IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 11, 2021, accepted September 21, 2021, date of publication September 29, 2021, date of current version October 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3116481

Explainable Artificial Intelligence for Tabular Data: A Survey

MARIA SAHAKYAN[1,2], ZEYAR AUNG[1], (Senior Member, IEEE), AND TALAL RAHWAN[2]
[1]Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates
[2]Department of Computer Science, New York University Abu Dhabi (NYUAD), Abu Dhabi, United Arab Emirates

Corresponding author: Talal Rahwan (talal.rahwan@nyu.edu)

The work of Maria Sahakyan was supported by Khalifa University, Abu Dhabi, UAE, by providing a Ph.D. Scholarship and Research Facilities.

ABSTRACT Machine learning techniques are increasingly gaining attention due to their widespread use in various disciplines across academia and industry. Despite their tremendous success, many such techniques suffer from the ''black-box'' problem, which refers to situations where the data analyst is unable to explain why such techniques arrive at certain decisions. This problem has fuelled interest in Explainable Artificial Intelligence (XAI), which refers to techniques that can easily be interpreted by humans. Unfortunately, many of these techniques are not suitable for *tabular data*, which is surprising given the importance and widespread use of tabular data in critical applications such as finance, healthcare, and criminal justice. Also surprising is the fact that, despite the vast literature on XAI, there are still no survey articles to date that focus on tabular data. Consequently, despite the existing survey articles that cover a wide range of XAI techniques, it remains challenging for researchers working on tabular data to go through all of these surveys and extract the techniques that are suitable for their analysis. Our article fills this gap by providing a comprehensive and up-to-date survey of the XAI techniques that are relevant to tabular data. Furthermore, we categorize the references covered in our survey, indicating the type of the model being explained, the approach being used to provide the explanation, and the XAI problem being addressed. Our article is the first to provide researchers with a map that helps them navigate the XAI literature in the context of tabular data.

INDEX TERMS Black-box models, explainable artificial intelligence, machine learning, model interpretability.

I. INTRODUCTION

The recent surge of interest in Machine Learning (ML) and Artificial Intelligence (AI) has spurred a wide array of models designed to make decisions in a variety of domains, including healthcare [1]–[3], financial systems [4]–[7], and criminal justice [8]–[10], just to name a few. When evaluating alternative models, it may seem natural to prefer those that are more accurate. However, the obsession with accuracy has led to unintended consequences, as developers often strove to achieve greater accuracy at the expense of interpretability by making their models increasingly complicated and harder to understand [11]. This lack of interpretability becomes a serious concern when the model is entrusted with the power to make critical decisions that affect people's well-being. These concerns have been manifested by the European Union's recent General Data Protection Regulation, which guarantees a *right to explanation*, i.e., a right to understand the rationale behind an algorithmic decision that affects individuals negatively [12]. To address these issues, a number of techniques have been proposed to make the decision-making process of AI more understandable to humans. These ''*Explainable AI*'' techniques (commonly abbreviated as XAI) are the primary focus of this survey.

Before delving into the literature of XAI, let us describe key terms that will be used throughout this article. First, it is important to distinguish between the terms *interpretation* and *explanation*, since they are often used interchangeably when in fact, they have different meanings. In particular, the former involves expressing abstract concepts using human-understandable terms, while the latter involves pointing out the features that have contributed to the outcome of a particular instance [13]. Admittedly, these definitions are informal, as there are no formal definitions in

The associate editor coordinating the review of this manuscript and approving it for publication was K. Kotecha.

		Agnostic	SVM	ANN	DNN	TE
Feature importance	Sensitivity analysis	[83-86], [103]		[96]		
	Partial dependence plots	[83],[87-90], [103], [104]				
	Local surrogate models	[24], [26-30], [37], [53], [54]				
	Cooperative game theory	[35-37], [63], [86], [90]			[37]	[39], [65]
	Tree approximation	[29]		[124]		
	Clustering	[26], [39]				
	Generalised additive models	[109], [110]				
	Repeated runs of model/technique	[32],[53],[54],[92]				
	Backpropagation				[37], [38], [64]	
	Graphical explanations	[23]				
	Other	[33], [34], [113]	[142]			[40], [42], [97]
Feature interactions	Iterative search					[148]
	Partial dependence plots	[87], [91]				[98]
	Projections mapping					[99]
	Cooperative game theory					[39], [65]
	Features grouping	[111],[118]				
	Generalised additive models	[110]				
	Other	[34], [113]				
Decision rules	Support vectors		[142-145]			
	Search algorithm	[25]	[142]			[130], [143]
	Local surrogate models	[29], [31]				
	Genetic programming	[31]		[153-155]		
	Clustering		[139],[140],[160]			
	Optimization	[115],[120]	[141]	[158]		
	Reverse engineering			[137]		
	Other	[56], [57]	[146],[147]	[135],[136], [156-159],[161]		[95], [148]
Simplified model	Tree approximation	[112],[114]		[121-125]	[94], [126],[127]	[129-131]
	Genetic programming			[122], [125]		
	Optimization			[124]	[127]	
	Clustering					[128]
	Input data modifications					[129], [130]
	Other					[132-134]
Counterfactuals	Genetic programming	[31]*,[80]*,[82]*				
	Gradient-based algorithm			[67]*,[69]*		
	Local surrogate models	[78]*				
	Autoencoder	[47]*		[70]*		
	Distance minimization	[81]*		[66]*		
	K-nearest neighbours and ANOVA	[76]				
	Sampling	[79]*				
	Mixed integer programming			[71]*		
	Other	[68]*,[77]*				

model explanation model inspection outcome explanation

FIGURE 1. A map of the references covered in this survey. Columns indicate the type of the model being explained, which can be SVM (Support Vector Machines), ANN (Artificial Neural Networks), DNN (Deep Neural Network), TE (Tree Ensembles), or Agnostic (i.e., they are not restricted to a particular model). Rows indicate the approach being used to provide the explanation, while colours indicate the black-box explanation problem being addressed. Furthermore, the rows are grouped into four categories, indicating the form of the explanation being provided, which is either Feature importance (which quantifies the relative importance of different features), Feature interaction (which provides insights into the way in which certain features interact with one another), If-then rules (which mimic the behaviour of the model being explained, using simple rules that are easily understandable), simplified models (which mimic the behaviour of the model being explained while being easier to understand), and counterfactuals (which identifies the features that need to be changed in order to obtain certain outcome). "*" indicates that the technique is limited to explain classification models only.

the XAI literature to date [14]. Another term that is frequently mentioned in the literature is **transparency**. According to Lipton's taxonomy [15], there are three notions of model transparency:

- **Simulatability**, reflecting the degree to which the user can contemplate the model in its entirety;
- **Decomposability**, reflecting the degree to which each input, parameter, and calculation can be explained intuitively;
- **Algorithmic transparency**, reflecting the degree to which the inner workings of the learning algorithm can be understood.

For example, rule-based models [16] are considered transparent since they use a series of if-then rules that can easily be understood without the need for any further explanation. Unlike transparent models, *black-box* models are those that do not explain their predictions in a way that humans can understand [17]. Examples of black-box models include artificial neural networks [18] and gradient boosting [19]. Although black-box models are hard to interpret by humans, they tend to have higher prediction accuracy compared to their transparent counterparts. This trade-off between accuracy and transparency gives raise to the *black-box explanation problem*, which involves explaining the rationale behind the decisions made by black-box models. By providing such explanations, one can continue to use highly-accurate black-box models without sacrificing transparency. According to Guidotti *et al.* [20], black-box explanation problems can be categorized into the following:

- **Model explanation problems**, which requires explaining the underlying logic behind a black-box model. This is typically done by approximating the black-box behaviour using an alternative model that is more transparent and interpretable.
- **Model inspection problems**, which requires providing visual or textual explanations of certain properties of the underlying model or its outcome, with the goal being to understand how internally the black box behaves when the input is changed.
- **Outcome explanation problems**, which requires explaining the model's outcome given an instance of interest, by either explaining how the outcome was generated or explaining how the outcome can be changed using counterfactual analysis.

Each of the above problems can be addressed using different types of techniques, which can be classified as follows:

- **Model-specific techniques**, which exploit the parameters and features of the model they are designed to explain [21]. The power of such techniques stems from their ability to access the model internals such as weights or structure, but this power comes at a price, since they cannot readily be generalised to other models.
- **Model-agnostic techniques**, which in principle can be used on any machine learning model to provide *post-hoc* explanations, i.e., explanations that are generated after the model has been trained [22]. The disadvantage of these techniques is that they cannot take advantage of model internals, since they are only capable of analysing input-out pairs.

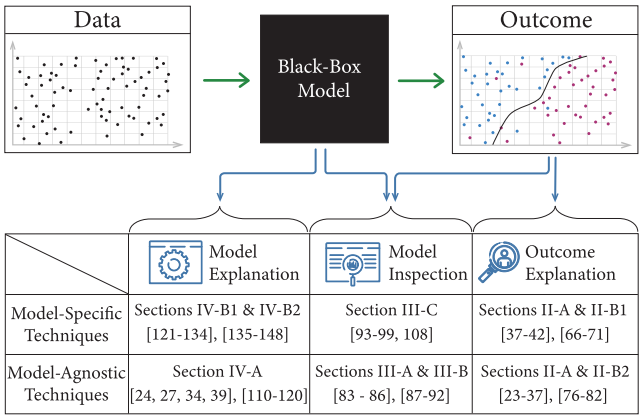

FIGURE 2. An overview of the studies covered in this survey. Explainable AI addresses three problems: (i) *Model Explanation*, which requires explaining the underlying logic behind a black-box model; (ii) *Model inspection*, which requires providing visual or textual explanations of certain properties of the underlying model or its outcome, and (iii) *Outcome explanation*, which requires explaining the model's outcome given an instance of interest, to justify the model's decision. Each of these problems can be addressed with different types of techniques, which can be classified into: (i) *model-specific techniques*, which are tailored for a particular type of models and cannot readily be generalised to other models, and (ii) *model-agnostic techniques*, which in principle can be used on any machine learning model. The illustration indicates the sections that cover each problem and each type of technique, as well as the references covered in each section.

Many XAI techniques [43]–[46], regardless of whether they are model-specific or model-agnostic, are not suitable for *tabular data*, i.e., data in which all records share the same features and every such feature is either numerical, categorical, or Boolean. Importantly, techniques that are tailored for images or text cannot readily be applied to tabular data [47]. This is because tabular data has unique characteristics such as potential dependencies and correlations between the features, the presence of both continuous and categorical features, and the temporal aspect of the data set. These characteristics are missing from image and text data sets, since they consist primarily of words and pixels in a two-dimensional space. Thus, it is important to clearly distinguish the techniques that are compatible with tabular data. Motivated by this observation, our article provides a comprehensive and up-to-date survey of the XAI techniques that are relevant to tabular data, and discusses a variety of directions from which the XAI problem has been approached in this context. Careful attention has been given to ensure that the intuitions behind the different techniques are presented in an accessible and clear manner, along with illustrated examples.

As such, the reader is not assumed to be an expert in machine learning and artificial intelligence. To help the reader navigate this literature, we provide a map that categorizes the references covered in our survey, see Figure 1. Here, columns indicate the type of the model being explained, rows

TABLE 1. A summary of the XAI techniques covered in Section II-A, all of which are designed for outcome explanation.

Name	Ref.	Year	Type	Underlying mechanism
ExplainD	[23]	2006	Agnostic	Graphical explanations based on additive models.
LIME	[24]	2016	Agnostic	Local linear models applied on random perturbations of the data and the predictions.
Anchors	[25]	2018	Agnostic	Reinforcement learning and graph search applied on random perturbations of the data and the predictions.
DLIME	[26]	2019	Agnostic	LIME combined with hierarchical clustering.
ILIME	[27]	2019	Agnostic	LIME, but with each perturbed instance weighed based on its influence on, and distance from, the instance to be explained.
ALIME	[28]	2019	Agnostic	LIME combined with an autoencoder.
Doctor XAI	[29]	2020	Agnostic	The idea behind LIME combined with a decision tree.
MAPLE	[30]	2018	Agnostic	Local linear modelling techniques combined with random forests.
LORE	[31]	2018	Agnostic	Genetic Algorithm.
LOCO	[32]	2018	Agnostic	Scoring of input features.
MES	[33]	2016	Agnostic	Monte Carlo algorithm.
MFI	[34]	2016	Agnostic	Feature ranking measure.
IME	[35]	2010	Agnostic	Coalitional game theory and the Shapley value.
-	[36]	2014	Agnostic	Coalitional game theory and sensitivity analysis.
Kernel SHAP	[37]	2017	Agnostic	LIME combined with the Shapley value.
DeepLIFT	[38]	2017	Specific	Assigning importance scores to inputs in a neural network.
Deep SHAP	[37]	2017	Specific	DeepLIFT combined with the Shapley value.
Tree SHAP	[39]	2020	Specific	SHAP, adopted for tree-based ML models.
Saabas	[40]	2014	Specific	Extracting explanations from a decision path in a decision tree.
-	[41]	2014	Specific	Transformation of backward models to forward models.
LionForest	[42]	2019	Specific	Explaining the decisions of random forests via unsupervised learning techniques and similarity metrics.

indicate the approach being used to provide the explanation, and colours indicate the black-box explanation problem being addressed. Furthermore, rows are grouped based on the form of the explanation being provided. This way, the reader can easily identify the reference(s) of interest.

Figure 2 provides an overview of this survey. In particular, for every black-box explanation problem (be it model explanation, model inspection, or outcome explanation) and every type of technique introduced in the literature (be it is model-specific or model-agnostic), the figure specifies the corresponding section(s) as well as the references covered therein.

II. XAI FOR OUTCOME EXPLANATION

In this section, we will review the XAI techniques that are proposed to address the outcome explanation problem. Recall that this problem involves explaining the rationale behind the model's decision for any given instance, without necessarily explaining the logic of the underlying model. The majority of techniques in this literature are model-agnostic. In contrast, only a few techniques are model-specific, and some of those are used as a foundation for their model-agnostic counterparts. In light of this observation, we decided to present both categories (i.e., model-agnostic and model-specific) in the same section (Section II-A), rather than dividing them into two distinct subsections. Section II-B presents counterfactual explanations, which are example-based explanations that specify how the instance can be modified such that its classification changes into another, desirable class. This section is divided into two subsections: Section 2 presents model-specific techniques, while Section 3 presents model-agnostic techniques.

