A Survey Of Methods For Explaining Black Box Models

Riccardo Guidotti[1,2], Anna Monreale[1], Franco Turini[1], Dino Pedreschi[1], Fosca Giannotti[2]

[1] University of Pisa, {name.surname}@di.unipi.it
[2] ISTI-CNR, Pisa, {name.surname}@isti.cnr.it

Abstract. In the last 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 scarifying 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, delineating explicitly or implicitly its own definition of interpretability and explanation. The aim of this paper 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.

Keywords: Open The Black Box, Explanations, Interpretability, Transparent Models

1 Introduction

The last 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 which 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, 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 will become law in May 2018. An innovative aspect of the GDPR, which has been much debated, are the clauses on automated (algorithmic) individual 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 [31,101,15], everybody agrees that 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 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 safety and on industrial liability. Companies increasingly market services and products by embedding 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. 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.

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 [106,34]: *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 and which kind of model is more interpretable? 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 simultaneously all these aspects 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 [59] highlighting unanswered questions and problems [24]. On the other hand, other works focus on particular aspects like the impact of representation formats on comprehensibility [36], or the interpretability issues in term of advantages and disadvantages of selected predictive models [27]. 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 paper is organized as follows. Firstly, in Section 3 we discuss what interpretability is. Section 2 show which are the motivations for requiring explanation for black box systems by illustrating some real cases. 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 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 [76]. 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" [74], 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). 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 [62].

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 were classified as high risk twice as much as whites who did not re-offend, and white repeat offenders were classified as low risk twice as much as black repeat offenders[3].

Similarly, a study at Princeton [11] 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 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 US 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)[4].

With respect to credit bureaus, it is shown in [12] 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 fifty points among three major US 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 example of bias we can consider [27] and [84]. In these works, the authors show how accurate black box classifiers may result from an accidental artifact in the training data. In [27] 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 enemy photos were taken on overcast days, while friendly photos on sunny days. Similarly, in [84] 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 differ-

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

ences remain between DNNs and humans? Despite the excellent performance of DNNs it seems to be a lot. In [93] 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 [69] a related result is shown. It is easy to produce images that DNNs believe to be recognizable with 99.99% confidence, but which are completely unrecognizable to humans (e.g., labeling white static noise as a tomato). Similarly in [46] visually-indistinguishable training-set are created using DNNs and linear models. With respect to text, in [58] effective methods to attack DNN text classifiers are presented. Experiments 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 [112] 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 what *interpretability* is. 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.

To *interpret* means to give or provide the meaning or to explain and present in understandable terms some concept[5]. 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 [24]. 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. On the other hand, 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

[5] `https://www.merriam-webster.com/`

for unacceptable results, then we do not need an interpretable model, and we can accept any black box.

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 [24].

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, etc. 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 [29,84,87], and no one provides real experiments about the time required to understand an explanation. 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, number of conjunctive terms) [22,32,40,84]. 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 [5,24,27,38]:

- *Interpretability*: to which extent the model and or the prediction are human understandable. The most addressed discussion is related to how the

interpretability can be measured. In [27] 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 predict unseen instances. The accuracy of a model can be measured using various evaluation measures like the accuracy score, the F1-score [95], etc. 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, etc. but with respect to the outcome of the black box which is considered as an oracle.

Moreover, besides these features strictly related to interpretability, yet according to [5,24,27,38] 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 (implicit or explicit) discrimination [85]; 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 non-discrimination which 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 [66,75,99]. 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 [27]. 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.

Data mining and machine learning models should have other important desiderata 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 the parameters or from 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*).

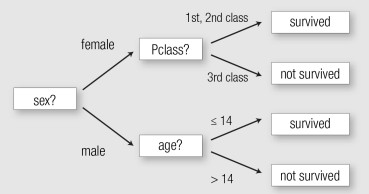

Fig. 1. Example of decision tree.

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* [27,36,84]. 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 1), and leaf nodes representing a class label. Each branch represents a possible outcome [79]. 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 [77,78,26]:

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

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 which 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 [68]; *list of rules* where given an ordered set of rules is considered true the consequent of the first rule which is verified [109]; *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 [102]; *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 [53].

The interpretation of rules and decision trees is different with respect to different aspects [27]. Decision trees are widely adopted for their graphical representation, while rules have a textual representation. The main difference is

Fig. 2. Example of feature importance for a linear model.

that textual representation does not provide immediately information about the more relevant attributes of a rule. On the other hand, 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. On the other hand, 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 [107]. 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 [107]. 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. accuracy, Laplace accuracy, etc.). Then, the outcome of the rules with the average highest weight among the top-k is returned as predicted class [109].

Finally, explanations can also be provided through linear models [47,84]. This can be done by considering and visualizing both the sign and the magnitude of the contribution of the attributes for a given prediction (see Figure 2). 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.

As last remark we point out that in general, when an explanation for a prediction is provided, it is often useful to analyze besides the explanation (satisfied rules, branch of the tree, set of weights, etc.), also instances which 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 [73].

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.

In [32] the complexity is identified by the number of regions, i.e., the parts of the model, for which the boundaries are defined. In [84] as complexity for linear models is adopted the number of non-zero weights, while for decision trees the depth of the tree. In [22] 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 a smaller length may be preferred as it is more interpretable. Similarly, in 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 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 [27]. In this kind of model, a more honest measure could be the average number of conditions evaluated to classify a set of test records [72]. However, this is more a "measure of the explanation" of a list of rules.

Differently from the not flexible representation of decision tree where the prediction of a single record is mutually exhaustive and exclusive, rules characterization contains only significant clauses. As a consequence, an optimal set of rules does not contain any duplicated information, given the fact that an outcome label can appear only one time in the consequent of a set of rules, while in a decision tree it typically comes out more than once. 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 [27]. 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 when using decision trees or rules 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 most understandable data format for humans is the *table* [36]. Since matrices and vectors are the typical data representation used by the vast majority of data mining and machine learning techniques, tables are also easily managed by these algorithms without requiring specific transformations.

Other forms of data which are very common in human daily life are *images* and *texts*. They are perhaps for human brain even more easily understandable than tables. On the other hand, the processing of these data for predictive models requires their transformation into vectors that make them easier to process by algorithms but less interpretable for humans. Indeed, on images and texts, the state of art techniques typically apply predictive models based on super 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 in order 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 sequence data, spatio-temporal data and complex network data that may be used by data mining and machine learning algorithms. However in the literature, to the best of our knowledge, there is no work addressing the interpretability of models for data different from images, texts, and tabular data.

4 Open The Black Box Problems

An accurate analysis and review of the literature lead 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. Details about reverse engineering approaches are discussed at the end of this section. On the other hand, it is used and exploited to build the explanations by most of the works presented in this survey. In the second case, given a dataset of training decision records the task consists in developing an interpretable predictor model together with its explanations.

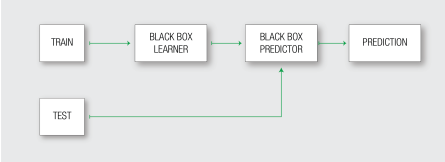

Fig. 3. Classification Problem.

Through a deep analysis of the state of the art we are able to further refine the first category obtaining three different problems. We name them *black box model explanation problem*, *black box outcome explanation problem*, and *black box inspection problem*. We name the second category *transparent box design problem*. All these problems can be formalized 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? 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 3).

