe, Brandon Pearcy, Cam MacDonell, and John Anvik. 2006. Visual explanation of evidence with additive classifiers. In *Proceedings of the National Conference on Artificial Intelligence*, Vol. 21.
[90] J. Ross Quinlan. 1987. Generating production rules from decision trees. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI'87)*, Vol. 87. 304–307.
[91] J. Ross Quinlan. 1987. Simplifying decision trees. *Int. J. Man-Mach. Stud.* 27, 3 (1987), 221–234.
[92] J Ross Quinlan. 1993. *C4.5: Programs for Machine Learning*. Elsevier.
[93] J. Ross Quinlan. 1999. Simplifying decision trees. *Int. J. Hum.-Comput. Stud.* 51, 2 (1999), 497–510.
[94] J Ross Quinlan and R. Mike Cameron-Jones. 1993. FOIL: A midterm report. In *Proceedings of the European Conference on Machine Learning*. Springer, 1–20.
[95] Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444* (2017).
[96] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386* (2016).
[97] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Nothing else matters: Model-agnostic explanations by identifying prediction invariance. *arXiv preprint arXiv:1611.05817* (2016).
[98] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 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*. ACM, 1135–1144.

[99] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI'18)*.

[100] Andrea Romei and Salvatore Ruggieri. 2014. A multidisciplinary survey on discrimination analysis. *Knowl. Eng. Rev.* 29, 5 (2014), 582–638.

[101] Salvatore Ruggieri. 2012. Subtree replacement in decision tree simplification. In *Proceedings of the 12th SIAM International Conference on Data Mining*. SIAM, 379–390.

[102] Andrea Saltelli. 2002. Sensitivity analysis for importance assessment. *Risk Anal.* 22, 3 (2002), 579–590.

[103] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. 2017. Evaluating the visualization of what a deep neural network has learned. *IEEE Trans. Neural Netw. Learn. Syst.* 28, 11 (2017), 2660–2673.

[104] Vitaly Schetinin, Jonathan E. Fieldsend, Derek Partridge, Timothy J. Coats, Wojtek J. Krzanowski, Richard M. Everson, Trevor C. Bailey, and Adolfo Hernandez. 2007. Confident interpretation of Bayesian decision tree ensembles for clinical applications. *IEEE Trans. Info. Technol. Biomed.* 11, 3 (2007), 312–319.

[105] Christin Seifert, Aisha Aamir, Aparna Balagopalan, Dhruv Jain, Abhinav Sharma, Sebastian Grottel, and Stefan Gumhold. 2017. Visualizations of deep neural networks in computer vision: A survey. In *Transparent Data Mining for Big and Small Data*. Springer, 123–144.

[106] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. 2016. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *arXiv preprint arXiv:1610.02391* (2016).

[107] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685* (2017).

[108] Ravid Shwartz-Ziv and Naftali Tishby. 2017. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810* (2017).

[109] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013).

[110] Sameer Singh, Marco Tulio Ribeiro, and Carlos Guestrin. 2016. Programs as black-box explanations. *arXiv preprint arXiv:1611.07579* (2016).

[111] Sören Sonnenburg, Alexander Zien, Petra Philips, and G. Rätsch. 2008. POIMs: Positional oligomer importance matrices—understanding support vector machine-based signal detectors. *Bioinformatics* 24, 13 (2008), i6–i14.

[112] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. 2014. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806* (2014).

[113] Irene Sturm, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. 2016. Interpretable deep neural networks for single-trial eeg classification. *J. Neurosci. Methods* 274 (2016), 141–145.

[114] Guolong Su, Dennis Wei, Kush R. Varshney, and Dmitry M. Malioutov. 2015. Interpretable two-level Boolean rule learning for classification. *arXiv preprint arXiv:1511.07361* (2015).

[115] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365* (2017).

[116] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2013).

[117] Hui Fen Tan, Giles Hooker, and Martin T. Wells. 2016. Tree space prototypes: Another look at making tree ensembles interpretable. *arXiv preprint arXiv:1611.07115* (2016).