A. EXPLAINING HOW THE OUTCOME WAS GENERATED

Table 1 provides a summary of the techniques that will be discussed in this section.

One of the first attempts to build a model-agnostic technique for explaining black-box model outcomes is the ExplainD framework, proposed by Poulin *et al.* [23]. It uses the concepts of additive models to weigh the importance of the input features in the decision of the classifier. Additive models are a generalisation of the multivariate regression where, instead of a single coefficient for each variable, a non-parametric function is used. In additive models, the outcome can be expressed as a weighted sum of independent variables so that the portion of the outcome contributed by

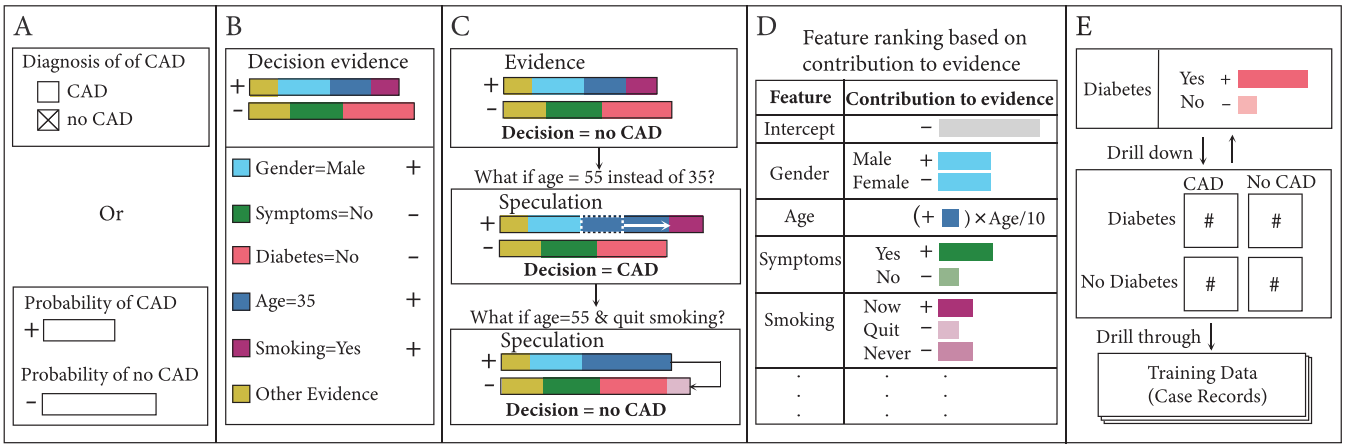

FIGURE 3. An illustration of the ExplainD framework. This illustration describes the intuition behind the ExplainD framework using an example of the diagnosis of obstructive Coronary Artery Disease (CAD). In this example, a physician uses an additive classifier to predict whether a 35-year-old man has CAD. A. The decision of the classifier is that the man has no CAD because the probability of CAD is smaller than the probability of no CAD. B. An illustration showing how the evidence for not having CAD is stronger than the evidence for having CAD (see how the rectangle marked by "–" is longer than the one marked by "+"). Each evidence consists of multiple additive components corresponding to different features. C. Decision speculation showing how the classification would change if the feature values were different. D. Ranking of the features based on their impact on the classifier's decision. E. To verify whether the classifier's decision match expectations, the user can audit the relationship between the decision label and any given feature. For example, if the feature is "Diabetes", then the user can slice the training data by label (CAD and No CAD) and feature value (Diabetes and No Diabetes) and examine the data summary in each quadrant to understand how the classifier determined the relationship between Diabetes and CAD.

one independent variable does not depend on the value of any other independent variables. ExplainD is a graphical explanation framework providing visualisations of the predicted classification (Fig. 3A), the relative strength of potential decisions and the contribution of each feature to those decisions (Fig. 3B), "what-if?" analysis by changing feature values (Fig. 3C), feature evidence in the context of the overall classifier (Fig. 3D) as well as the possibility to access the data supporting the evidence of feature contributions (Fig. 3E).

Compared to previously proposed techniques for explaining particular model decisions, e.g., belief networks [48] or naive Bayesian classifiers [49], [50], ExplainD produces explanations for a broader range of additive models, covering also logistic regression and Support Vector Machines (SVM). An extension of ExplainD was proposed by Robnik and Kononenko [51], which covers not only additive classifiers but also probabilistic models.

A decade after the development of ExplainD, a groundbreaking model-agnostic technique was proposed by Ribeiro *et al.* [24] called LIME, which stands for Local Interpretable Model-agnostic Explanations. This technique is capable of explaining the predictions of any classifier and works as follows. Local surrogate models—models that are interpretable and used to explain individual predictions of ML models—are trained to approximate the predictions of the underlying model. The goal is to understand why the ML model made a certain prediction. This is achieved by creating a new data set from perturbed data points around the instance of interest by sampling from the original data and its corresponding predictions of the ML model. On this new data set, LIME trains a local surrogate model whereby each sampled instance is weighted by its proximity to the instance of interest. An example is illustrated in Figure 4.

One of the advantages of LIME is that its computational complexity allows for running it on thousands of instances in a reasonable time. As a result, if it is used to explain a representative sample of instances, then these explanations would constitute a global explanation of the model under consideration. On the other hand, LIME has a number of limitations. First, since the generated explanations are based on random perturbations, the outcome may lack stability [52]. Second, LIME is sensitive to the data set dimensionality, and when used on a relatively large number of features (e.g., 100 or more), the local explanation is unable to discriminate between relevant and irrelevant features, which may result in a poor performance [53], [54]. A number of techniques have been recently proposed based on ideas that are similar to the ones used in LIME, some of which are designed to address the aforementioned limitations. In particular:

- Shankaranarayana and Runje [28] use an autoencoder in the LIME framework as the weighing function and empirically demonstrate that this improves the stability of LIME.
- Zafar and Khan. [26] propose another technique to improve the stability of LIME, which uses hierarchical clustering instead of random perturbations to group the training data and then selects the cluster that is most relevant to the instance being explained.
- Visani *et al.* [53], [54] propose two stability indices, both of which are calculated by repeated calls of LIME. While these indices do not address LIME's stability issues, they help the user to understand any potential instability in their obtained results.
- Elshawi *et al.* [27] weigh each instance in the permuted data set based on (i) its influence in a linear model on the instance to be explained, and (ii) its distance

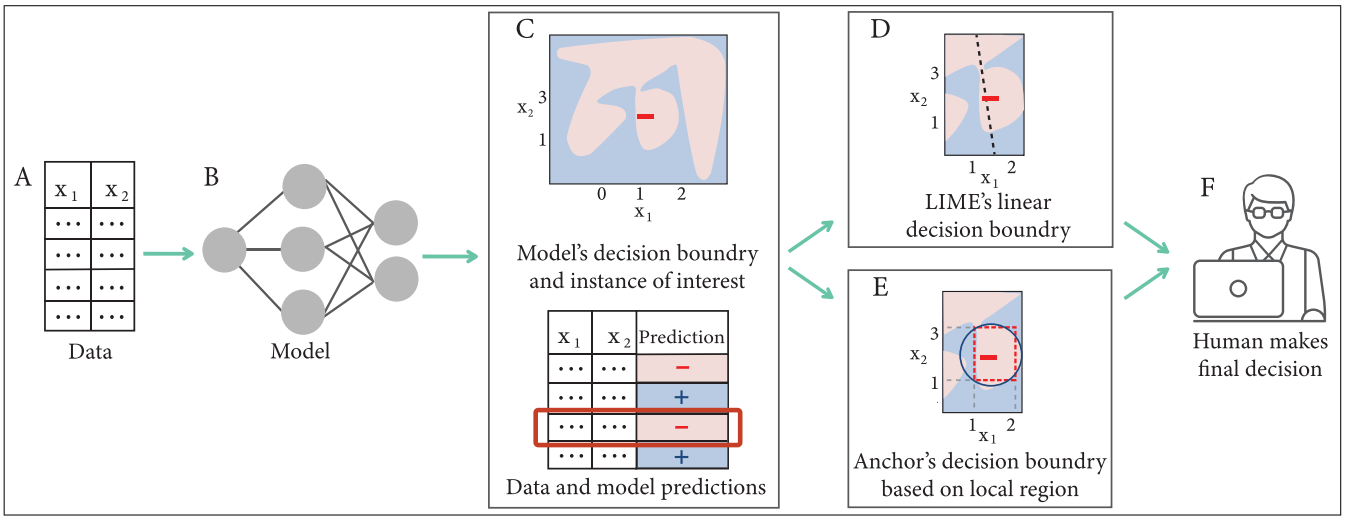

FIGURE 4. An illustration of LIME and Anchors. This illustration describes the intuition behind two alternative techniques, namely LIME and Anchors, on a sample data set with two features. **A.** The training data set where every example has two features, x_1 and x_2. **B.** The data is fed to an ML model such as a neural network. **C.** The ML model produces a decision boundary, dividing the data space into regions where the data is classified as "−" (highlighted in pink) and other regions where the data is classified as "+" (highlighted in blue). The figure also highlights the instance that requires explanation; see the data point marked as "−" in the data space. **D.** The data and the model's predictions are fed to LIME along with the instance that requires explanation. Then, LIME samples instances from the data set and runs a linear model locally, assigning higher weights to the points that are closer to the instance of interest. The outcome of this model is a decision boundary, depicted in the figure as a dashed line, which LIME uses to determine how important each feature is to the classification in this neighbourhood. **E.** The same as **(D)** but for Anchors instead of LIME. In particular, this technique determines the range of feature values that define an "anchor", i.e., a region surrounding the given instance where the classification matches the instance; see the dashed rectangle. The output of this technique is the if-then rule defining this region. For example, in our illustration the output would be the following rule: *If* $(1 \leq x_1 \leq 2)$ *and* $(1 \leq x_2 \leq 3)$*, then classify the data point as "−".* **F.** The analyst is provided with instance-level explanations of how different features influence the model's prediction and makes the final decision accordingly.

to the instance being explained; this way, the resulting coefficients become reflective of the feature ranking.

- Panigutti *et al.* [29] use the permuted data set to train a decision tree for each patient in a health-record data set and then extract a rule-based explanation from the decision tree. This ensures that the resulting explanation is compatible with sequential, multi-labelled, ontology-based data.

In their follow-up papers, the developers of LIME underline the importance and challenges of model-agnostic techniques [55] and propose alternatives to LIME that are capable of explaining model outcome via "if-then" rules [56], [57]. These techniques served as the foundation for the subsequently developed *Anchors* [25]—a technique based on random perturbations, just like LIME, but focuses entirely on the neighbourhood of the instance being explained. Specifically, it uses reinforcement learning and graph search to construct an "anchor", i.e., a region of the neighbourhood where the classification matches the instance; this region is defined by a range of values for each feature. These ranges are then interpreted as if-then rules, as illustrated in Figure 4E. The resulting if-then rules explain not only the instance under consideration but also every instance falling in the anchor. The limitations of Anchors include the risk of getting overly specific and the risk of getting potentially-conflicting anchors.

Another model-agnostic technique for outcome explanation is LOCO (Leave One Covariate Out) [32], which scores the instance features by repeatedly running the model, each time leaving one feature out. At the end of this process, the absolute impact of each feature is calculated, and the one with the highest score is taken to be the most important for that instance. Other model-agnostic techniques use Monte Carlo sampling [33], feature importance ranking [34], genetic algorithms [31], and local linear modelling combined with random forests [30].

Next, we explain a completely different design paradigm, which is based on an important solution concept from *Cooperative Game Theory* called the *Shapley value* [58]. Before explaining the XAI techniques that are based on the Shapley value, we will first explain how it is calculated. Typically, a coalitional game consists of a set of *players* and a *characteristic function* that specifies the "value" of every possible *coalition*, i.e., a subset of players. This value is meant to reflect the worth of the coalition or the quality of its outcome. The *grand coalition* is the one consisting of all players, and one of the fundamental research questions in Cooperative Game Theory is how to divide the value of this coalition fairly among the players. The canonical solution concept designed for this purpose is the Shapley value, the building block of which is the *marginal contribution*. Specifically, for any player p_i and any coalition C_j, the marginal contribution of p_i to C_j is the difference in value that p_i makes when joining C_j. The Shapley value builds on the idea of the marginal contribution while taking into consideration all the possible sequences in which the players could have joined the grand coalition. For example, given four players, $\{p_1, p_2, p_3, p_4\}$,