A *predictor*, also named model or classifier, is a function $b : \mathcal{X}^m \rightarrow \mathcal{Y}$ where $\mathcal{X}^m$ is the *feature space* with m corresponding to the number of features, and $\mathcal{Y}$ is the *target space*. The feature space $\mathcal{X}$ can correspond to any basic data type like the set of integers $\mathcal{X} = \mathbb{I}^m$, reals $\mathcal{X} = \mathbb{R}^m$, booleans $\mathcal{X} = \{0,1\}^m$, and strings $\mathcal{X} = S^m$, where $S = \Sigma^*$ and $\Sigma = \{a, b, c, \ldots, \}$ is the alphabet (a finite non-empty set of symbols). The feature space $\mathcal{X}$ can also be a complex data type composed of different basic data type. For example, $\mathcal{X} = \mathbb{I} \times \mathbb{R}^2 \times S$ contains an integer feature, two real features and a string feature. On the other hand, the target space $\mathcal{Y}$ (with dimensionality equals to one) contains the different labels (classes or outcomes) and identifies a semantic concept where $\mathcal{Y}$ can be a set of booleans, integers or strings.

A predictor b is the output of a *learner* function $\mathscr{L}_b$ such that $\mathscr{L}_b : (\mathcal{X}^{n \times m} \times \mathcal{Y}^n) \rightarrow (\mathcal{X}^m \rightarrow \mathcal{Y})$. The learner $\mathscr{L}_b$ takes as input a dataset $D = \{X, Y\}$ with n samples where $X \in \mathcal{X}^{n \times m}$ and $Y \in \mathcal{Y}^n$ and returns the predictor b. Given a data record in the feature space $x \in \mathcal{X}^m$, the predictor b can be employed to predict the target value $\hat{y}$, i.e., $b(x) = \hat{y}$.

Typically, in supervised learning [95], a training dataset D_{train} is used for training the learner $\mathscr{L}_b(D_{train})$ which builds the predictor b, and a test dataset D_{test} is used for evaluating the performance of b. Given $D_{test} = \{X, Y\}$, the

evaluation is performed by observing for each couple of data record and target value $(x, y) \in D_{test}$ the number of correspondences between y and $b(x) = \hat{y}$.

In the following we indicate with b a black box predictor belonging to the set of *uninterpretable* data mining and machine learning models. According to Section 3, b is a black box because the reasoning behind the function is not understandable by humans and the outcome returned does not provide any clue for its choice. In real-world applications, b is an opaque classifier resulting from a learning $\mathcal{L}_b$. Similarly, we indicate with c a comprehensible predictor for which is available a global or a local explanation.

The performance of the comprehensible predictor c is generally evaluated by two measures. The *accuracy* is used to evaluate how good are the performance of both the black box predictor b and the comprehensible predictor c. The *fidelity* is employed to evaluate how good is the comprehensible predictor c in mimicking the black box predictor b. Indeed, given a data set $D = \{X, Y\}$ we can apply to each record $x \in X$ both the predictors: *(i)* for the black box b we get the set of predictions $\hat{Y} = \bigcup_{x \in X} b(x)$, while *(ii)* for the comprehensible predictor c we get the set of predictions $\bar{Y} = \bigcup_{x \in X} c(x)$.

Thus, we can evaluate the accuracy of the black box b and of the comprehensible predictor c by comparing the real target values Y against the predicted target values $\hat{Y}$, and $\bar{Y}$ with $accuracy(\hat{Y}, Y)$ and $accuracy(\bar{Y}, Y)$, respectively. Moreover, we can evaluate the behavior of the predictor c with respect to b evaluating the fidelity of c by means of the function $fidelity(\hat{Y}, \bar{Y})$. Note that the *fidelity* score can be calculated by applying the same calculus of the *accuracy* function where as target value is used the prediction $\bar{Y}$ of the black box b instead of the real values Y.

Black Box Model Explanation

Given a black box model solving a classification problem, *the black box explanation problem* consists in providing an interpretable and transparent model which is able to mimic the behavior of the black box and which is also understandable by humans (see Figure 4). In other words, the interpretable model approximating the black box must be globally interpretable. As consequence, we define the black box model explanation problem as follows:

Definition 1 (Black Box Model Explanation). *Given a black box predictor b and a dataset $D = \{X, Y\}$, the* black box model explanation problem *consists in finding a function $f : (\mathcal{X}^m \to \mathcal{Y}) \times (\mathcal{X}^{n \times m} \times \mathcal{Y}^n) \to (\mathcal{X}^m \to \mathcal{Y})$ which takes as input a black box b and a dataset D, and returns a comprehensible global predictor c_g, i.e., $f(b, D) = c_g$, such that c_g is able to mimic the behavior of b, and exists a global explanator function $\varepsilon_g : (\mathcal{X}^m \to \mathcal{Y}) \to \mathcal{E}$ that can derive from c_g a set of explanations $E \in \mathcal{E}$ modeling in a human understandable way the logic behind c_g, i.e., $\varepsilon_g(c_g) = E$.*

A large set of the papers reviewed in this survey describe various designs for the function f to solve the black box explanation problem. The set of explanations E can be modeled for example by a decision tree or by a set of rules [36],

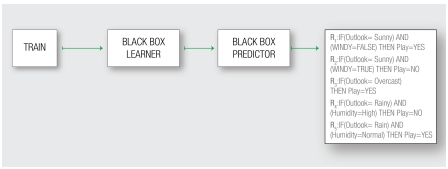

Fig. 4. Black Box Model Explanation Problem.

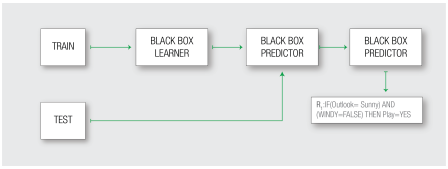

Fig. 5. Black Box Outcome Explanation Problem.

while the comprehensible global predictor c_g is the predictor returning as global explanation ε_g the decision tree or the set of rules.

Black Box Outcome Explanation

Given a black box model solving a classification problem, the *black box outcome explanation problem* consists in providing an interpretable outcome, that is a method for providing an explanation for the outcome of the black box. In other words, the interpretable model must return the prediction together with an explanation about the reasons for that prediction, i.e., the prediction is only locally interpretable. It is not required to explain the whole logic behind the black box but only the reasons for the choice of a particular instance. Consequently, we define the black box outcome explanation problem as:

Definition 2 (Black Box Outcome Explanation). *Given a black box predictor b and a dataset $D = \{X, Y\}$, the* black box outcome explanation problem *consists in finding a function $f : (\mathcal{X}^m \rightarrow \mathcal{Y}) \times (\mathcal{X}^{n \times m} \times \mathcal{Y}^n) \rightarrow (\mathcal{X}^m \rightarrow \mathcal{Y})$ which takes as input a black box b and a dataset D, and returns a comprehensible local predictor c_l, i.e., $f(b, D) = c_l$, such that c_l is able to mimic the behavior of b, and exists a local explanator function $\varepsilon_l : ((\mathcal{X}^m \rightarrow \mathcal{Y}) \times (\mathcal{X}^m \rightarrow \mathcal{Y}) \times \mathcal{X}^m) \rightarrow \mathcal{E}$ which takes as input the black box b, the comprehensible local predictor c_l, and a data record x with features in $\mathcal{X}^m$, and returns a human understandable explanation $e \in \mathcal{E}$ for the data record x, i.e., $\varepsilon_l(b, c_l, x) = e$.*

We report in this survey recent works describing very diversified approaches to implement function f, overcoming the limitations of explaining the whole model (illustrated in Section 6). As an example, in this view of the problem, we can consider that the explanation e_l may be either a path of a decision tree or an association rule [27].

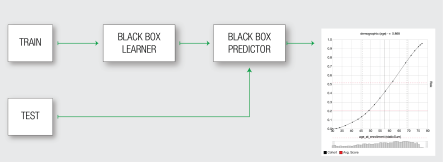

Fig. 6. Black Box Inspection Problem.