[118] Pang-Ning Tan et al. 2006. *Introduction to Data Mining*. Pearson Education, India.

[119] Jayaraman J. Thiagarajan, Bhavya Kailkhura, Prasanna Sattigeri, and Karthikeyan Natesan Ramamurthy. 2016. TreeView: Peeking into deep neural networks via feature-space partitioning. *arXiv preprint arXiv:1611.07429* (2016).

[120] Nava Tintarev and Judith Masthoff. 2015. Explaining recommendations: Design and evaluation. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer, 353–382.

[121] Gabriele Tolomei, Fabrizio Silvestri, Andrew Haines, and Mounia Lalmas. 2017. Interpretable predictions of tree-based ensembles via actionable feature tweaking. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 465–474.

[122] Ryan Turner. 2016. A model explanation system. In *Proceedings of the IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP'16)*. IEEE, 1–6.

[123] Wouter Verbeke, David Martens, Christophe Mues, and Bart Baesens. 2011. Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Syst. Appl.* 38, 3 (2011), 2354–2364.

[124] Marina M.-C. Vidovic, Nico Görnitz, Klaus-Robert Müller, and Marius Kloft. 2016. Feature importance measure for non-linear learning algorithms. *arXiv preprint arXiv:1611.07567* (2016).

[125] Carl Vondrick, Aditya Khosla, Tomasz Malisiewicz, and Antonio Torralba. 2013. Hoggles: Visualizing object detection features. In *Proceedings of the IEEE International Conference on Computer Vision*. 1–8.

[126] Sandra Wachter, Brent Mittelstadt, and Luciano Floridi. 2017. Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *Int. Data Priv. Law* 7, 2 (2017), 76–99.
[127] Fulton Wang and Cynthia Rudin. 2015. Falling rule lists. In *Proceedings of the Conference on Artificial Intelligence and Statistics*. 1013–1022.
[128] Jialei Wang, Ryohei Fujimaki, and Yosuke Motohashi. 2015. Trading interpretability for accuracy: Oblique treed sparse additive models. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1245–1254.
[129] Tong Wang. 2017. Multi-value rule sets. *arXiv preprint arXiv:1710.05257* (2017).
[130] Tong Wang, Cynthia Rudin, Finale Velez-Doshi, Yimin Liu, Erica Klampfl, and Perry MacNeille. 2016. Bayesian rule sets for interpretable classification. In *Proceedings of the IEEE 16th International Conference on Data Mining (ICDM'16)*. IEEE, 1269–1274.
[131] Philippe Weinzaepfel, Hervé Jégou, and Patrick Pérez. 2011. Reconstructing an image from its local descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11)*. IEEE, 337–344.
[132] Adrian Weller. 2017. Challenges for transparency. *arXiv preprint arXiv:1708.01870* (2017).
[133] Dietrich Wettschereck, David W. Aha, and Takao Mohri. 1997. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. In *Lazy Learning*. Springer, 273–314.
[134] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International Conference on Machine Learning*. 2048–2057.
[135] Xiaoxin Yin and Jiawei Han. 2003. CPAR: Classification based on predictive association rules. In *Proceedings of the SIAM International Conference on Data Mining*. SIAM, 331–335.
[136] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. 2015. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579* (2015).
[137] Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*. Springer, 818–833.
[138] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530* (2016).
[139] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2921–2929.
[140] Yichen Zhou and Giles Hooker. 2016. Interpreting models via single tree approximation. *arXiv preprint arXiv:1610.09036* (2016).
[141] Zhi-Hua Zhou, Yuan Jiang, and Shi-Fu Chen. 2003. Extracting symbolic rules from trained neural network ensembles. *AI Commun.* 16, 1 (2003), 3–15.
[142] Alexander Zien, Nicole Krämer, Sören Sonnenburg, and Gunnar Rätsch. 2009. The feature importance ranking measure. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 694–709.
[143] Luisa M. Zintgraf, Taco S. Cohen, Tameem Adel, and Max Welling. 2017. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595* (2017).

Received January 2018; revised June 2018; accepted June 2018