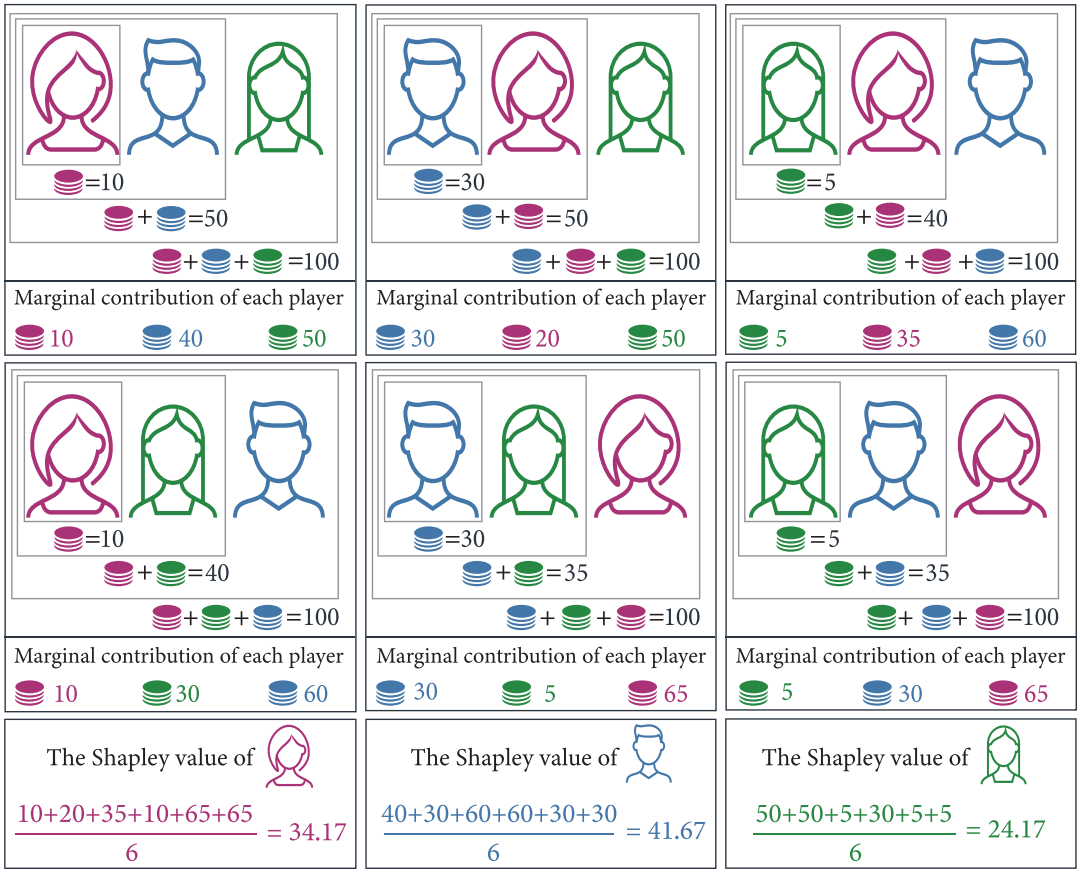

FIGURE 5. An illustration of the Shapley Value. In a coalitional game, every subset of players is called a "coalition", and the game specifies a value for every coalition indicating the quality of its outcome. The Shapley value is a formula introduced by Lloyd Shapley [58] to fairly measure the contribution of each player to the coalition consisting of all players. This is done by considering all possible joining orders of the players. Given the three players in our example, namely Red, Blue and Green, one possible joining order is that Red joined first, followed by Blue, then Green (see the top-left corner). Another possible joining order is that Green joined first, followed by Red, then Blue (top-right corner). For every joining order, the Shapley value computes the "marginal contribution" of each player, which is the difference in value caused when that player joins the coalition. For example, given the joining order in the top-left corner, the marginal contribution of Red, Blue, and Green is 10, 40, and 50, respectively. Then, the Shapley value of each player is defined as its average marginal contribution, taken over all joining orders (see the bottom row of the figure).

one of the possible joining orders is: (p_2, p_4, p_1, p_3), which means that p_2 was the first to join the grand coalition, followed by p_4, then p_1 and finally p_3. Another possible joining order is (p_1, p_3, p_4, p_2), and so on. Now, when computing the Shapley value of, say, p_4, we consider all possible joining orders, and for each such order, we compute the marginal contribution of p_4 to the coalition of players who joined before it. For example, given (p_2, p_4, p_1, p_3), we compute the marginal contribution of p_4 to the coalition $\{p_2\}$, and given (p_1, p_3, p_4, p_2), we compute the marginal contribution of p_4 to the coalition $\{p_1, p_3\}$, and so on. The Shapley value of p_4 is then simply its average marginal contribution, taken over all possible joining orders. A 3-player example is illustrated in Figure 5. The number of computations required to calculate the Shapley value grows exponentially with the number of players involved. Specifically, given n players, there are $n!$ possible joining orders to consider. Although there exists an alternative formula with which the Shapley value can be computed in $O(2^n)$ instead of $O(n!)$ time, the number of calculations remains prohibitive given a large n. Fortunately, the Shapley value can be approximated by sampling from the space of possible joining orders, and a bound can be established on the resulting estimation [59], making it possible to approximate the Shapley value in linear time.

Having explained how the Shapley value is calculated, we are now ready to present the XAI literature that builds on it. Specifically, in this literature, the players in the coalitional game correspond to different features, and the coalition values correspond to the prediction quality that is attained when using different subsets (or "coalitions") of features. Modelling the features as players of a coalitional game enables us to capitalize over decades of research in cooperative game theory, allowing us to take advantage of its well-developed theoretical foundation and its rich repository of solution concepts. One such solution concept that is particularly attractive to use as a measure of feature quality is the Shapley value, as it is considered to be the gold standard for quantifying the contribution of a player, taking into consideration all the

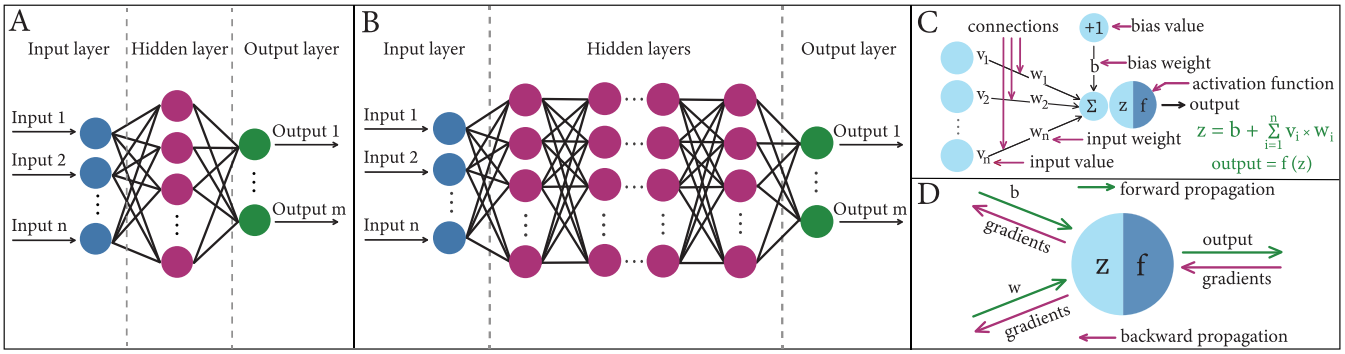

FIGURE 6. An illustration of neural networks and backward propagation. The nodes represent neurons, which belong to either the input layer, the output layer, or any of the hidden layers in between. A. An illustration of Artificial Neural Network (ANN). B. An illustration of Deep Neural Network (DNN), where the number of hidden layers is much larger than in a typical ANN. C. An illustration of the inner workings of neurons in hidden layers. D. Forward propagation refers to the typical flow within an ANN or DNN, which starts from the input layer, goes through the hidden layers, and ends at the output layer. In backward propagation, for each output node, the derivative of the prediction error is calculated with respect to the weights of the edges pointing to that node. The results are then used to perform similar calculations but for the layer before the last, and then for the layer before it, etc., all the way back to the first layer. Finally, the gradient values are subtracted from the weights to minimize the error.

contributions that other players make when interacting with that player. Furthermore, the Shapley value can be approximated in linear time, unlike the vast majority of solution concepts in cooperative game theory, whose calculation is often intractable.

To the best of our knowledge, the approach of using the Shapley value to measure feature contributions was first proposed by Lipovetsky and Conklin [60]. In particular, the authors use the Shapley value to evaluate the coefficients of a linear regression model as well as the relative usefulness of different features in the presence of multicollinearity. The Shapley value was subsequently used to provide insights into the inner workings of artificial neural networks [61] and for feature selection in regression and classification models [62].

Strumbelj and Kononenko [63] were the first to use the Shapley value for outcome explanation. Specifically, the authors compute the Shapley value to explain the outcome of classification models under the assumption of input feature independence. In their follow-up paper [35], instead of computing the Shapley, the authors use sampling-based approximation to speed up the computation and allow for a greater number of features to be considered in the analysis. The authors further extended their work by proposing a sensitivity analysis-based approach to account for potential feature interactions [36] and proposing an algorithm that minimizes the approximation error for a given number of samples.

Another, more recent framework that builds on the Shapley value was proposed by Lundberg and Lee [37]. Their framework, called SHAP—which stands for Shapley Additive ExPlanations—is meant to unify existing methods that fall at the intersection of cooperative game theory, machine learning, and XAI. Specifically, the authors proposed two different versions of SHAP:

- *Kernel SHAP*: A model-agnostic framework for outcome explanation, which combines LIME [24] with the Shapley value. Recall that LIME creates a new data set from perturbed data points around the instance of interest and then trains a local surrogate model on this data set whereby each sampled instance is weighted by its proximity to the instance of interest. Kernel SHAP is similar to LIME in the sense it also trains a local surrogate model, but the difference is that the model is restricted to weighted linear regressions, and the training is done on a different data set. More specifically, Kernel SHAP considers the instance of interest (i.e., the instance that needs to be explained) as a coalitional game in which the players are the features of that instance. Then, to quantify each feature's contribution to the model prediction for this instance, Kernel SHAP samples coalitions from this game and runs a surrogate model for each coalition. This procedure is shown to approximate the Shapley value of each feature of the instance of interest, and these values constitute the desired explanation since they reflect each feature's contribution to the outcome of the instance of interest. One of the limitations of Kernel SHAP is that it ignores features dependencies. Another limitation is that it is computationally expensive. Thus, although it can explain a single instance in a reasonable time, it cannot do so for a large number of instances. This, in turn, makes Kernel SHAP not suitable for explaining models globally [22] since such explanations are typically derived from explaining a large number of representative instances.
- *Deep SHAP*: A model-specific framework, designed to explain the outcome of deep learning models by combining the Shapley value with DeepLIFT [38], [64]. The latter is a model-specific technique that provides outcome explanations for neural networks. These explanations take the form of scores reflecting the importance of input features. The scores are generated using backpropagation—a widely used algorithm in machine learning; see Fig. 6 for an illustration. More specifically,

the scores are computed by first attributing values to the input features, representing the effect of setting the input to a reference value as opposed to its original value; this reference value is chosen empirically by the user based on domain-specific knowledge. The scores are then generated by backpropagating through the neural network. Deep SHAP uses the same idea as DeepLIFT, but instead of assigning a single reference value to each input feature, it borrows ideas from cooperative game theory by considering each input feature to be a player in a coalitional game and approximating the reference values using the Shapley value.

Extending their work, Lundberg *et al.* [39], [65] proposed SHAP interaction values that directly measure local feature interaction effects. They also proposed Tree SHAP, which is a variation of SHAP for tree-based machine learning models such as random forests, decision trees, and gradient boosted trees. More specifically, Tree SHAP works by representing the features as players in a coalitional game and assigning the coalition values based on conditional expectations while taking into consideration the tree structure. The Shapley value is then approximated to determine the importance of features to the instance of interest. Tree SHAP is faster than Kernel SHAP and provides insights not only to the model outcome but also to the model as a whole; see Section IV-A for more details.

Other model-specific outcome explanation techniques include the one proposed by Mollas *et al.* [42], which uses unsupervised learning techniques and a similarity metric to explain individual outcomes of random forests. Another technique was proposed by Haufe *et al.* [41], which transforms non-linear models in terms of multivariate classifiers into interpretable linear models. Finally, we mention the technique proposed by Saabas [40], which explains the predictions of decision trees by following the decision path and attributing changes in the model's expected outcome to each feature along this path. However, this is done using conditional expectations that take into consideration only a single order of the features, instead of taking an average over all possible joining orders, as is the case with Tree SHAP.

B. EXPLAINING HOW THE OUTCOME CAN BE CHANGED

Another type of outcome explanation techniques are example-based explanations. In this context, the most widely studied topic is counterfactual explanation, which is a technique that works by selecting particular instances from the data set to explain the behavior of the machine learning model or the underlying data distribution. The difference between this and other outcome explanation techniques is that instead of providing a summary of the features that were important in the machine learning decision making, it provides a ''counterfactual explanation'' which identifies the features that need to be changed in order to obtain a certain outcome. Counterfactual explanations are usually obtained by formulating and solving an optimization problem, the objective of which is to minimize a loss function. The difference between the existing techniques for obtaining counterfactuals lies in the optimization method and the loss function being used [21]. Similar to other types of XAI techniques, these types of explanations can be model-agnostic and model-specific.

Counterfactual explanations have been used in a wide range of disciplines, including social science [14], data mining [72], philosophy [73] and psychology [74]. A recent review article by Verma *et al.* [75] summarizes a number of recent techniques of counterfactual explanations in machine learning, but this article does not focus on tabular data. As such, in this section, we provide the reader with an overview of counterfactual explanations that are compatible with tabular data. In particular, Section II-B1 presents the techniques that are model specific, while Section II-B2 presents those that are model agnostic.

1) MODEL-SPECIFIC COUNTERFACTUAL EXPLANATIONS

We start with the work of Wachter *et al.* [66], who proposed ''unconditional counterfactual explanations'', a technique for differentiable models, i.e., models for which the gradients of the loss function can be calculated, such as neural networks. Generating the counterfactual explanations is formulated as an optimization problem. The objective of this problem is to minimize the distance between the counterfactual and the original data point, subject to the constraint that the classifier associates the counterfactual with the desired label. The disadvantage of this approach is that it does not handle categorical features well when the number of categories is large. The authors suggest running the method separately for each combination of feature values of the categorical features, although the number of combinations to consider grows exponentially, making it impractical for a large number of categories.

Extending this work, Mothilal *et al.* [67] constructed an optimization problem that considers a diverse set of counterfactual examples, rather than a single one. This way, the user can choose the counterfactual that is more feasible to follow in order to achieve the desired outcome. This method is called DiCE (Diverse Counterfactual Explanations). Solving the optimization problem requires considering the trade-off between diversity and proximity, and the trade-off between continuous and categorical features. The authors use gradient descent to solve the optimization problem. Furthermore, they provide quantitative evaluation metrics for evaluating any set of counterfactual examples.

Ustun *et al.* [68] suggest using *recourse* in order to explain the outcomes of the black-box model. Here, recourse is defined as the ability of a person to obtain the desired outcome from a fixed model by altering *actionable* input features, i.e., features that can be controlled by the person. Examples of actionable features include those corresponding to the income or the number of loans. In contrast, features that correspond to age or ethnicity are not. The authors argue that only actionable features should be considered when extracting counterfactuals. The authors then formulate a

TABLE 2. A summary of the model-specific counterfactual explanations covered in Section II-B1.

Name	Ref.	Year	Type	Underlying mechanism
Unconditional counterfactuals	[66]	2017	Differentiable models	Distance minimization between the counterfactual and the instance under consideration.
DiCE	[67]	2020	Differentiable models	Gradient descent.
-	[68]	2019	Differentiable models	Alternation of actionable features.
-	[69]	2019	Differentiable models	Gradient-based algorithm.
-	[70]	2019	Differentiable models	Class prototypes and autoencoder.
-	[71]	2019	Differentiable models	Mixed integer programming.

discrete optimization problem, which searches over the possible changes that a specific person can make in order to obtain the desired classification outcome. This problem is solved by expressing it as an integer program.