Black Box Inspection Problem

Given a black box model solving a classification problem, the *black box inspection problem* consists in providing a representation (visual or textual) for understanding either how the black box model works or why the black box returns certain predictions more likely than others.

Definition 3 (Black Box Inspection Problem). *Given a black box predictor b and a dataset $D = \{X, Y\}$, the* black box inspection problem *consists in finding a function $f : (\mathcal{X} \to \mathcal{Y}) \times (\mathcal{X}^n \times \mathcal{Y}^n) \to \mathcal{V}$ which takes as input a black box b and a dataset D, and returns a visual representation of the behavior of the black box, $f(b, D) = v$ with V being the set of all possible representations.*

For example, the function f may be a technique based on sensitivity analysis that, by observing the changing occurring in the predictions when varying the input of b, returns a set of visualizations (e.g, partial dependence plots [48], or variable effect characteristic curve [17]) highlighting the feature importance for the predictions.

Transparent Box Design Problem

Given a classification problem the *transparent box design problem* consists in providing a model which is locally or globally interpretable on its own.

Definition 4 (Transparent Box Design Problem). *Given a dataset $D = \{X, Y\}$, the* transparent box design problem *consists in finding a learning function $\mathscr{L}_c : (\mathcal{X}^{n \times m} \times \mathcal{Y}) \to (\mathcal{X}^m \to \mathcal{Y})$ which takes as input the dataset $D = \{X, Y\}$ and returns a (locally or globally) comprehensible predictor c, i.e., $\mathscr{L}_c(D) = c$. This implies that there exists a local explanator function ε_l or a global explanator function ε_g (defined as before) that takes as input the comprehensible predictor c and returns a human understandable explanation $e \in \mathcal{E}$ or a set of explanations E.*

For example, the functions $\mathscr{L}_c$ and c may be the decision tree learner and predictor respectively, while the global explanator ε_g may return as explanation a system for following the choices taken along the various branches of the tree, and ε_l may return a textual representation of the path followed according to the decision suggested by the predictor.

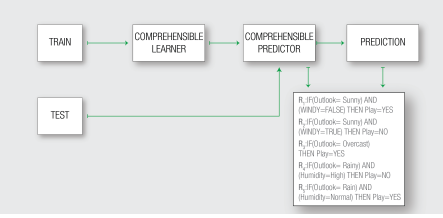

Fig. 7. Transparent Box Design Problem.

Thus, according to our problem definitions, in this survey, when we say 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 that share the same problem definition, while the subsections correspond to the different solutions adopted to develop the explanators. 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 which 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 *black box model explanation problem*. These approaches provide a globally interpretable predictor which is able to mimic the black box. On the other hand, in Section 7 are reviewed the methods solving the *black box 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.

For each of the sections above, we propose a further categorization with respect to the type of explanator adopted. This categorization reflects on the papers grouped into the various subsections:

- *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 set of features used by the black box together with their weight.
- *Salient 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.
- *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 black box 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, in order 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.
- *Neurons Activation (NA).* 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.

In the following, we list all the black boxes opened in the reviewed papers. These black boxes are all supervised learning algorithm designed to solve a classification problem [95].

- *Neural Network (NN).* Inspired by biological neural networks, artificial neural networks learn to do tasks by considering examples. A NN is 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 that it sends.
- *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 TEs. They combine the predictions of different decision trees each one trained on an independent subset 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 SVN is a classifier that 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. The most 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. On the other hand, in image processing Convolutional Neural Networks (CNNs) are typically used.

Moreover, recently *agnostic* approaches for explaining black boxes are being developed. An *Agnostic Explanator (AGN)* is a comprehensible predictor which 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 paper, if not differently specified, we term *agnostic* the approaches defined to explain any type of black box.

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 in order 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 many other types of data are also used like sequences, networks, mobility trajectories, etc. However, they are not used as input in the methods of the papers proposing a solution for opening the black box.

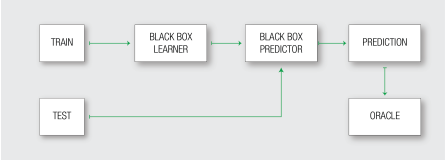

Fig. 8. Reverse Engineering approach: the learned black box predictor is queried with a test dataset to produce an oracle which associate to each record a label which is not real but assigned by the black box.

Table 1 lists the methods for opening and explaining black boxes and summarizes the various fundamental features and characteristics listed so far, together with additional information that we believe could be useful for the reader. The columns *Examples*, *Code* and *Dataset* indicates 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. We point out that Table 1 reports the main references only, while existing extensions or derived works are discussed in the survey. Table 2 reports the legend of Table 1, i.e., the expanded acronym and the meaning of the features in Table 1. Moreover, in order to provide the reader with a useful tool to find a particular set of papers with determined characteristics, Appendix A provides Tables 3, 4 and 5, 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 largely used approach to solve the black box model and outcome explanation problems and the black box inspection problem. We refer to this approach as *reverse engineering* because the black box predictor b is queried with a certain test dataset in order to create an *oracle* dataset that in turn will be used to train the comprehensible predictor (see Figure 8). 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 discovered that DNN built for classification problems on texts and images can be easily fooled (see Section 2). Not human perceptible changes in an image can lead a DNN to label the record as something else. Thus, according

Table 1. Summary of methods for opening and explaining black boxes.