Another recourse algorithm was proposed by Joshi *et al.* [69]. The authors model the underlying data distribution or manifold, i.e., topological space that locally resembles Euclidean space near each point. They provide a gradient-based algorithm that allows sampling from the latent space of the model to find the smallest set of changes that would improve the outcome. The proposed algorithm provides recourse for classification and causal decision-making systems. To handle immutable variables, the authors propose and train conditional variants of causal decision-making systems. The proposed recourse algorithm can highlight whether a decision-making algorithm is systematically confounding specific attributes. Similar to the previous two, this technique is also limited to differentiable models.

Looveren and Klaise [70] proposed a technique for finding counterfactual explanations of differentiable classifier predictions using *class prototypes*. Specifically, a class prototype is the average encoding over the K nearest instances in the latent space with the same class label. The latent space is obtained by using an autoencoder, which is a type of artificial neural network designed to learn a representation for a set of data, typically for dimensionality reduction, by training the network to ignore the noise in an unsupervised manner. The class prototypes are used in the objective function to guide the perturbations quickly towards an interpretable counterfactual. The authors propose alternative metrics for quantifying the interpretability at the instance level. They also propose pairwise distance measures to convert the embeddings of categorical variables into a numeric space, which allows them to define meaningful perturbations when generating counterfactuals.

Russell [71] proposed a search algorithm and a set of constraints for counterfactuals based on mixed-integer programming. The primary goal of this technique is to explain financial decisions, assuming that the classifier is linear (e.g. SVM, or linear/logistic regression) and that the data has been transformed via mix-encoding or dummy variable encoding. The author formulated a cost minimization optimization problem and a set of constraints restricting the state of features altered in previously generated counterfactuals, to make sure that diverse counterfactuals are generated.

2) MODEL-AGNOSTIC COUNTERFACTUAL EXPLANATIONS

Guidotti *et al.* [31] proposed a technique called LORE, which stands for Local Rule-Based Explanations of black-box decision systems. It uses a genetic algorithm to generate a synthetic neighborhood of the given instance. Based on this neighborhood, an explanation is derived, which consists of a decision rule and a set of counterfactual rules, suggesting the changes in the instance features that lead to a different outcome. Due to the nature of these explanations, the user is not only given a specific example of how to obtain actionable recourse for the instance at hand but is also given an abstract characterization of its neighboring instances that have the opposite label. This way, the user can compare the instance to its neighbor, and can identify the differences between the two, which led to the instances being classified differently.

Grath *et al.* [76] proposed a technique called positive counterfactuals. This technique adapts counterfactuals to credit applications in order to explain the positive outcomes of the classifiers, rather than focusing only on negative outcomes. In the case of positive outcomes, the authors interpret counterfactuals as a *safety margin*, i.e., tolerance from the decision boundary of the classifier. Thus, instead of answering the question of why was the loan denied, they measure the distance between the application and the decision boundary, to quantify how close the application was from being denied. The counterfactuals are generated by optimizing the loss function proposed by Wachter *et al.* [66]. Instead of considering all features, the authors focus only on the important ones. By reducing the number of features to be considered, the authors end up reducing the number of changes that the user can choose from in order to obtain the desirable outcome, which makes the decision easier to make for the user. The way in which the authors assess the importance of each feature is based on two weighting strategies. The first relies on the global feature importance using analysis of variance (ANOVA F-values). The second strategy follows a K-Nearest Neighbors approach to identify cases that are similar to the instance except that their classification yielded the desired outcome. Finally, the authors rely on the size

TABLE 3. A summary of model-agnostic counterfactual explanations covered in Section II-B2.

Name	Ref.	Year	Underlying mechanism
LORE	[31]	2018	Genetic algorithm.
Positive counterfactuals	[76]	2018	K-Nearest Neighbours and ANOVA F-values.
MACE	[77]	2020	Logic formulae to express the predictive model and the distance function.
CLER	[78]	2019	Local regression models: combination of LIME and unconditional counterfactual explanations.
Growing spheres	[79]	2019	Decompositional algorithm for uniform sampling.
CERTIFATI	[80]	2019	Genetic algorithm.
C-CHVAE	[47]	2020	Variational autoencoder (VAE).
FACE	[81]	2020	Shortest path algorithm.
MOC	[82]	2020	ICE curves and genetic algorithm.

of explanations as a proxy to measure their interpretability, favoring the explanations that involve the smallest number of features that the user should change in order to obtain the desired outcome.

Karimi *et al.* [77] proposed a technique called MACE, which stands for Model-Agnostic Counterfactual Explanations. This technique generates the nearest counterfactual explanations under any given distance function while supporting additional plausibility constraints. The authors map the nearest counterfactual problem into a sequence of satisfiability problems, by expressing both the predictive model and the distance function (as well as the plausibility and diversity constraints) as logic formulae.

White and Garces [78] proposed CLER: Counterfactual Local Explanations via Regression. The proposed technique explains both the outcome of the model and how it would change if things would have been different. CLER generates counterfactual explanations that identify the minimum changes necessary to flip a prediction's classification. It then builds local regression models, using the counterfactuals to measure and improve the fidelity of its regressions. In contrast, the LIME method [24] which also uses regression to generate local explanations, neither measures its own fidelity nor generates counterfactuals. CLER's regressions are found to have on average 40 percent higher fidelity than LIME. CLER provides counterfactual explanations by building on the strengths of two state-of-the-art explanatory methods, while at the same time addressing their weaknesses. The authors mention that the work of Wachter *et al.* [66] misses the part of feature interaction and the equation scope does not cover the neighborhood around the instance. CLER uses step-wise regression on the neighborhood of the instance of interest to generate counterfactuals and measure the fidelity of the regression coefficients.

Laugel *et al.* [79] proposed Growing Spheres, an instance-based approach the idea of which is to explain a single prediction of a model through comparison. The proposed technique locally explores the input space of the classifier to identify its decision boundary. Given a data point whose classification needs to be explained, the proposed method consists of identifying a close neighbor classified in a different class to determine the minimal change needed in the instance at hand to change the classifier's prediction.

Sharma *et al.* [80] proposed a unified and model-agnostic approach called Counterfactual Explanations for Robustness, Transparency, Interpretability, and Fairness of Artificial Intelligence models (CERTIFAI). Given a model and input instance, CERTIFAI uses a custom genetic algorithm to generate counterfactuals. In addition to that, the authors introduce Counterfactual Explanation-based Robustness Scores (CERScore) which is a black-box model robustness score function that enables the comparison between different models trained on different data sets. Given a model, the CERScore is defined as the expected distance between the input instances and their corresponding counterfactuals. A higher CERScore implies that the model is more robust.

Pawelczyk *et al.* [47] proposed a framework called C-CHVAE which stands for Counterfactual Conditional Heterogeneous Autoencoder. It works for tabular data without the specification of distance or cost functions in the input space. The authors suggest embedding counterfactual search into a variational autoencoder (VAE), which is a data density approximator. The idea of using VAE as a search method is to find counterfactuals that are proximate and connected to the input data. Given the original data, the encoder produces a more concise representation of the data, allowing the framework to search for potential counterfactuals in the lower dimensional neighborhood. Next, the authors perturb the low dimensional data representation and feed the perturbed representation into the decoder. For small perturbations, the decoder gives a potential counterfactual by reconstructing the input data from the perturbed representation. Next, the potential counterfactual is passed to the pre-trained classifier, to determine whether the prediction has been altered. The authors also suggest adding a quality measure to the generated counterfactual explanations, to quantify

TABLE 4. A summary of the XAI techniques covered in Section III, all of which are designed for model inspection.

Name	Ref.	Year	Type	Underlying mechanism
-	[83]	2010	Agnostic	Sensitivity analysis and Gaussian process classification.
VEC	[84], [85]	2011	Agnostic	Sensitivity analysis and visualisation techniques.
QII	[86]	2016	Agnostic	Sensitivity analysis and coalitional game theory.
VIN	[87]	2004	Agnostic	Partial dependence plots & functional ANOVA decomposition.
ICE	[88]	2015	Agnostic	Partial dependence plots and estimated functional relationship.
Prospector	[89]	2016	Agnostic	Partial dependence plots and random perturbations
-	[90]	2018	Agnostic	Partial dependence plots, ICE, and SHAP.
-	[91]	2018	Agnostic	Partial dependence plots and feature influence quantification.
-	[92]	2016	Agnostic	Partial dependence plots and Orthogonal projections of input attributes.
-	[93]	2017	Specific	Deep Neutral Network visualisation on information plane.
TreeView	[94]	2016	Specific	Surrogate decision tree approximation of Deep Neutral Networks.
Integrated Gradients	[95]	2017	Specific	Attributing the prediction to the input features of Deep Neutral Networks.
-	[96]	2002	Specific	Sensitivity analysis, and assessment of both the input features and the connections of Artificial Neutral Networks.
-	[97]	2014	Specific	Feature contributions for random forest models.
-	[98]	2012	Specific	Feature importance generated by random forest models and partial dependence plots.
Forest Floor	[99]	2016	Specific	Visualising random forest models based on feature contributions.

the degree of difficulty. This measure is based on the similarity between the input feature and the potential counterfactual explanation.

Poyiadzi *et al.* [81] proposed a technique called Feasible and Actionable Counterfactual Explanations (FACE), which can obtain counterfactuals by quantifying the trade-off between the path length and the density along that path. The density along the path can be minimized by the shortest path algorithm. The proposed approach respects the underlying data distribution and the resulting counterfactual are connected via high-density paths to the explained instance.

Dandl *et al.* [82] proposed a Multi-Objective Counterfactuals (MOC) method which translates the counterfactual search into a multi-objective optimization problem. The proposed approach returns a diverse set of counterfactuals with different trade-offs between the proposed objectives and maintains diversity in the feature space. This enables a more detailed analysis giving more options for actionable user responses to change the predicted outcome. Compared to the technique proposed by Wachter *et al.* [66], this technique uses a distance metric for mixed feature spaces and two additional objectives: the first measures the number of feature changes to obtain sparse and more interpretable counterfactuals; the second measures the closeness to the nearest observed data points for more plausible counterfactuals. The authors propose to measure the feature importance for a single prediction with the Individual Conditional Expectation (ICE) curves [88]—curves showing how the prediction changes when the feature is changed, while other features are fixed to the values of the considered observation. They use the Nondominated Sorting Genetic Algorithm [100] with some modifications to solve the multi-level optimization problem.

III. XAI FOR MODEL INSPECTION

Model inspection techniques provide insights for understanding either how the black-box model works or why it returns predictions with a higher probability for some instances than for others. In other words, model inspection techniques are designed to understand some specific properties of the black-box model or its predictions. The goal is to understand how internally the black-box behaves when the input is changed. This is done by estimating the relevance of a feature by changing the input or some internal components and recording how the model is affected by such changes.

As mentioned earlier in the introduction, model inspection techniques are divided into two categories: model-agnostic and model-specific. As for the former one, it can be further categorized into those that are based on sensitivity analysis (which will be explained in Section III-A) and those that are based on partial dependence plots (which will be explained in Section III-B). As for the model-specific inspection techniques, they will be discussed in Section III-C. A summary of the techniques considered in this section can be found in Table 4.

A. MODEL INSPECTION VIA SENSITIVITY ANALYSIS

Sensitivity analysis consists of methods capable of quantifying how the uncertainty of an ML model is related to

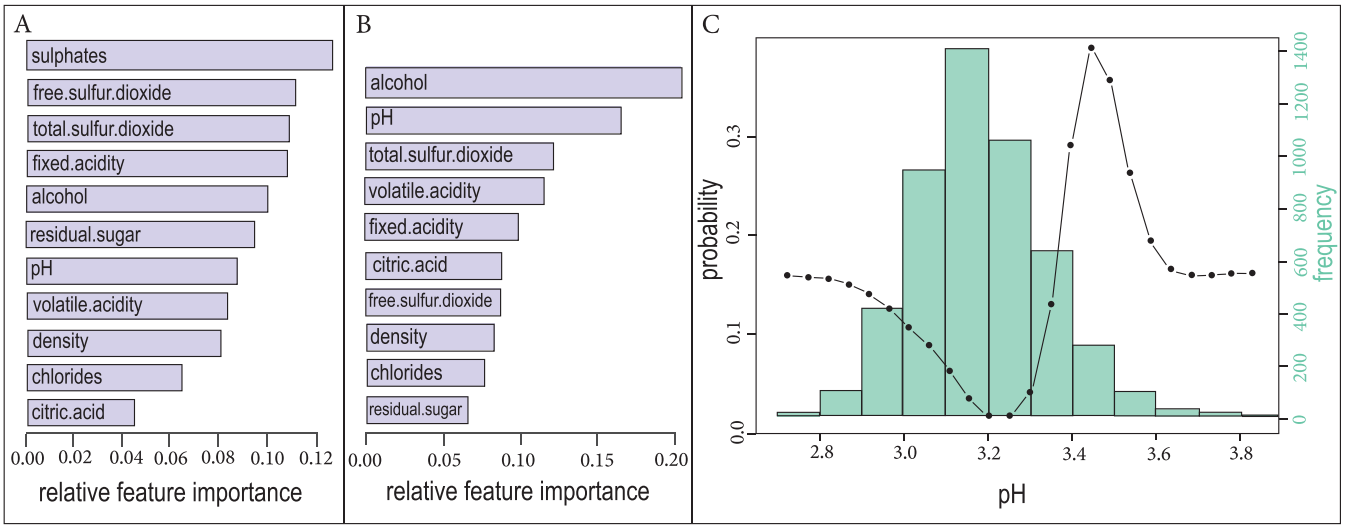

FIGURE 7. An illustration of feature importance and the VEC curve. This example uses a data set on white wine quality [101], consisting of 11 features and an output variable representing the quality of wine on a scale from 3 to 9. A. Bar plot representing feature importance based on the regression analysis of the white wine quality data set, where the goal is to predict human expert taste preferences. B. Bar plot representing the two-dimensional interactions between "sulphates" (since it has the highest importance in Fig. 7A) and each of the remaining features. This reveals that, when predicting wine quality, alcohol and pH have higher interactions with sulphate compared to the other features. C. Analysis of the white wine quality data set to classify wine as either *low quality* (where the output variable is <6) or *high quality* (where the output variable is ≥ 6). In addition to the histogram of pH (which is the most indicative feature of wine quality according to the paper), the figure shows the Variable Effect Characteristic (VEC) curve, which represents the influence of pH on the model's classification. A non-linear influence of the pH can be observed, highlighting the fact that pH values within the range [3.4, 3.5] maximize the probability of wine quality being classified as high.

the uncertainty in the input features. Such analysis can be used to explore the robustness and accuracy of the model results under certain conditions and may reveal unexpected relationships between the model parameters and the input features. It measures the effect of input changes on the model's behaviour and helps to enhance the model's stability, especially since it identifies the inputs that cause significant uncertainty in the model performance [102]. Sensitivity analysis is widely used in the XAI literature as a model-agnostic explanation technique. Next, we explain some of these techniques.

Baehrens *et al.* [83] proposed a technique based on Gaussian process classification, which is a non-parametric classification method. The proposed technique provides local explanation vectors to explain the prediction results at the instance level. These explanations consist of local gradients characterizing how a data point has to be moved to change its predicted label. These local gradients also highlight the most influential features in the decision of the model for a particular instance.

Datta *et al.* [86] proposed a technique to estimate the influence of specific input features on the black-box model results, taking into consideration any correlated features. Their approach is based on the observation that, in order to estimate the influence of different features, it is insufficient to analyse them in isolation since there might be synergies between the different features. To take any such synergies into consideration, the authors borrow solution concepts from cooperative game theory, including the Shapley value and the Banzhaf index.

Cortez and Embrechts [84], [103] proposed yet another approach, which uses well-established sensitivity analysis techniques combined with visualisation that are meant to provide more intuitive explanations. Here, the sensitivity analysis considers the range, gradient, and variance of the predictions. The outcome of this analysis is then visualised using bar plots as well as a curve which they call the ''Variable Effect Characteristic (VEC) curve'', which is used to demonstrate the average impact of a given feature in the model; see Fig. 7 for an illustrated example. The authors also propose three novel sensitivity analysis techniques [85], namely data-based sensitivity analysis, Monte-Carlo sensitivity analysis, and cluster-based sensitivity analysis, as well as one novel input importance measure, namely average absolute deviation; see the paper for more details.

B. MODEL INSPECTION VIA PARTIAL DEPENDENCE PLOTS

Partial Dependence Plot (PDP), initially proposed by Friedman [19], is a tool for visualising the average marginal effect of one or two non-correlated features on the predicted outcome of an ML model. It can help to observe whether the relationship between the outcome variable and a feature is linear, monotonic, or more complex. PDPs are intuitive and easy to implement, but they work under the assumption that the feature being analysed is not correlated with other features, which is an assumption that may not always hold. Next, we explain some of the PDP techniques.

Hooker proposed a technique [87] for evaluating the significance of non-additive interactions between any set of features. The implementation of this technique uses a

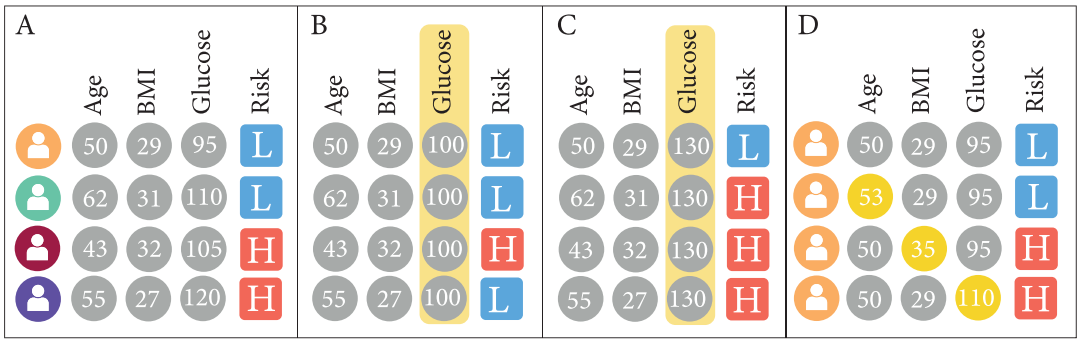

FIGURE 8. An illustration describing how Partial Dependence Plots (PDPs) are computed. This example is based on a data set predicting the risk of developing diabetes, where each row represents a patient, and each column represents a feature. The "Risk" column is the output of the predictive model (H means High risk, while L means How risk). A. The original feature values of each patient. B. If the analyst wants to examine the impact of glucose on the prediction, partial dependence can be applied by varying the value of glucose while keeping all other features intact to determine how glucose affects the prediction. In this example, the glucose value is set to 100 for all patients, which changes the prediction of one patient from high risk to low risk. C. The glucose value is now set to 130 for all patients, which changes the prediction of one patient from low risk to high risk. The findings in Fig. 8B and 8C suggest that there is partial dependence between glucose and the prediction. D. Prospector—one of many PDP-based techniques—can also be used for local inspection. In this illustration, all rows represent the same patient. The first row corresponds to the original feature values of that patient, while each of the remaining rows corresponds to a change in one of those values, which is highlighted in yellow. As can be seen, changing the age does not change the prediction, while changing BMI or glucose changes the prediction from low risk to high risk.

variable interaction network visualisation generated using functional ANOVA decomposition. This way, the user can visualise the importance of the features along with their interdependencies.

Krause *et al.* [89], [104] proposed a technique called Prospector, which is an interactive visual analytics system. It introduces random perturbation of the input feature values of the black-box model to understand the extent to which every feature affects the prediction. The idea of Prospector is to observe how the output varies by changing one input feature at a time; see Fig. 8 for an illustrated example. Finally, we note that this technique can also give insights into the most important features.

Goldstein *et al.* [88] proposed Individual Conditional Expectation (ICE) plots, which help to visualise the average partial relationship between the outcome and certain input features. ICE plots' contribution to PDP plots is its ability to highlight the variations in the fitted values.

A recent technique combining PDPs, ICE, and SHAP was proposed by Casalicchio *et al.* [90]. This technique visualises the expected conditional feature importance instead of the expected conditional prediction. This way, the user can determine the extent to which each feature contributes to the prediction of the model, either on a local or a global level.

Adler *et al.* [91] proposed a technique that focuses on the indirect influence of some features via other related features on the model outcome. This technique builds on the work of Feldman *et al.* [105], who certify and remove bias in classifiers, based on the observation that the information content of a feature can be estimated by predicting it from the remaining features. The novelty in Adler *et al.*'s work is that, instead of *eliminating* a feature's influence to remove the associated bias, the authors *quantify* the feature's influence.

Adebayo and Kagal [92] proposed a process that iteratively transforms the input features in a given data set to obtain several new data sets, where one feature is removed after each iteration. The authors then compare the different data sets to determine the impact that each feature has on the black-box model. The way in which a feature, f_i, is "removed" at any given iteration is as follows. The authors use a particular type of linear transformation called orthogonal projection [106], whereby every feature other than f_i is made orthogonal to f_i, thereby removing any linear dependence between these features. The technique proposed by Adebayo and Kagal is part of a larger project called FairML [107], the idea of which is to measure the model's dependence on its input features by changing them. This way, if a small change to an input feature significantly changes the model output, it means that the model is sensitive to that particular feature.

C. MODEL INSPECTION VIA MODEL-SPECIFIC TECHNIQUES

The model inspection techniques that were presented thus far in this section are all model agnostic. In this subsection, we present the techniques that are model-specific.

Olden and Jackson [96] used sensitivity analysis to interpret the internal mechanisms of Artificial Neural Networks (ANN). By assessing the importance of connections within the network, and the contributions of the input features to the output, the technique allows to remove null neural pathways and insignificant input features from the network, thereby reducing its complexity and improving its interpretability. Tishby and Zaslavsky [108] noted that neural networks with certain properties could be studied in the *information plane*—the plane of mutual information values that each layer preserves on the input and output variables. This is done to extract relevant information that a given feature may contain

about the output. Schwartz-Ziv and Tishby [93] extended this idea, which provides a better understanding of the training dynamics, learning processes, and internal representations of deep learning.

Thiagarajan *et al.* [94] proposed a technique called TreeView, the goal of which is to provide a visual representation of DNNs using a decision tree. This is achieved by applying hierarchical partitioning to the feature space, thereby revealing the iterative rejection of unlikely class labels and the identification of the most likely label.

Sundararajan *et al.* [95] studied the problem of attributing the prediction of a DNN to its input features and proposed an attribution method called *integrated gradients*, which is a variation on computing the gradient of the prediction output with regard to input features.

Finally, we present a number of techniques that are built around random forests. In particular, Auret and Aldrich [98] use variable importance measures and partial dependence plots generated by random forest models as an extension to regression models to analyse feature interactions and identify root causes of abnormal processes. Welling *et al.* [99] proposed *Forest Floor*, a tool that allows the user to visualise random forest models and observe feature interactions. To do so, the authors first use feature contributions to decompose decision trees, then use projection mappings from the feature space to the prediction space. Another technique designed to interpret random forests is proposed by Palczewska *et al.* [97], which extends the generic idea of feature ranking by proposing median analysis, cluster analysis, and log-likelihood. This way, it becomes possible to identify patterns of interest in the input features and determine the influence of each feature on the model prediction for an individual instance.

IV. XAI FOR MODEL EXPLANATION

In this section we will discuss the XAI techniques that are capable of explaining the logic of the black-box ML model. Such explanations are typically derived by building another, interpretable model that mimics the black-box model. Section IV-A discusses the techniques that are model-agnostic, while Section IV-B covers those that are model-specific.

A. MODEL EXPLANATION VIA MODEL-AGNOSTIC TECHNIQUES

As mentioned in Section II, some techniques that were designed for outcome explanation can also provide global explanation of the underlying model. In this sub-section, these techniques will be discussed first, followed by other model agnostic techniques for model explanation. A summary of all techniques described in this sub-section is presented in Table 5.

Starting with SP-LIME [24], it takes a representative sample set from the input data and computes the LIME coefficients for all instances in that set, thereby providing a global view of the ML model's decision boundary.

Another technique that is also based on LIME is called ILIME [27]. We already discussed this earlier in Section II as a technique for outcome explanation, but here we discuss how it can also be used for model explanation. This can be done by grouping similar instances into clusters using a dendrogram and then providing an outcome explanation of a representative instance taken from each cluster. Tree SHAP [39] is another outcome explanation technique from Section II that can also be used for model explanation. This is done using a set of tools that provide a global understanding of the model by computing local explanations across all samples. These tools include (i) a global feature importance plot, (ii) a local explanation summary plot, (iii) feature interaction plots, which combine feature effects after subtracting the main effect of individual features, and (iv) local explanation embeddings, which support both supervised clustering as well as interpretable dimensionality reduction.

Some model explanation techniques are based on General Additive Models (GAMs) [116]. Generally speaking, GAMs are models that capture non-linearities in the data since the dependent variable is related to the independent ones by functions that are not necessarily linear. The model explanation techniques that are based on GAMs specify the importance of individual features along with the shape of the function that captures both the linearities and non-linearities. One such technique was proposed by Lou *et al.* [109], which can interpret regressions, single trees, and tree ensembles by modelling the dependent variable as a sum of univariate models. In the refined technique proposed by the same authors [110], called Generalized Additive Models plus Interactions ($GA^2M - models$), they suggest making the explanations easier to understand by adding only selected pairs of interacting features to the model. To this end, they propose a method for ranking the pairs of features in order to identify the ones that are most beneficial to the model. Caruana *et al.* [117] demonstrated potential applications of GA^2M in a healthcare-related case study.

Another technique of model explanation is called GoldenEye [111], which is based on data randomization to identify groups of features whose interactions have an impact on the prediction. The feature groups and the dependencies therein represent the global explanation. A year later, Henelius *et al.* [118] proposed a refined version of GoldenEye called GoldenEye++, which utilises a more sensitive grouping metric.

Krishnan and Wu [112] proposed a technique called PALM, which stands for Partition Aware Local Model. It assists the debugging of machine learning algorithms by summarizing the training data set. To do so, it approximates the black-box model by partitioning the training data set using a surrogate model and a set of sub-models, which in turn, approximate the patterns within each partition. The sub-models can be complex to capture the sophisticated local patterns. However, the surrogate model is designed to be a decision tree so that the user can easily identify the branch of the tree in which the misclassification is occurring.

TABLE 5. A summary of model-agnostic XAI techniques covered in Section IV-A, which are all designed for model explanation.

Name	Ref.	Year	Underlying mechanism
SP-LIME	[24]	2016	Computing LIME coefficients for a representative sample set of data.
ILIME	[27]	2019	Aggregation of local explanations by hierarchical clustering.
Tree SHAP	[39]	2020	Generation of summary plots and feature interaction plots based on the computed SHAP coefficients across all samples.
GAMs	[109]	2012	Modelling the dependent variable as a sum of univariate models.
GA^2M	[110]	2013	GAMs with addition of selected interacting pairs of features.
GoldenEye	[111]	2014	Modelling data randomization and feature interactions.
PALM	[112]	2017	Partitioning the training data and approximating the patterns in each partition by using sub-models; a decision tree is then trained using these sub-models to approximate the ML model.
FIRM	[113]	2009	Assessing the importance of a feature by estimating its total impact on the score of a trained predictor, taking into consideration potential correlations between features.
MFI	[34]	2016	FIRM with the added ability to assess the significance of features.
-	[114]	2017	Approximation of the ML model using a decision tree.
MUSE	[115]	2019	Learning a small number of compact decision sets capturing the behaviour of a given black-box model.

Zien *et al.* [113] proposed yet another model explanation technique called Feature Importance Ranking Measure (FIRM), which is a generalisation of a classification technique called Positional Oligomer Importance Matrices (POIMs). In more detail, POIMs considers the correlations between different features in order to measure the impact of binary features on the performance of SVM-based sequence classifiers that are used for string classification [119].

FIRM is a generalisation of POIMs that can be used on real-valued features (rather than just binary ones) and on a broader range of classifiers (rather than just SVM for string classification). As FIRM takes into account any correlations between the features, it is able to identify the most relevant ones even when the training data contains considerable noise. Vidovic *et al.* [34] proposed a technique that builds on FIRM, called Measure of Feature Importance (MFI). This technique can be applied to any classifier and can provide outcome explanations as well model explanations. To do so, the authors provide formulas for assessing which features matter the most to the machine learning algorithm being explained. It is worth mentioning that MFI is intrinsically non-linear and can detect inconspicuous features that only impact the prediction function through their interaction with other features.