Name	Ref.	Authors	Year	Problem	Explanator	Black Box	Data Type	General	Random	Examples	Code	Dataset
Trepan	[20]	Craven et al.	1996	Model Expl.	DT	NN	TAB	✓				✓
-	[50]	Krishnan et al.	1999	Model Expl.	DT	NN	TAB	✓		✓		✓
DecText	[9]	Boz	2002	Model Expl.	DT	NN	TAB	✓	✓			✓
GPDT	[39]	Johansson et al.	2009	Model Expl.	DT	NN	TAB	✓	✓	✓		✓
Tree Metrics	[14]	Chipman et al.	1998	Model Expl.	DT	TE	TAB					✓
CCM	[23]	Domingos et al.	1998	Model Expl.	DT	TE	TAB	✓	✓			✓
-	[29]	Gibbons et al.	2013	Model Expl.	DT	TE	TAB	✓	✓			
STA	[114]	Zhou et al.	2016	Model Expl.	DT	TE	TAB		✓			
CDT	[87]	Schetinin et al.	2007	Model Expl.	DT	TE	TAB			✓		
-	[32]	Hara et al.	2016	Model Expl.	DT	TE	TAB		✓	✓		✓
TSP	[94]	Tan et al.	2016	Model Expl.	DT	TE	TAB					✓
Conj Rules	[19]	Craven et al.	1994	Model Expl.	DR	NN	TAB		✓			
G-REX	[37]	Johansson et al.	2003	Model Expl.	DR	NN	TAB	✓	✓	✓		
REFNE	[115]	Zhou et al.	2003	Model Expl.	DR	NN	TAB	✓	✓	✓		✓
RxREN	[6]	Augasta et al.	2012	Model Expl.	DR	NN	TAB		✓	✓		✓
SVM+P	[70]	Nunez et al.	2002	Model Expl.	DR	SVM	TAB			✓		✓
-	[28]	Fung et al.	2005	Model Expl.	DR	SVM	TAB			✓		✓
inTrees	[22]	Deng	2014	Model Expl.	DR	TE	TAB			✓		✓
-	[61]	Lou et al.	2013	Model Expl.	FI	AGN	TAB	✓		✓	✓	✓
GoldenEye	[33]	Henelius et al.	2014	Model Expl.	FI	AGN	TAB	✓	✓	✓	✓	✓
PALM	[51]	Krishnan et al.	2017	Model Expl.	DT	AGN	ANY	✓		✓		✓
-	[97]	Tolomei et al.	2017	Model Expl.	FI	TE	TAB			✓		
-	[108]	Xu et al.	2015	Outcome Expl.	SM	DNN	IMG			✓	✓	✓
-	[25]	Fong et al.	2017	Outcome Expl.	SM	DNN	IMG			✓		✓
CAM	[113]	Zhou et al.	2016	Outcome Expl.	SM	DNN	IMG			✓	✓	✓
Grad-CAM	[89]	Selvaraju et al.	2016	Outcome Expl.	SM	DNN	IMG			✓	✓	✓
-	[56]	Lei et al.	2016	Outcome Expl.	SM	DNN	TXT			✓		✓
LIME	[83]	Ribeiro et al.	2016	Outcome Expl.	FI	AGN	ANY	✓	✓	✓	✓	✓
MES	[98]	Turner et al.	2016	Outcome Expl.	DR	AGN	ANY	✓		✓		✓
NID	[71]	Olden et al.	2002	Inspection	SA	NN	TAB			✓		
GDP	[7]	Baehrens	2010	Inspection	SA	AGN	TAB	✓		✓		✓
IG	[92]	Sundararajan	2017	Inspection	SA	DNN	ANY			✓		✓
VEC	[16]	Cortez et al.	2011	Inspection	SA	AGN	TAB	✓		✓		✓
VIN	[35]	Hooker	2004	Inspection	PDP	AGN	TAB	✓		✓		✓
ICE	[30]	Goldstein et al.	2015	Inspection	PDP	AGN	TAB	✓		✓		✓
Prospector	[48]	Krause et al.	2016	Inspection	PDP	AGN	TAB	✓		✓		✓
Auditing	[2]	Adler et al.	2016	Inspection	PDP	AGN	TAB	✓		✓	✓	✓
OPIA	[1]	Adebayo et al.	2016	Inspection	PDP	AGN	TAB	✓		✓		
-	[110]	Yosinski et al.	2015	Inspection	NA	DNN	IMG			✓		✓
TreeView	[96]	Thiagarajan et al.	2016	Inspection	DT	DNN	TAB			✓		✓
IP	[90]	Shwartz et al.	2017	Inspection	NA	DNN	TAB			✓		
-	[81]	Radford	2017	Inspection	NA	DNN	TXT			✓		
CPAR	[109]	Yin et al.	2003	Transp. Design	DR	-	TAB					✓
FRL	[102]	Wang et al.	2015	Transp. Design	DR	-	TAB			✓	✓	✓
BRL	[57]	Letham et al.	2015	Transp. Design	DR	-	TAB			✓		
TLBR	[91]	Su et al.	2015	Transp. Design	DR	-	TAB			✓		✓
IDS	[53]	Lakkaraju et al.	2016	Transp. Design	DR	-	TAB			✓		
Rule Set	[104]	Wang et al.	2016	Transp. Design	DR	-	TAB			✓	✓	✓
1Rule	[64]	Malioutov et al.	2017	Transp. Design	DR	-	TAB			✓		✓
PS	[8]	Bien et al.	2011	Transp. Design	PS	-	ANY			✓		✓
BCM	[44]	Kim et al.	2014	Transp. Design	PS	-	ANY			✓		✓
-	[63]	Mahendran et al.	2015	Transp. Design	PS	-	IMG			✓	✓	✓
-	[47]	Kononenko et al.	2010	Transp. Design	FI	-	TAB			✓		✓
OT-SpAMs	[103]	Wang et al.	2015	Transp. Design	DT	-	TAB			✓	✓	✓

Table 2. Legend of Table 1. In the following are described the features reported and the abbreviations adopted.

Feature	Description
Problem	Model Explanation, Outcome Explanation, Black Box Inspection, Transparent Design
Explanator	DT - Decision Tree, DR - Decision Rules, FI - Features Importance, SM - Saliency Masks, SA - Sensitivity Analysis, PDP - Partial Dependence Plot, NA - Neurons Activation, PS - Prototype Selection
Black Box	NN - Neural Network, TE - Tree Ensemble, SVM - Support Vector Machines, DNN - Deep Neural Network, AGN - AGNostic black box
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 or permutation of the original dataset is required for the explanation
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

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 [84].

Such reverse engineering approach can be classified as *generalizable* or not (or pedagocial vs. decompsitional as described in [65]). 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 9-*(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. On the other hand, 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 9-*(right)*). For example, if an approach is designed to interpret random forest and internally use a concept of distance between trees, then such approach can not be utilized to explain predictions of a NN. A not generalizable approach can not be black box agnostic.

In Table 1 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

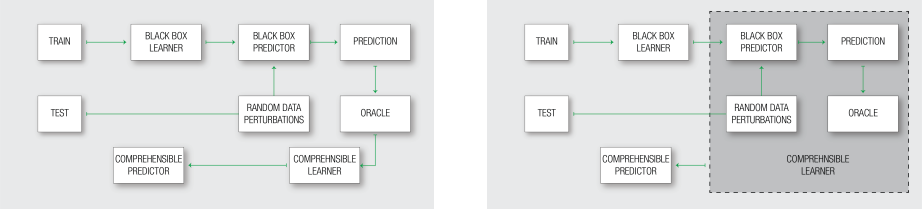

Fig. 9. *(Left)* Generalizable reverse engineering approach: internal peculiarities of the black box are not exploited to build the comprehensible predictor. *(Right)* Not Generalizable reverse engineering approach: the comprehensible predictor is the result of a procedure involving internal characteristics of the black box.

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 Black Box Model Explanation Problem

In this section we review the methods for opening the black box facing the *black box model explanation problem* (see Section 4.1). That is, the proposed methods provide globally interpretable models which are able to mimic the behavior of black boxes and which 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 they use rules. Section 6.3 describes the methods which are designed to work with any type of black box. Finally, Section 6.4 contains the remaining ones.

6.1 Explanation via Single Tree Approximation

In this section we present a set of works addressing the *black box model explanation problem* by implementing in different ways the function f. All the following works adopt a decision tree as comprehensible global predictor c_g, and consequently represent the explanation ε_g with the decision tree itself. Moreover, we point out that all the methods presented in this section work on tabular data.

Explanation of Neural Networks. The following papers describe the implementation of functions f which are able to interpret a black box b consisting in a *Neural Network (NN)* [95] 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 $\hat{y}$. 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. [20]. 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 [95] together with an estimation of the current model fidelity. Another advantage of Trepan with respect to common tree classifiers like ID3 or C4.5 [95] 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 training data.

In [50], 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 [9], Boz describes *DecText*, another procedure that uses a decision tree c to explain neural network black boxes b. The overall procedure recalls Trepan [20] 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 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, than the maximum fidelity and the tree are updated.

In [39] Johansson et al. use *Genetic Programming* to evolve Decision Trees (the comprehensible global predictor c_g), in order to mimic the behavior of a neural network ensemble (the black box 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.

Explanation of Tree Ensembles. Richer collections of trees provide higher performance and less uncertainty in the prediction. On the other hand, it is generally difficult to make sense of the resultant forests. The papers presented in this section describe functions f for approximating a black box model b consisting in *Tree Ensembles (TE)* [95] (e.g. random forests) with a global comprehensible predictor c_g in the form of a decision tree, and explanation ε_g as a the decision tree as before.

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, Chipman et al., in [14] 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.