Another recent technique for model explanation is proposed by Bastani *et al.* [114]. In particular, the authors propose to construct global explanations of black-box models in the form of a decision tree that approximates the original model. The proposed technique actively samples from the training examples to avoid overfitting while extracting the decision tree.

Last but not least, we mention the work of Lakkaraju *et al.*, who proposed a technique called Model Understanding through Subspace Explanations (MUSE) [115]. This technique provides a better understanding of a given black-box model by explaining how it behaves in sub-spaces characterized by certain features of user interest. The construction of explanations is guided by an objective function that simultaneously optimizes for fidelity, unambiguity, and interpretability. This, in turn, yields a small number of decision rules that make the model easy to understand. In another study, the same authors proposed a technique called BETA, which stands for Black-box Explanations through Transparent Approximations [120], where they use a two-level Boolean rule predictor to explain the black-box model.

B. MODEL EXPLANATION VIA MODEL-SPECIFIC TECHNIQUES

The techniques presented in this section are each designed to address the model explanation problem for only specific type of models, which is either a neural network, a support vector machine, or a tree ensemble. We categorize these techniques based on the way in which the explanation is generated, which is either using tree approximation (Section IV-B1) or using rule extraction (Section IV-B2).

1) MODEL EXPLANATION VIA TREE APPROXIMATION

This subsection presents various model-specific techniques that provide model explanations based on tree approximation. A summary of these techniques can be found in Table 6.

A set of studies propose to use a decision tree to mimic the global behaviour of the underlying black-box model. Craven and Shavlic [121] were the first to use this approach to explain neural networks. Their technique, called Trepan, queries the neural network to build a decision tree that approximates the concepts represented by the networks by maximizing the gain ratio. Trepan inspired a number of researchers, including

TABLE 6. A summary of the model-specific covered in Section IV-B1, all of which provide model explanation using tree approximation. Here, the type of the model being explained is either ANN (Artificial Neural Networks), DNN (Deep Neural Networks), or tree ensembles.

Name	Ref.	Year	Type	Underlying mechanism
Trepan	[121]	1996	ANN	Query the neural network to extract decision tree approximation and maximize the gain ratio.
-	[122]	1999	ANN	Query the trained neural network by using a genetic algorithm and extract a decision tree approximation.
ANN-DT	[123]	1999	ANN	Binary decision tree extraction from a trained neural network taking into consideration feature importance.
DecText	[124]	2002	ANN	Trepan with the addition of four novel splitting methods.
-	[125]	2009	ANN	Decision tree extraction from neural networks using genetic programming.
-	[126]	2016	DNN	Utilisation of gradient boosted trees to extract interpretable models from neural networks.
-	[127]	2018	DNN	Model complexity penalty, which encourages the deep neural network to reduce its complexity, thereby becoming more interpretable.
-	[128]	1998	Tree Ensembles	Grouping similar trees in a tree ensemble by using dissimilarity measures.
CMM	[129]	1998	Tree Ensembles	Training a tree ensemble model on multiple synthetic data sets to generate a decision tree whereby every leaf corresponds to a single data set.
CAD-MDD	[130]	2013	Tree Ensembles	Generation of a synthetic data set which mimics the distribution of the original data set, which facilitates the explanation process in cases where the original data set is small.
-	[131]	2016	Tree Ensembles	CAD-MDD with the addition of hypothesis testing.
-	[132]	2016	Tree Ensembles	Using trees and distance functions to select data points representing each class; these points constitute the explanation.
-	[133]	2007	Tree Ensembles	Quantitative evaluation of uncertainty using a Markov chain Monte Carlo technique.
-	[134]	2016	Tree Ensembles	Using expectation maximization as a post processing technique to improve model interpretability.

Krishnan *et al.* [122], who proposed a technique that queries the neural network using a genetic algorithm. The idea is to take an output vector of the neural network and use the query to obtain the corresponding input vector; these vectors are then used to extract a decision tree that behaves just like the neural network (i.e., provides a similar output given the same input), but is inherently easier to understand than the neural network. Another technique that is inspired by Trepan is the one proposed by Schmitz *et al.* [123]. This technique extracts binary decision trees from a trained neural network. However, unlike Trepan, where the neural network has to have a discrete output, the technique of Schmitz *et al.* can be applied on neural networks even when the output is continuous. Furthermore, their technique takes feature importance into consideration, unlike Trepan. Boz [124] proposed yet another technique that is based on Trepan, with the added ability to identify the most relevant features during the tree construction. Finally, we mention the work of Johanson and Niklasson [125], who use genetic programming for evolving decision trees to mimic the behaviour of a neural network.

All of the aforementioned techniques provide explanations for neural networks that are not deep. Next, we present two techniques that can be used for deep neural networks. In particular, Che *et al.* [126] proposed a technique that uses gradient boosted trees to learn interpretable models from a deep neural network. Wu *et al.* [127] proposed another technique whereby the deep network is optimized based on a function that penalizes model complexity; this results in deep networks that are less complex and hence easier to interpret. More specifically, the authors show that their optimization technique produces deep time-series models whose decision boundaries can be approximated by small decision trees.

Having discussed how neural networks can be explained, we now discuss how to explain tree ensembles. In particular, tree ensembles are combinations of decision trees that produce superior predictive performance compared to a single decision tree. Some examples of tree ensembles are random forests and boosted trees. Next, we present the techniques proposed in the literature to explain tree ensembles, starting with the work of Chipman *et al.* [128]. The authors note that, although random forests may contain hundreds of trees, many of those trees typically have similar topologies that differ by only a few nodes. Based on this observation, the authors propose distance metrics for tree objects and use those metrics to identify and cluster similar trees. Finally, they pick a representative sample from each cluster, which they call *an archetype*, and provide those archetypes as the model explanation.

TABLE 7. A summary of the model-specific covered in Section IV-B2, all of which provide model explanation using rule extraction. Here, the type of the model being explained is either ANN (Artificial Neural Networks), SVM (Support Vector Machine), or tree ensembles.

Name	Ref.	Year	Type	Underlying mechanism
-	[135]	1994	ANN	Learning rules from the training data and its randomized extension.
-	[136]	1999	ANN	Binary rule extraction from truth tables generated based on trained ANN.
RxREN	[137]	2012	ANN	Rule extraction from feed forward ANN via reverse engineering.
-	[138]	2002	SVM	Group prototype vectors for each class by using clustering.
SVM+ Prototypes	[139]	2002	SVM	Decision function determination by means of SVM and clustering algorithm to find prototype vectors.
-	[140]	2005	SVM	Non-parametric clustering to identify the prototype vectors.
-	[141]	2005	SVM	Constraint programming to convert SVM into a set of non-overlapping and interpretable rules.
SQRex-SVM	[142]	2007	SVM	Rule extraction using a sequential search algorithm.
-	[143]	2005	SVM	Rule extraction by using the model support vectors and a machine learning technique with explanation capability.
-	[144]	2004	SVM	Rule extraction using intervals defined by hyper-rectangular forms.
-	[145]	2005	SVM	Fuzzy rule extraction based on the coordinates projected from support vectors.
MK-SVM	[146]	2007	SVM	Using feature selection, rule extraction and prediction modelling to improve the explanation capability of SVM.
GSVC	[147]	2006	SVM	Interpretation of SVM decisions in terms of input space segmentation.
InTrees	[148]	2019	Tree ensembles	Iterative search for the matching rule given a new observation.

Other techniques have been proposed to explain tree ensembles, which are all based on two steps: (1) generating a large synthetic data set using the prediction of the random forest; (2) training a decision tree on this synthetic data set to mimic the behaviour of random forest. The decision tree is then provided as the explanation since it is inherently easy to understand by humans. The first technique that uses these steps was proposed by Domingos [129]; their technique is called CMM, which stands for Combined Multiple Models. Specifically, given a tree ensemble, the proposed approach first modifies the input data set a number of times and each time learns a set of black-boxes. Then, a decision tree is built using these data sets and black-boxes. Inspired by this approach, Gibbons *et al.* [130] propose a different technique designed to take advantage of both the accuracy of tree ensembles and the interpretability of single trees. To do so, the authors fit a random forest to the input data first. After that, they generate a synthetic data set that is much larger than the original one while preserving the distribution of each feature. A single tree is then fitted to this synthetic data set, with the goal being to mimic the output of the random forest as closely as possible while using enough data to reduce the sensitivity of the tree to small perturbations. Finally, the authors cut the decision tree to reduce its depth to a number between 6 and 11, in order to make the tree more understandable to humans. This technique is especially useful in cases where the original data set is small (e.g., the one used by the authors, which contained only 656 observations) since the synthetic data set is much larger than the original one. As an extension of this work, Zhou and Hooker [131] proposed to use hypothesis testing to identify the best way in which the data is divided at each branch of the decision tree. More specifically, they compute the Gini index to ensure that the division maximizes information gain.

Other studies of model approximation via decision trees include the approximation of (1) random forests and gradient boosted trees by sampling observations from each class [132], (2) Bayesian decision trees ensembles by implementing a Markov Chain Monte Carlo technique [133], and (3) additive tree models by utilising the expectation-maximization algorithm as a post-processing method [134].

2) MODEL EXPLANATION VIA RULE EXTRACTION

This subsection presents a number of model-specific techniques that provide model explanations using rule extraction. A summary of these techniques is provided in Table 7.

There are a number of approaches that have been developed to extract if-then rules from trained neural networks [149]–[152]. These approaches treat the extraction as a search problem, where the search involves finding rules that explain the activation of output and hidden units in the network. However, these methods share the limitation of being computationally challenging when the size of the search space is exponential in the number of input features. Unlike these studies, Craven and Shawlik [135] proposed an alternative approach that considers the rule extraction process not as a search problem but rather as a learning problem, where the target is to identify the input(s) that correspond to each output from a trained neural network. Now, if an input-output pair was not covered by the extracted rules,

then a conjunctive rule is formed from that input-output pair, which considers all the possible antecedents. This process terminates when all input-output pairs are covered by the rules.

Taha *et al.* [136] proposed three techniques of rule extraction from ANN; the suitability of each depends on the network parameters. The first is a black-box rule extraction technique for a network with binary outputs; it generates truth tables from the trained ANN and extracts binary rules from it. The second and third techniques are link rule extraction techniques that are specific to feedforward neural networks with a single hidden layer. They both consider one node at a time, and for each such node, search for different combinations of input links whose weighted sum exceeds the bias of that node. Then, for each of these combinations, the techniques generate a rule whose premises are the input nodes to this combination. The second technique is for identifying the most important embedded knowledge with an adjustable level of detail, whereas the third one gives a more universal understanding.

Augusta and Kathirvalavakumar. [137] proposed a reverse engineering technique called RxREN, which extracts if-then rules from conventional feedforward neural networks. Other techniques of rule extraction from neural networks include genetic programming [153]–[155], function-analysis [156], Boolean function approximation of neural network nodes [157], information gain maximization of hidden layers [158], and recursive discretization of the activation values of hidden nodes [159].

Having presented the rule extraction techniques for explaining neural networks, we now present those that explain SVMs. We start with the work of Núñez *et al.* [139], who proposed a technique called SVM+Prototypes. It works by first determining the decision boundaries in the input space defined by the SVM. The next step is to identify prototype vectors representing each class by using a clustering algorithm. Finally, using geometric methods, these vectors are joined with the support vectors to define ellipsoids in the input space that can be transformed into if-then rules. In a follow-up study, the authors propose an improved version of this technique [160]. Another technique was proposed by Zhang *et al.* [140] based on support vector clustering—a non-parametric clustering algorithm that does not make any assumption on the number or shape of the clusters in the data—to find prototype vectors for each class and then define small hyper-rectangles around them. Fung *et al.* [141] proposed an algorithm based on constraint programming for converting linear SVM or other hyperplane-based linear classifiers into a set of non-overlapping and human-interpretable rules. Each iteration of the rule extraction algorithm is formulated as a constrained optimization problem that is computationally feasible. Unlike SVM+Prototypes [139], this technique does not require data pre-processing steps (e.g., clustering), which can be computationally expensive.

Barakat and Bradley [142] proposed a rule extraction method that learns rules directly from the support vectors of trained SVM using a sequential search algorithm. The rules are generated based on an ordered search of the most discriminative features measured by interclass separation. Barakat and Diedrich [143] proposed a technique that again uses the support vectors from trained SVM as well as the associated parameters to extract rules. Fu *et al.* [144] proposed a technique to extract if-then rules from SVM by using intervals defined by hyper-rectangular forms. The hyper-rectangles are created using the intersection between the support vector and the decision boundary of SVM. Chaves *et al.* [145] proposed a fuzzy rule extraction method from SVM instead of propositional rules, which increases the interpretability of the generated rules. The proposed technique consists of three steps. First, the projections of support vectors are determined on the coordinate axis, then a number of triangular fuzzy sets are constructed for each coordinate, and lastly, a rule is generated from each support vector. Other techniques include extracting if-then rules from SVM using feature selection, prediction modelling, and rule extraction [146], and extracting linear rules in local regions of the input space using a growing support vector classifier [147].

Finally, after presenting the rule extraction techniques that explain ANN and SVM, we end this subsection by presenting a technique that explains tree ensembles. In particular, Deng [148] proposed a framework called InTrees, which extracts, measures, prunes, and selects decision rules from tree ensembles and calculates frequent variable interactions. InTrees extracts the rules by treating the trade-offs among the frequency of the rules appearing in the trees, the errors made by the predictions, and the length of the rules. The described technique is also known as Simplified Tree Ensemble Learner (STEL). STEL rules may subsequently be combined into a rule-based classifier that iteratively searches for the matching rule given a new observation. Given two rules with similar frequency and accuracy, the rule with the smaller length may be preferred as it is more interpretable.