On the other hand, random data enrichment and model combination are the basis for the *Combined Multiple Model (CCM)* procedure f presented in [23]. 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 [79]) 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 [23], the authors of [29] 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 in order 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). [114] proposes *Single Tree Approximation (STA)*, an extension of [29] 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 [87] present an approach for the probabilistic interpretation of the black box b *Bayesian decision trees ensembles* [10] 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 sub-sequentially refining the training dataset D. Similarly to [14],

also 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 [32], 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. Therefore, 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 method f that works as follows. First, it learns an ATM b generating a number of regions. 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. In particular, to obtain the simpler model an Expectation Maximization algorithm is adopted [95] minimizing the Kullback-Leibler divergence from the ensemble b.

The authors of [94] propose *Tree Space Prototype (TSP)*, an approach f for interpretating tree ensembles (the black box 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 design of the procedure to extract the tree prototypes used for classification.

6.2 Explanation via Rule Extraction

Another commonly used state of the art 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 *black box 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 change accordingly to the type of rules extracted by c_g. Similarly to the previous section, also all the methods presented in this section work on tabular data.

Explanation of Neural Networks. The following papers describe the implementation of functions f which are able to interpret a black box b consisting in a *Neural Network (NN)* [95]. In the literature already exists a survey specialized on techniques extracting rules from neural networks [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 [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 black box model explanation problem. The survey [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. Quality includes accuracy, fidelity, consistency, i.e., different training of the NN extract the rules that lead to the same classification of unseen examples.
- Algorithmic complexity.

A typical paper analyzed in [5] is [19] 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 [40] Johansson et al. exploit *G-REX* [37], 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 corresponds 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. [38] extends G-REX for handling regression problems by generating regression trees, and classification problems by generating fuzzy rules.

In [115] 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 [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.

Explanation of Support Vector Machines. The following papers show implementations of functions f for explaining a black box b consisting in a *Support Vector Machine* (SVM) [95] still returning a comprehensible global predictor c_g consisting in a set of decision rules.

The authors of [70] 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.

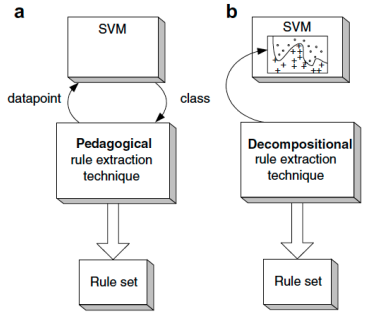

Fig. 10. From [65]: pedagogical *(a)* and decompositional *(b)* rule extraction techniques.

Fung et al., in [28], 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 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 [65] the authors propose a qualitative comparison of the explanations returned by techniques for extraction of rules from SVM black boxes (e.g. SVM+P [70], Fung method [28]) against the redefining of methods designed for explaining neural networks, i.e., C4.5 [95], Trepan [20] and G-REX [37]. How we anticipated in the previous section, the authors delineate the existence of two type of approaches to extract rules: *pedagogical* and *decompositional*. Pedagogical techniques f directly extract rules which relate the inputs and outputs of the predictor (e.g. [20,37]), while decompositional approaches are closely intertwined with the internal structure of the SVM (e.g. [70,28]). We recall that, in Table 1 we identify with the term generalizable the pedagogical approaches.

Explanation of Tree Ensembles. Finally, in [22], 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. In particular, 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 framework 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 *black box 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 [60]. 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 [61]. A case study on health care showing the application of the GAM the refinement is presented in [13]. In particular, this approach is used for the prediction of the pneumonia risk and hospital 30-day readmission.

In [33] the authors present an iterative algorithm f that allows finding features and dependencies exploited by a classifier when producing a prediction. The attributes and the dependencies among the grouped attributes depict the global explanation ε_g. The proposed approach f named *GoldenEye* is based on tabular data randomization (within class permutation, dataset permutation, etc.) and on grouping attributes with interactions have an impact on the predictive power.

In [51], *PALM* is presented (*Partition Aware Local Model*) to implement f. In particular, PALM is a method that is 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 a arbitrarily complex model 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.

6.4 Explanation via Other Approaches

In [97] a solution for the *black box model explanation problem* is presented. It adopts an approach that can not 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.

7 Solving the Black Box Outcome Explanation Problem

In this section we review the methods solving the *black box outcome explanation problem* (see Section 4.1). These methods provide a locally interpretable model which is able to explain the prediction of the black box in understandable terms for humans. This category of approaches using a local point of view with respect to the prediction is becoming the most studied in the last 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 which are able to provide a local explanation for any type of black box.

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)*, i.e. a subset of the original record which is mainly responsible for the prediction. For example, as 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 not generalizable and often strictly tied with the particular type of network, i.e., convolutional, recursive, etc.

The work [108] introduces an *attention based model* f which automatically identifies the contents of an image. The black box is a neural network which 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 11-*left*) for each word in the caption. A similar result is obtained by Fong et al. in [25]. 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 propose to use *saliency maps* as explanations for black boxes to highlight the salient part of the images (see Figure 11-*right*).

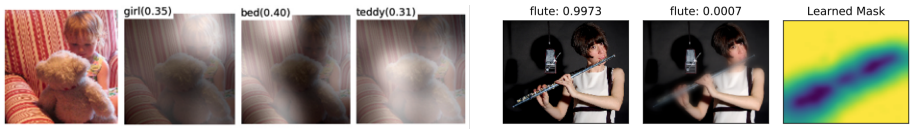

Fig. 11. Saliency Masks for explanation of deep neural network. *(Left)* From [108] the elements of the image highlighted. *(Right)* From [25] the mask and the level of accuracy on the image considering and not considering the learned mask.

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 [113], 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. [89] defines its relaxed generalization Grad-CAM which visualizes the linear combination of a late layer's activations and label-specific weights (or gradients for [113]). All these approaches arbitrarily invoke different 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 [113] or access to intermediate layers [89].

With respect to texts, in [56] the authors develop an approach f which 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 *black box 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, they can be employed for diversified data types.

In [84], Ribeiro et al. present 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. In other words, LIME can return an understandable explanation for the prediction obtained by any black box. 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 which is claimed to be human interpretable. [83] and [82] propose extensions of LIME with an analysis of particular aspects and cases.

A similar approach is presented in [98], 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.

8 Solving the Black Box Inspection Problem

In this section we review the methods for opening the black box facing the *black box 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 why the black box returns certain predictions more likely than others. In [88], 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.

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)*. Sensitivity analysis studies the correlation between the uncertainty in the output of a predictor and that one in its inputs [86]. All the following methods work on tabular datasets.

Sensitivity analysis for "illuminating" the black box was first proposed by Olden in [71] 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 [7] the authors propose a procedure based on *Gaussian Process Classification (GDP)* which allows explaining the decisions of any classification method through an explanation vector. That is, the procedure f is black box agnostic. The explanation vectors 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_l.

In [21], 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 of black

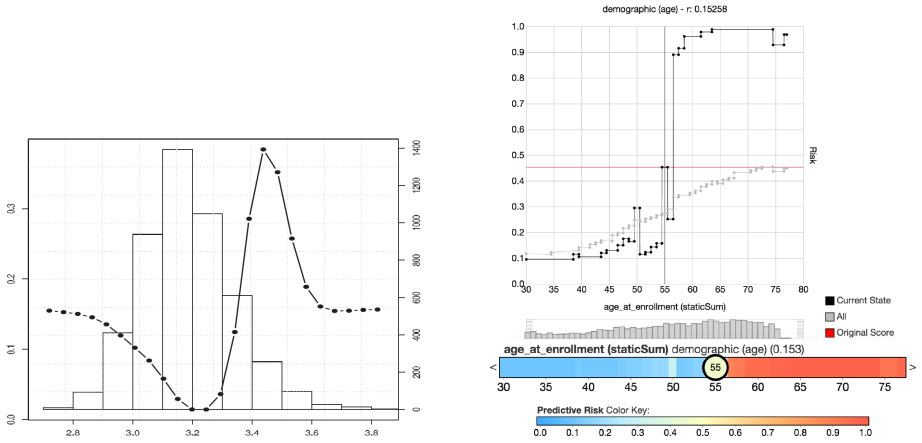

Fig. 12. *(Left).* From [16] VEC curve and histogram for the pH input feature (x-axis) and the respective high quality wine probability outcome (left of y-axis) and frequency (right of y-axis). *(Right).* From [48] Age at enrollment shown as line plot (top) and partial dependence bar (middle). Color denotes the predicted risk of the outcome.

box predictors. In practice, the output consists in the feature importance for outcome predictions.