V. MODELS THAT ARE INTERPRETABLE TO START WITH

All the techniques discussed thus far are considered post-hoc, i.e., they are designed to explain trained models. Despite their advantages, such techniques have their limitations and weaknesses. In particular, recent studies [17], [161] argue that post-hoc explanations are not reliable since they are not necessarily faithful to the underlying models and present correlations rather than information about the original computation. These studies also claim that the trade-off between model accuracy and interpretability is not inevitable, especially if the data is well structured with meaningful features. The stability of post-hoc explanation techniques has also been criticized, showing that some of them are locally not stable enough [162] or, on the contrary, too stable and hence do not have sufficient local accuracy [163]. Another limitation was highlighted by Slack *et al.* [164], who demonstrate that perturbation-based outcome explanation techniques can be fooled via so-called ''scaffolding techniques''. Such techniques are able to hide

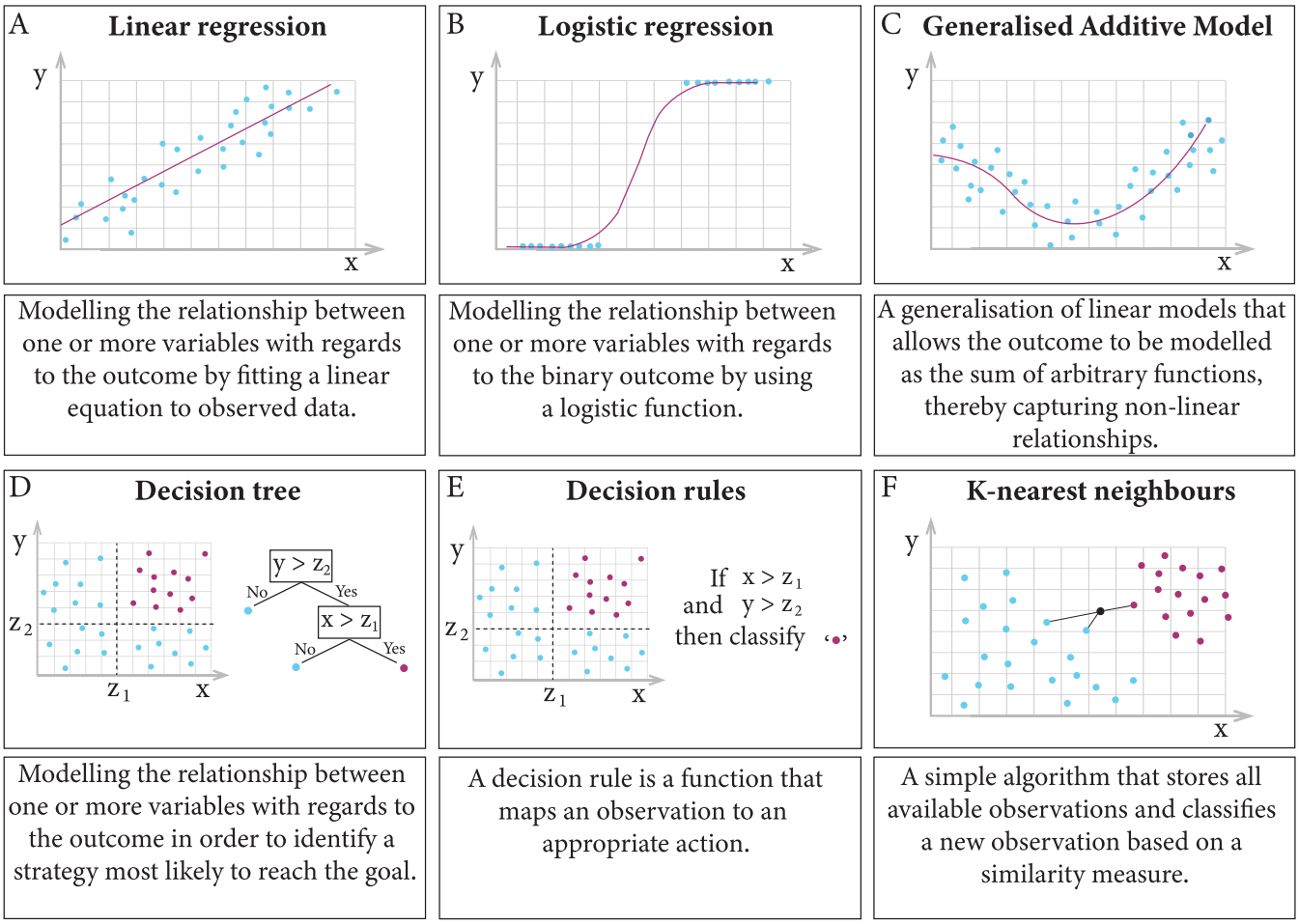

FIGURE 9. An illustration of some popular types of interpretable models.

biases in any given classifier by allowing adversarial entries to craft an arbitrarily designed explanation, which looks completely innocuous, to the point that the attack cannot be detected by the user. Note that such an attack is applicable as long as it is possible to differentiate between the perturbed instance and the input data. Finally, we mention the work by Mittelstadt *et al.* [165], who noted that more research is needed to better understand the link between the outcome of the model and the outcome of the XAI technique.

The aforementioned limitations motivate the development of models that are inherently interpretable to start with. More specifically, a model is considered interpretable if it is understandable by itself, without the need to involve any other technique. There are a number of well-known interpretable models and some recently proposed ones, both of which will be discussed next. An illustration of different types of interpretable models can be found in Figure 9.

- **Linear regression** predicts the target as a weighted sum of the input features under the assumption of independence [166]. The linearity of the learned relationship makes the model easy to interpret. Despite being interpretable, linear regression does not capture the non-linear interactions and does not have high predictive performance.
- **Logistic regression** is an extension of the linear regression model for classification problems with only two outcomes [167]. It can be extended from binary classification to multi-class classification. It has the advantage that it returns not only the classes but also the corresponding probabilities. However, the disadvantage is that it is less intuitive and harder to interpret compared to linear regression since the feature weights are not additive.
- **Generalised Additive Model (GAM)** is a linear model based on the assumption that the output variable can be modelled as a sum of arbitrary functions of each feature [116]. This way, GAM is able to capture any non-linear relationships between the features and the output variable while still being easy to interpret since it is the sum of feature effects. As such, GAM provides a smooth transition from linear models to more flexible ones. One of the disadvantages of GAM is that it heavily relies on assumptions about the data generation process.

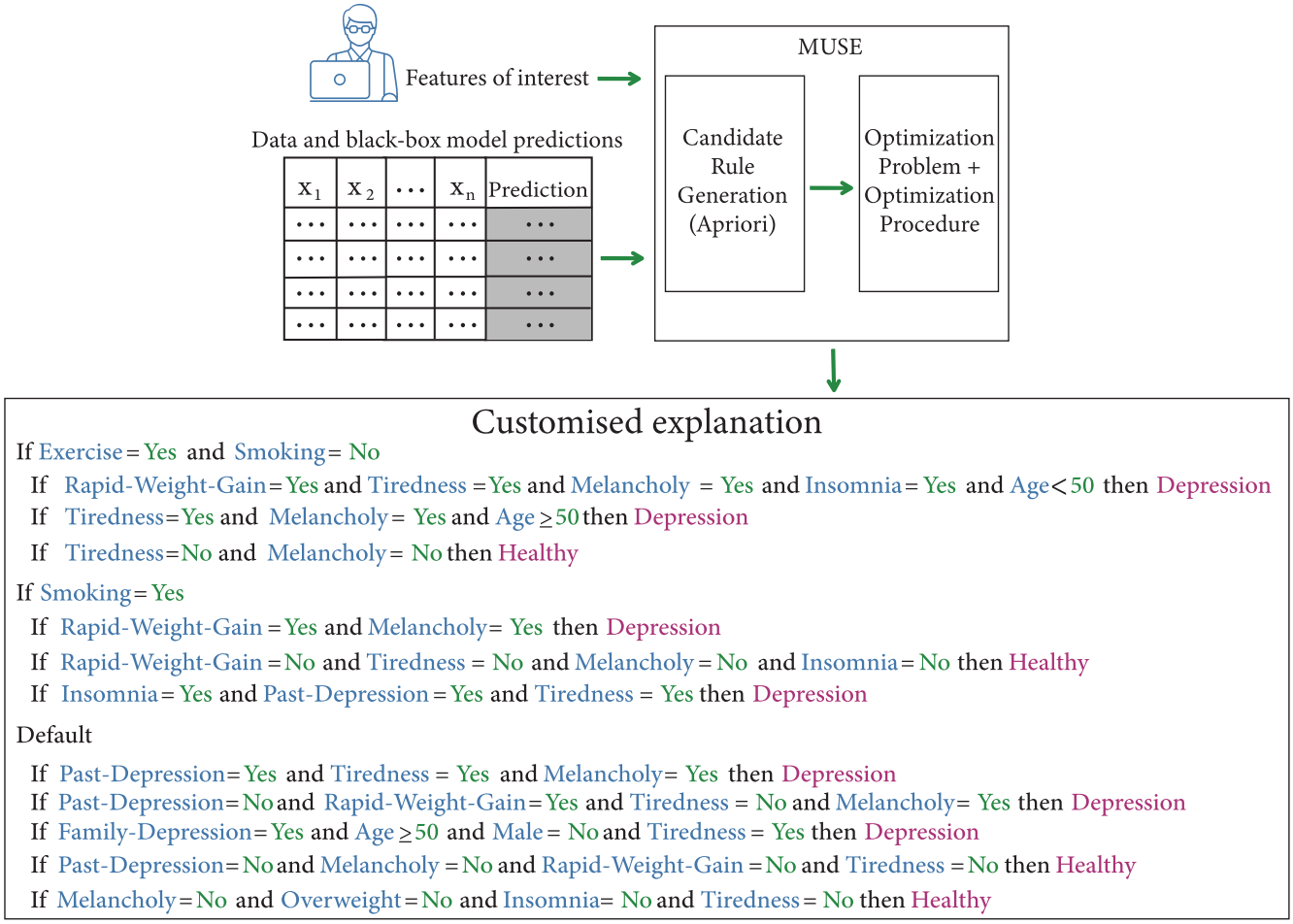

FIGURE 10. An illustration of the algorithmic flow of the MUSE approach. MUSE (which stands for Model Understanding through Subspace Explanations) takes as input the data along with the predictions of a black-box model, as well as the features of interest that are specified by the user. Then, MUSE generates decision rules based on the features of interest. For each such rule, it uses an objective function that jointly optimizes for fidelity to the original model, unambiguity and interpretability of the explanations. Finally, it gives the output in the form of customized explanations.

- **Decision trees** are suitable in situations where the relationship between features and outcome is non-linear or where the features interact with each other. Tree-based models work by splitting the input data a number of times according to certain cut-off values in the features. This leads to the creation of different subsets of data. The final subsets are called leaf nodes, and the intermediary subsets are called internal nodes. The advantages of such models are that they capture the feature interactions and are easy to visualise and interpret. However, they have the disadvantage of being not suitable for linear relationships between input features and being sensitive to small changes in the training data set.
- **Decision rules** are simple "if-then" statements, consisting of a condition and a prediction. An example of a decision rule can be: "IF it is sunny today AND it is August, THEN it will also be sunny tomorrow". A single decision rule or a combination of several ones can be used to make predictions. The percentage of instances to which the condition of a rule applies is called the support. The accuracy of a rule is a measure of how accurate the rule is in predicting the correct class for the instances to which the condition of the rule applies. Usually, there is a trade-off between accuracy and support: By adding more features to the condition, we can achieve higher accuracy but lose support. Along with being easy to interpret, decision rules have the advantage of being compact and fast to compute since they only use relevant features. Some disadvantages of decision rules are that they mostly focus on classification tasks rather than regression, and they are best used with data sets in which the features are categorical.
- **Naive Bayes classifier** uses Bayes' theorem of conditional probabilities and is based on the assumption of independence of features. It is a simple technique for constructing classifiers, and there is a family of algorithms for training such classifiers. Naive Bayes is an interpretable model because of the independence assumption and can be interpreted on the modular level [169]. However, the disadvantage is that its predictive power is relatively weak due to its "naive" assumptions.
- **K-nearest neighbours** can be used for classification as well as regression and uses the nearest neighbours of the

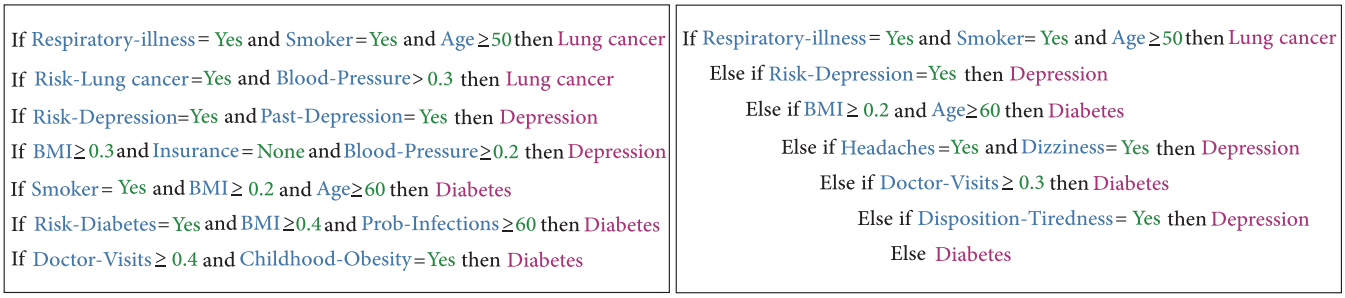

FIGURE 11. An illustration of IDS. On the left is the Interpretable Decision Set (IDS) model proposed in [168], and on the right is the typical decision list learned from the same medical diagnosis data set. Arguably, decision sets (on the left) are easier to understand and interpret because the rules apply independently. In contrast, every rule in a decision lists (on the right) depends on all the rules that are above it. Thus, while the order of the rules in decision lists is crucial, it does not matter for decision sets.

given data point to produce its prediction. For classification, it computes the distances between the point of interest and all the examples in the data and determines the class of that point based on the class that is most frequently out of the closest k examples in the data. For regression, it simply takes the average of the outcome of the neighbours. Thus, the model can be thought of as locally interpretable since the reason behind the classification of any point of interest can be understood in terms of the classification of other examples that are ''local'', i.e., close to that point. However, if the user requires additional explanation, be it local or global, this model cannot provide any.