[92] 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 [16,17] 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 realized are barplots for the features importance, and *Variable Effect Characteristic curve (VEC)* [18] plotting the input values (x-axis) versus the (average) outcome responses (see Figure 12 - *(left)*).

8.2 Inspection via Partial Dependence

In this section we report a set of approaches solving the *black box inspection problem* by implementing a function f which returns a *Partial Dependence Plot (PDP)*. Partial dependence plot 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 [35], the authors present an approach f aimed at evaluating the importance of non-additive interactions between any set of features. The implementa-

tion 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 the importance of the features together with their interdependences. Goldstein et al. provide in [30] a technique f which extends classical PDP named *Individual Conditional Expectation (ICE)* to visualize the model approximated by a black box b that help in visualizing the average partial relationship between the outcome and some features. ICE plots improves PDP by highlighting the variation in the fitted values across the range of covariates. In [48], 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 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 certain prediction ε_l so that it can help in providing a valuable interpretation (see Figure 12 - *(right)*). In [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 the global inspection ε_g 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 [1], where the authors present an iterative procedure based on *Orthogonal Projection of Input Attributes (OPIA)*, for enabling the interpretability of black box predictors.

8.3 Inspection via Other Approaches

In the following, we present solutions for the *black box inspection problem* that adopt an approach f which can be categorized as none of the previous ones. They all refer to DNNs as black box b and are not generalizable.

[110] proposes two tools for visualizing and interpreting DNNs and for understanding what computations DNNs perform at intermediate layers and which neurons activate. These tools visualize the activations of each layer of a trained CNN during the process 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, changing in correspondence of different inputs, helps to generate an explanation on the DNNs behave. [96] shows the extraction of a visual interpretation 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) potentially 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 generates and shows a surrogate decision tree from the forest as an approximation of the black box.

Shwartz-Ziv et al. in [90] showed the effectiveness of the *Information Plane* f visualization of DNNs highlighting that the empirical error minimization of

each stochastic gradient descent phase epoch is always followed by a slow representation compression.

Finally, it is worth mentioning that [81] presents the discovery that a single neuron unit of a DNN can perform alone a sentiment analysis task after the training of the network reaching the same level of performance of strong baselines. Also in [111], Zeiler et al. backtrack the network computations to identify which image patches are responsible for certain neural activations.

9 Solving the Transparent Box Design Problem

In this section we review the approaches designed to solve the classification problem using a transparent method which is locally or globally interpretable on its own, i.e., solving the *transparent box design problem* (see Section 4.1).

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 [109] 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 [80], 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.

Wang and Rudin, in [102] 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.

In [57], the authors tackle the problem to build a system for medical scoring which 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 [91]. 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 [54] the authors propose a method f exploiting a two-level boolean rule predictor to

solve the black box 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 [53]. 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 [104] 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 in order 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 [64] 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 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) [95]. Different definitions and requirements to find a prototype are specified in each work using the prototypes to explain the black box.

In [8], 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 [44,45] 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 set of features which 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 which are assumed to be correct for each type of data. [42] proposes an extension of BCM which exploits humans interaction to improve the prototypes. Finally, in [43] 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, [63] proposes a method b to change the image representations in order to use only information from the original image representation 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. There is a significant amount of work in understanding the representation of DNN by means of artifact images, [41,100,105,111].

We conclude this section presenting how [25] 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 8.3, NN and DNN are known to be affected by surprising artifacts. For example, [52] shows that a nearly-invisible image perturbation can lead a NN to classify an object for another; [69] constructs abstract synthetic images that are classified arbitrarily; [63] finds deconstructed versions of an image which are indistinguishable from the viewpoint of the DNN from the original image, and also with respect to texts [58] 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

In the following we present solutions for the *transparent box design problem* adopting approaches f that can not be categorized as the previous ones. [47] describes a method f based on Naive Bayes aimed to explain individual predictions ε_l of black boxes b. The proposed approach exploits notions from *coalitional game theory*, and explains predictions utilizing the contribution of the value of

different individual features ε_l (see Figure 2). The method is agnostic with respect to the black box used and is tested only on tabular data. Finally, in [103] 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 paper 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 problems: *black box model explanation problem*, *black box outcome explanation problem*, *black box inspection problem* and *transparent box design problem*. Then, we have proposed a classification of methods studied in the literature which 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 paper, 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 conducted in this paper 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 completely ignored the importance of studying 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 also 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 paper.

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.

A common assumption of all categories of works presented in this paper 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 [55] by Lakkaraju et al. for the evaluation of the prediction models performances on labeled data where the decision of decision-makers (either humans or black-boxes) is taken in the presence of unobserved features.

Another open research question is related to providing explanations in the field of *recommender systems*. When a suggestion is provided to a customer, it should come together with the reasons for this recommendation. In [67] the authors define a case-based reasoning approach to generate recommendations with the opportunity of obtaining both the explanation of the recommendation process and of the produced recommendations.

Lastly, 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 [49] that presents a software agent learned to simulate the Mario Bros game only utilizing explanations rather than the logs of previous plays.

Table 3. Summary of methods for opening and explaining black boxes with respect to the problem faced.

Problem	References
Model Explanation	[20], [50], [9], [39], [14], [29], [114], [87], [32], [94], [5], [19], [37], [115], [6], [70], [28], [65], [22], [61], [33], [51], [97]
Outcome Explanation	[108], [25], [113], [89], [56], [83], [98]
Black Box Inspection	[71], [7], [92], [16], [35], [30], [48], [2], [1], [110], [96], [90], [81]
Transparent Design	[109], [102], [57], [91], [53], [104], [64], [8], [44], [63], [47], [103]

A Supplementary Materials

Acknowledgement

This work is partially supported by the European Community's H2020 Program under the funding scheme "INFRAIA-1-2014-2015: Research Infrastructures", grant agreement 654024, *SoBigData*, `http://www.sobigdata.eu`.

References

1. J. Adebayo and L. Kagal. Iterative orthogonal feature projection for diagnosing bias in black-box models. *arXiv preprint arXiv:1611.04967*, 2016.
2. P. Adler, C. Falk, S. A. Friedler, G. Rybeck, C. Scheidegger, B. Smith, and S. Venkatasubramanian. Auditing black-box models for indirect influence. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 1–10. IEEE, 2016.
3. R. Agrawal, R. Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499, 1994.
4. Y. A. A. S. Aldeen, M. Salleh, and M. A. Razzaque. A comprehensive review on privacy preserving data mining. *SpringerPlus*, 4(1):694, 2015.
5. R. Andrews, J. Diederich, and A. B. Tickle. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-based systems*, 8(6):373–389, 1995.
6. M. G. Augasta and T. Kathirvalavakumar. Reverse engineering the neural networks for rule extraction in classification problems. *Neural processing letters*, 35(2):131–150, 2012.
7. D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. MÃžller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.
8. J. Bien and R. Tibshirani. Prototype selection for interpretable classification. *The Annals of Applied Statistics*, pages 2403–2424, 2011.

Table 4. Summary of methods for opening and explaining black boxes with respect to the explanator adopted.

Explanator	References
Decition Tree (DT)	[20], [50], [9], [39], [14], [23], [29], [114], [87], [32], [94], [51], [96], [103]
Decision Rules (DR)	[5], [19], [37], [115], [6], [70], [28], [65], [22], [98], [109], [102], [57], [91], [53], [104], [64]
Features Importance (FI)	[61], [33], [97], [83]
Saliency Mask (SM)	[108], [25], [113], [89], [56]
Sensitivity Analysis (SA)	[71], [7], [92], [16]
Partial Dependence Plot (PDP)	[35], [30], [48], [2], [1]
Neurons Activation (NA)	[110], [90], [81]
Prototype Selection (PS)	[8], [44], [63]

9. O. Boz. Extracting decision trees from trained neural networks. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 456–461. ACM, 2002.
10. L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.
11. A. Caliskan-Islam, J. J. Bryson, and A. Narayanan. Semantics derived automatically from language corpora necessarily contain human biases. *arXiv preprint arXiv:1608.07187*, 2016.
12. C. Carter, E. Renuart, M. Saunders, and C. C. Wu. The credit card market and regulation: In need of repair. *NC Banking Inst.*, 10:23, 2006.
13. R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1721–1730. ACM, 2015.
14. H. Chipman, E. George, and R. McCulloh. Making sense of a forest of trees. *Computing Science and Statistics*, pages 84–92, 1998.
15. G. Comandè. Regulating algorithms regulation? first ethico-legal principles, problems, and opportunities of algorithms. In *Transparent Data Mining for Big and Small Data*, pages 169–206. Springer, 2017.
16. P. Cortez and M. J. Embrechts. Opening black box data mining models using sensitivity analysis. In *Computational Intelligence and Data Mining (CIDM), 2011 IEEE Symposium on*, pages 341–348. IEEE, 2011.
17. P. Cortez and M. J. Embrechts. Using sensitivity analysis and visualization techniques to open black box data mining models. *Information Sciences*, 225:1–17, 2013.
18. P. Cortez, J. Teixeira, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Using data mining for wine quality assessment. In *Discovery Science*, volume 5808, pages 66–79. Springer, 2009.
19. M. Craven and J. W. Shavlik. Using sampling and queries to extract rules from trained neural networks. In *ICML*, pages 37–45, 1994.
20. M. Craven and J. W. Shavlik. Extracting tree-structured representations of trained networks. In *Advances in neural information processing systems*, pages 24–30, 1996.
21. A. Datta, S. Sen, and Y. Zick. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pages 598–617. IEEE, 2016.
22. H. Deng. Interpreting tree ensembles with intrees. *arXiv preprint arXiv:1408.5456*, 2014.

Table 5. Summary of methods for opening and explaining black boxes with respect to the black box explained.

Black Box	References
Neural Network (NN)	[20], [50], [9], [39], [23], [5], [19], [37], [115], [6], [71]
Tree Ensemble (TE)	[14], [23], [29], [114], [87], [32], [94], [22], [97]
Support Vector Machines (SVM)	[70], [28], [65]
Deep Neural Network (DNN)	[108], [25], [113], [89], [56], [92], [110], [96], [90], [81]
Agnostic black box (AGN)	[61], [33], [51], [83], [98], [7], [16], [35], [30], [48], [2], [1]

23. P. Domingos. Knowledge discovery via multiple models. *Intelligent Data Analysis*, 2(1-4):187–202, 1998.
24. F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning. 2017.
25. R. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. *arXiv preprint arXiv:1704.03296*, 2017.
26. E. Frank and I. H. Witten. Generating accurate rule sets without global optimization. 1998.
27. A. A. Freitas. Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter*, 15(1):1–10, 2014.
28. G. Fung, S. Sandilya, and R. B. Rao. Rule extraction from linear support vector machines. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 32–40. ACM, 2005.
29. R. D. Gibbons, G. Hooker, M. D. Finkelman, D. J. Weiss, P. A. Pilkonis, E. Frank, T. Moore, and D. J. Kupfer. The cad-mdd: a computerized adaptive diagnostic screening tool for depression. *The Journal of clinical psychiatry*, 74(7):669, 2013.
30. A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1):44–65, 2015.
31. B. Goodman and S. Flaxman. Eu regulations on algorithmic decision-making and a right to explanation. In *ICML workshop on human interpretability in machine learning (WHI 2016), New York, NY. http://arxiv. org/abs/1606.08813 v1*, 2016.
32. S. Hara and K. Hayashi. Making tree ensembles interpretable. *arXiv preprint arXiv:1606.05390*, 2016.
33. A. Henelius, K. Puolamäki, H. Boström, L. Asker, and P. Papapetrou. A peek into the black box: exploring classifiers by randomization. *Data mining and knowledge discovery*, 28(5-6):1503–1529, 2014.
34. J. M. Hofman, A. Sharma, and D. J. Watts. Prediction and explanation in social systems. *Science*, 355(6324):486–488, 2017.
35. G. Hooker. Discovering additive structure in black box functions. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 575–580. ACM, 2004.
36. J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, 51(1):141–154, 2011.
37. U. Johansson, R. König, and L. Niklasson. Rule extraction from trained neural networks using genetic programming. In *13th International Conference on Artificial Neural Networks*, pages 13–16, 2003.
38. U. Johansson, R. König, and L. Niklasson. The truth is in there-rule extraction from opaque models using genetic programming. In *FLAIRS Conference*, pages 658–663, 2004.
39. U. Johansson and L. Niklasson. Evolving decision trees using oracle guides. In *Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on*, pages 238–244. IEEE, 2009.
40. U. Johansson, L. Niklasson, and R. König. Accuracy vs. comprehensibility in data mining models. In *Proceedings of the seventh international conference on information fusion*, volume 1, pages 295–300, 2004.
41. H. Kato and T. Harada. Image reconstruction from bag-of-visual-words. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 955–962, 2014.

42. B. Kim, E. Glassman, B. Johnson, and J. Shah. ibcm: Interactive bayesian case model empowering humans via intuitive interaction. 2015.
43. B. Kim, O. O. Koyejo, and R. Khanna. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances In Neural Information Processing Systems*, pages 2280–2288, 2016.
44. B. Kim, C. Rudin, and J. A. Shah. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *Advances in Neural Information Processing Systems*, pages 1952–1960, 2014.
45. B. Kim, J. A. Shah, and F. Doshi-Velez. Mind the gap: A generative approach to interpretable feature selection and extraction. In *Advances in Neural Information Processing Systems*, pages 2260–2268, 2015.
46. P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730*, 2017.
47. I. Kononenko et al. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11(Jan):1–18, 2010.
48. J. Krause, A. Perer, and K. Ng. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5686–5697. ACM, 2016.
49. S. Krening, B. Harrison, K. M. Feigh, C. L. Isbell, M. Riedl, and A. Thomaz. Learning from explanations using sentiment and advice in rl. *IEEE Transactions on Cognitive and Developmental Systems*, 9(1):44–55, 2017.
50. R. Krishnan, G. Sivakumar, and P. Bhattacharya. Extracting decision trees from trained neural networks. *Pattern recognition*, 32(12), 1999.
51. S. Krishnan and E. Wu. Palm: Machine learning explanations for iterative debugging. In *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*, page 4. ACM, 2017.
52. A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.
53. H. Lakkaraju, S. H. Bach, and J. Leskovec. 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*, pages 1675–1684. ACM, 2016.
54. H. Lakkaraju, E. Kamar, R. Caruana, and J. Leskovec. Interpretable & explorable approximations of black box models. *arXiv preprint arXiv:1707.01154*, 2017.
55. H. Lakkaraju, J. Kleinberg, J. Leskovec, J. Ludwig, and S. Mullainathan. 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*, pages 275–284. ACM, 2017.
56. T. Lei, R. Barzilay, and T. Jaakkola. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*, 2016.
57. B. Letham, C. Rudin, T. H. McCormick, D. Madigan, et al. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371, 2015.
58. B. Liang, H. Li, M. Su, P. Bian, X. Li, and W. Shi. Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006*, 2017.
59. Z. C. Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.
60. Y. Lou, R. Caruana, and J. Gehrke. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158. ACM, 2012.

61. Y. Lou, R. Caruana, J. Gehrke, and G. Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–631. ACM, 2013.
62. S. Lowry and G. Macpherson. A blot on the profession. *British medical journal (Clinical research ed.)*, 296(6623):657, 1988.
63. A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5188–5196, 2015.
64. D. M. Malioutov, K. R. Varshney, A. Emad, and S. Dash. Learning interpretable classification rules with boolean compressed sensing. In *Transparent Data Mining for Big and Small Data*, pages 95–121. Springer, 2017.
65. D. Martens, B. Baesens, T. Van Gestel, and J. Vanthienen. Comprehensible credit scoring models using rule extraction from support vector machines. *European journal of operational research*, 183(3):1466–1476, 2007.
66. D. Martens, J. Vanthienen, W. Verbeke, and B. Baesens. Performance of classification models from a user perspective. *Decision Support Systems*, 51(4):782–793, 2011.
67. D. McSherry. Explanation in recommender systems. *Artificial Intelligence Review*, 24(2):179–197, 2005.
68. P. M. Murphy and M. J. Pazzani. Id2-of-3: Constructive induction of m-of-n concepts for discriminators in decision trees. In *Proceedings of the eighth international workshop on machine learning*, pages 183–187, 1991.
69. A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436, 2015.
70. H. Núñez, C. Angulo, and A. Català. Rule extraction from support vector machines. In *Esann*, pages 107–112, 2002.
71. J. D. Olden and D. A. Jackson. Illuminating the black box: a randomization approach for understanding variable contributions in artificial neural networks. *Ecological modelling*, 154(1):135–150, 2002.
72. F. E. Otero and A. A. Freitas. Improving the interpretability of classification rules discovered by an ant colony algorithm. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 73–80. ACM, 2013.
73. G. L. Pappa, A. J. Baines, and A. A. Freitas. Predicting post-synaptic activity in proteins with data mining. *Bioinformatics*, 21(suppl_2):ii19–ii25, 2005.
74. F. Pasquale. *The black box society: The secret algorithms that control money and information*. Harvard University Press, 2015.
75. M. J. Pazzani, S. Mani, W. R. Shankle, et al. Acceptance of rules generated by machine learning among medical experts. *Methods of information in medicine*, 40(5):380–385, 2001.
76. D. Pedreshi, S. Ruggieri, and F. Turini. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 560–568. ACM, 2008.
77. J. R. Quinlan. Generating production rules from decision trees. In *ijcai*, volume 87, pages 304–307, 1987.
78. J. R. Quinlan. Simplifying decision trees. *International journal of man-machine studies*, 27(3):221–234, 1987.
79. J. R. Quinlan. *C4. 5: Programs for Machine Learning*. Elsevier, 1993.
80. J. R. Quinlan and R. M. Cameron-Jones. Foil: A midterm report. In *European conference on machine learning*, pages 1–20. Springer, 1993.

81. A. Radford, R. Jozefowicz, and I. Sutskever. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*, 2017.
82. M. T. Ribeiro, S. Singh, and C. Guestrin. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*, 2016.
83. M. T. Ribeiro, S. Singh, and C. Guestrin. Nothing else matters: Model-agnostic explanations by identifying prediction invariance. *arXiv preprint arXiv:1611.05817*, 2016.
84. M. T. Ribeiro, S. Singh, and C. Guestrin. 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*, pages 1135–1144. ACM, 2016.
85. A. Romei and S. Ruggieri. A multidisciplinary survey on discrimination analysis. *The Knowledge Engineering Review*, 29(5):582–638, 2014.
86. A. Saltelli. Sensitivity analysis for importance assessment. *Risk analysis*, 22(3):579–590, 2002.
87. V. Schetinin, J. E. Fieldsend, D. Partridge, T. J. Coats, W. J. Krzanowski, R. M. Everson, T. C. Bailey, and A. Hernandez. Confident interpretation of bayesian decision tree ensembles for clinical applications. *IEEE Transactions on Information Technology in Biomedicine*, 11(3):312–319, 2007.
88. C. Seifert, A. Aamir, A. Balagopalan, D. Jain, A. Sharma, S. Grottel, and S. Gumhold. Visualizations of deep neural networks in computer vision: A survey. In *Transparent Data Mining for Big and Small Data*, pages 123–144. Springer, 2017.
89. R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *arXiv preprint arXiv:1610.02391*, 2016.
90. R. Shwartz-Ziv and N. Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.
91. G. Su, D. Wei, K. R. Varshney, and D. M. Malioutov. Interpretable two-level boolean rule learning for classification. *arXiv preprint arXiv:1511.07361*, 2015.
92. M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*, 2017.
93. C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.
94. H. F. Tan, G. Hooker, and M. T. Wells. Tree space prototypes: Another look at making tree ensembles interpretable. *arXiv preprint arXiv:1611.07115*, 2016.
95. P.-N. Tan et al. *Introduction to data mining*. Pearson Education India, 2006.
96. J. J. Thiagarajan, B. Kailkhura, P. Sattigeri, and K. N. Ramamurthy. Treeview: Peeking into deep neural networks via feature-space partitioning. *arXiv preprint arXiv:1611.07429*, 2016.
97. G. Tolomei, F. Silvestri, A. Haines, and M. Lalmas. 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*, pages 465–474. ACM, 2017.
98. R. Turner. A model explanation system. In *Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on*, pages 1–6. IEEE, 2016.
99. W. Verbeke, D. Martens, C. Mues, and B. Baesens. Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Systems with Applications*, 38(3):2354–2364, 2011.

100. C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba. Hoggles: Visualizing object detection features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8, 2013.
101. S. Wachter, B. Mittelstadt, and L. Floridi. Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *International Data Privacy Law*, 7(2):76–99, 2017.
102. F. Wang and C. Rudin. Falling rule lists. In *Artificial Intelligence and Statistics*, pages 1013–1022, 2015.
103. J. Wang, R. Fujimaki, and Y. Motohashi. Trading interpretability for accuracy: Oblique treed sparse additive models. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1245–1254. ACM, 2015.
104. T. Wang, C. Rudin, F. Velez-Doshi, Y. Liu, E. Klampfl, and P. MacNeille. Bayesian rule sets for interpretable classification. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 1269–1274. IEEE, 2016.
105. P. Weinzaepfel, H. Jégou, and P. Pérez. Reconstructing an image from its local descriptors. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 337–344. IEEE, 2011.
106. A. Weller. Challenges for transparency. *arXiv preprint arXiv:1708.01870*, 2017.
107. D. Wettschereck, D. W. Aha, and T. Mohri. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. In *Lazy learning*, pages 273–314. Springer, 1997.
108. K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.
109. X. Yin and J. Han. Cpar: Classification based on predictive association rules. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 331–335. SIAM, 2003.
110. J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.
111. M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.
112. C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.
113. B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.
114. Y. Zhou and G. Hooker. Interpreting models via single tree approximation. *arXiv preprint arXiv:1610.09036*, 2016.
115. Z.-H. Zhou, Y. Jiang, and S.-F. Chen. Extracting symbolic rules from trained neural network ensembles. *Ai Communications*, 16(1):3–15, 2003.