Each of the above models can be applied by itself for classification and/or regression. There are also a number of studies utilising some of these models to build interpretable frameworks. Among the ones that extract decision rules from the data, we start with the work of Liu and Tan [170], who proposed an algorithm called X2R that can learn rules from raw data, be it numeric and discrete. The algorithm uses discretization, feature selection, and concept learning to generate rules from such data; these rules are what make the model interpretable. Yin and Han [171] proposed CPAR, which stands for Classification based on Predictive Association Rules. It takes advantage of both associative classification (higher classification accuracy) and traditional rule-based classification (generating rules directly from training data). Instead of classical rule, Wang and Rudin [172] use a Bayesian framework to extract a ''falling rule list'', which is an ordered list of if-then rules where the order of rules determines which example should be classified by each rule, and the estimated probability of success decreases monotonically down the list. Letham *et al.* [173] use Bayesian Rule Lists (BRL) which discretize a high-dimensional, multivariate feature space into a series of interpretable decision rule lists. Such a list consists of if-then rules covering the whole feature space. Another method to extract decision rules using a Bayesian framework is proposed by Wang *et al.* [174]. This method, called Bayesian Rule Sets (BRS), provides justifications behind the predictions, as well as descriptions of a different class. In BRS, the shape of the model can be controlled by the user through Bayesian priors.

Another recent example of rule extraction is the work of Lakkaraju *et al.* [168], who proposed a framework for generating prediction models by extracting Interpretable Decision Sets (IDS), i.e., independent if-then rules. Since each rule is applicable independently, decision sets are simple, brief, and easy to interpret. The authors formalize an objective function that simultaneously optimizes the accuracy and interpretability of the rules. In particular, the proposed approach can learn short, accurate, and non-overlapping rules to cover the whole feature space. An example of IDS is illustrated in Figure 11.

VI. APPLICATIONS OF XAI

XAI has been used across a wide range of domains, including healthcare [175]–[181] (Section VI-A), finance [182]–[186] (Section VI-B), criminal justice [187]–[190] (Section VI-C) and other domains [191], [192] (Section VI-D). Given our focus on tabular data, we will discuss the applications that rely on such data.

A. HEALTHCARE

Caruana *et al.* [117] applied GA^2M—a model explanation technique discussed in Section IV-A—in the healthcare domain. Specifically, the authors conducted two case studies: (i) pneumonia risk prediction; and (ii) 30-day hospital readmission. In the former case study, the goal was to predict the probability of death so that patients at high risk can be admitted to the hospital while patients at low risk are treated as outpatients. In the latter case study, the goal was to predict which patients are likely to be readmitted to the hospital within 30 days after being released from the hospital: if the patients are returned to the hospital unusually quickly, this means that the hospital did not provide adequate care at the time the patient was at the hospital. The authors showed that GA^2M can be used to explain the model's predictions for any given patient, placing the focus on the most important features for that particular patient.

Khedkar *et al.* [193] trained a neural network on electronic health records to predict heart failure risk based on the medical history of the patients. After training the neural network, the authors used LIME—an outcome explanation technique discussed in Section II—to identify the features that

contributed positively and those that contributed negatively to the heart failure risk for each patient. Devam *et al.* [194] used a variety of XAI techniques on the heart disease data set, with the goal being to demonstrate to practitioners the understandability and interpretability of XAI.

Thimoteo *et al.* [195] conducted a study whose goal is to help Brazil fight the COVID-19 pandemic more effectively. To this end, they used SHAP—an outcome explanation technique discussed in Section II—to highlight the differences (in terms of clinical features) between patients who test positive for COVID-19 and those who test negative. Using SHAP, the authors were also able to provide global interpretability of the trained models for classifying COVID-19 patients. A different application of SHAP was proposed by Hu *et al.* [196], who used it to identify the individual-level features that had the greatest impact on the 30-day mortality rate of influenza patients with critical illness in Taiwan.

For more applications of XAI in healthcare, refer to the surveys by Adadi and Berrada [197] and Pawar *et al.* [198], as well as the tutorial by Ahmad *et al.* [199].

B. FINANCE

Sachan *et al.* [200] proposed an XAI decision support system to automate the loan underwriting process by belief-rule-base (which is an extension of traditional IF-THEN rule-based systems). The proposed system explains the chain of events that lead to a particular decision for a given loan application. A business case study is presented where textual explanations are produced to justify the rejection of certain loan applications.

Benhamou *et al.* [201] used gradient boosted decision trees to analyse the stock market during the March-2020 equity crash. Their goal was to predict the risks that arise when credit is borrowed using peer-to-peer lending platforms. SHAP was then used to provide an explanation of these predictions. More specifically, they grouped the borrowers who had negative SHAP coefficients (and thus were deemed risky) and those that had positive SHAP coefficients (deemed non-risky) and then analysed each group to understand their credit score and predict their future behaviour.

Nassar *et al.* [185] proposed a general framework for achieving a more trustworthy and explainable AI by leveraging features of blockchain, smart contracts, trusted oracles, and decentralized storage. Their approach involves polling multiple predictor nodes, each running an AI model and providing an explanation to its model's outcome, in order to gradually build a reputation for each of these predictors. This framework can be applied in the financial sector, e.g., for customer profiling, tax auditing, and fraud detection. Walambe *et al.* [184] also proposed a similar blockchain-based XAI system for credit risk assessment.

An overview of the use of XAI in the financial sector is given in [186].

C. CRIMINAL JUSTICE

Loyola-González [187] analysed the Mexico City crime database, with the goal being to forecast criminal behavior. In addition to the typical features used in such studies (e.g., the crime's type, date, time, and location), the authors also considered other features related to weather. A ''contrast pattern'' mining approach was used to generate explanations. More specifically, a random forest with 100 decision trees was built using a random subset of the features for each tree. After that, human-readable patterns were extracted from the paths that connect the root node to the leaves. Finally, as a filtering stage, a small set of patterns was selected based on the user's needs.

Zhong *et al.* [188] proposed a technique called QAjudge to explain legal judgment predictions that are generated using reinforcement learning. This technique visualises the prediction generation process in a way that makes it easier for humans to understand. This is done by iteratively asking human-readable questions and then predicting the judgements based on the human-readable answers.

For a more thorough discussion of XAI applications in the area of law and criminal justice, see the surveys by Deeks [190] and Atkinson *et al.* [189].

D. OTHER DOMAINS

In addition to healthcare, finance, and criminal justice, XAI has been applied in other domains. For example, Yang *et al.* [191] used data from the Los Angeles region in 2010–2019 to explore the relationships between the injuries caused by truck crashes and built environment factors such as population density, percentage of residential land uses, road characteristics, freight generators (e.g., distance to the airport or major warehouses) and road infrastructure (e.g., the density of street lights and traffic signals). This exploration was done using a gradient boosting ML model and SHAP. In this context, the SHAP dependency plots were used to highlight any important non-linear relationships between the independent features and the dependent variables, thereby providing insights into the improvement of existing policies. Finally, we mention the work of Sargsyan *et al.* [192], who classified students based on their LIME coefficient. This allowed them to identify the students who have similar academic attainment indicators, thereby providing a more nuanced view of the success indicators.

VII. LIMITATIONS & CHALLENGES OF XAI

Despite being a powerful tool, XAI suffers from certain limitations related to accuracy and relevance (Section VII-A), robustness to adversaries (Section VII-B) as well as ethics (Section VII-C).

A. ACCURACY & RELEVANCE OF EXPLANATIONS

Payrovnaziri *et al.* [181] reviewed 42 papers on XAI models that use real-world electronic health record data, highlighting potential challenges and future directions of XAI from the

medical professionals' perspective. In particular, the authors mentioned that (i) not all visualisations produced by XAI techniques are interpretable by medical professionals, nor do they necessarily provide better interpretability; (ii) there is a need to incorporate more longitudinal features in XAI to improve the robustness of the models; (iii) the lack of a universal definition for explainability is an ongoing issue to be addressed. Some opportunities identified by these professionals include using XAI to assist medical professionals overcome their medical knowledge biases and become more objective.

Rudin [17] argues that XAI techniques do not necessarily have perfect fidelity with respect to the original black-box model they are trying to explain. Having said that, an inaccurate and low-fidelity XAI technique can limit the users' trust in the explanation and hence also their trust in the black-box model itself. The author also criticized how the term ''explanation'' is often used in a misleading way. This is because XAI techniques do not always attempt to mimic the calculations made by the original model. Such techniques may be using completely different features, making them arguably not faithful to the computation of the black-box. Finally, the author argued that the post-hoc explanations in some cases only present correlations of the underlying computations.

B. ROBUSTNESS AGAINST ADVERSARY

Slack *et al.* [164] criticised post-hoc explanation techniques that rely on input perturbations (such as LIME and SHAP), showing that they are not robust to adversarial attacks. More specifically, the authors used the COMPAS recidivism data set [202] and proposed a novel scaffolding technique capable of hiding the biases of any given classifier by allowing an adversarial entity to craft an arbitrary desired explanation. Another criticism of LIME and SHAP was made by Mittelstadt *et al.* [165], who demonstrated that their explanations can be counter-intuitive when dealing with structured data.

Kuppa and Le-Khac [203] proposed a taxonomy for XAI covering various security properties and threat models relevant to cybersecurity. The authors also proposed a novel black-box adversarial attack technique for testing the consistency, correctness, and confidence properties of gradient-based XAI methods. Using three security-relevant data sets (one of which was a tabular data set), the authors demonstrated that their technique achieves the goals of an attacker armed with threat models that are used in the real world. One of the reasons behind the success of their technique is that the explanation methods do not accurately reflect the true state of the model, thereby opening a window for the adversary to exploit both models and explainers simultaneously.

C. ETHICAL ISSUES

In addition to the aforementioned technical issues, the ethical issue is also a crucial challenge for XAI. The general ethical problem of possible discrimination (such as racism, sexism, and ageism) by AI systems naturally extends to XAI. There were a number of instances, including famous court cases, of biased decisions/actions by AI systems in the past [204]. Hester and Gray [9] observed that for black men, being tall increases the threat of being stereotyped and stopped by the police. In principle, any AI model that is based on past human-generated data might inherently yield similar biases. Care must be taken so that the AI system's decisions and the associated explanations are free from any forms of discrimination, bias, and unfairness.

O'Hara [205] emphasized the need for transparency in AI systems and the requirement for the human operator to intervene when the AI's decision and/or explanation are deemed discriminative. The author pointed out that XAI has become more important because the EU's General Data Protection Regulation (GDPR) [206] has brought explanation into its framework of data protection. The term ''explanation'' is mentioned in Recital 71, which states that ''In any case, such processing [e.g., automatic profiling] should be subject to suitable safeguards, which should include specific information to the data subject and the right to obtain human intervention, to express his or her point of view, to obtain an explanation of the decision reached after such assessment and to challenge the decision.'' GDPR issues punitive fines to transgressors, thereby increasing the awareness of the need for fair XAI.

VIII. FUTURE DIRECTIONS OF XAI

A. HUMAN-CENTERED XAI

Some recent studies started focusing on the human understandability of XAI techniques to identify opportunities for new research directions. Wang *et al.* [207] proposed a conceptual framework for building human-centered, decision-theory-driven XAI. The authors designed and experimented with an explainable clinical diagnosis system that could diagnose patients in an intensive care unit. This system not only supports the clinicians but also reduces both the diagnostic errors as well as cognitive biases. The goal was to explore how the users would interact with explanations generated from the model using a real data set. Based on these experiments, the authors concluded that the users were interested in using the sensitivity analysis to test the stability of any predicted diagnosis by asking counterfactual questions, perturbing input values, seeing a partial dependence plot, or reading through the list of counterfactual rules. If the rules were too long, the authors noticed that some users were skimming through but not reading those rules. The authors also found that the users were interested in employing not just one but rather a diverse range of XAI techniques. While much work has focused on developing various XAI techniques, more work is needed to integrate different types of explanations into a single explanation. An example of such a technique is proposed by Coppers *et al.* [208], which is designed for more sensible usage of translation suggestions.

B. BETTER USER INTERFACE/EXPERIENCE

Liao *et al.* [209] interviewed 20 user-experience-and-design practitioners working on various AI products. Based on these interviews, the authors suggested directing the XAI research attention to question-driven frameworks in order to address the users' needs. The authors also noted that the interviewed practitioners were struggling with the gaps between algorithmic output and human-consumable explanations. A similar and very recent work by Rebanal *et al.* [210] proposed an interactive approach called XAIgo, which uses question-answering to explain deterministic algorithms to non-expert users. It first classifies the question type based on a taxonomy and then generates an answer based on a set of rules that extract information from representations of the algorithm's internal states. The authors concluded that non-expert users ask both algorithm-related and concept-related questions, implying that question-answer-based XAI should be able to provide answers to both types of questions.

Antoniadi *et al.* [211] performed a literature review for XAI techniques that are applicable to clinical decision support systems. The authors concluded that there is a lack of user studies exploring the needs of clinicians in the XAI literature. They also concluded that medical experts have difficulties with a gap between the ML outputs and their explanations and highlighted the need for interdisciplinary research that not only explains ML outputs in a transparent and interpretable manner but draws inspiration from the way humans explain concepts to one another.

Bruckert *et al.* [212] argued that semantic and contextual information must be taken into account while generating explanations. They also argued that human interpretable explanations must shed light on logical as well as causal correlations. Finally, the authors noted that the implementation of explainable machine learning systems in the medical domain must draw inspiration from different disciplines and professions.

Holzinger *et al.* [213], [214] argued that, in order to assess the quality of explanation, one must distinguish between explainability and causability. The latter is defined as the extent to which an explanation of a statement to a user achieves a specified level of causal understanding with effectiveness, efficiency, and satisfaction in a specified context of use. The authors also proposed what they referred to as the System Causability Scale, the goal of which is to provide an evaluation tool to measure the quality of the explanation process as well as the quality of the explanation interface.

C. MULTIMODAL XAI

Humans can easily understand the meaning of the text, video, audio, and images together in the same context. In contrast, it is much more challenging for AI to process such ''multimodal'' signals. In recent years, there has been progress towards developing systems that can make multimodal inferences. The advantage of multimodality is its ability to extract and combine critical and comprehensive information from a variety of sources, thereby allowing for a far richer representation of the problem at hand. Several applications of multimodal AI were proposed in the literature [215], [216], but despite their outstanding performance, there is a lack of social acceptance due to their black-box nature [217]. Recently, some research was conducted for explaining multimodal AI systems [178], [218]–[222]. Out of these, we will discuss the works of Srinivasan *et al.* [219] and Holzinger *et al.* [178] since they are the only ones capable of handling tabular data—the focus of our survey.

Srinivasan et al [219] used GANs to generate user-friendly explanations for loan denials. In addition to the loan data set, the authors also conducted a survey on Amazon Mechanical Turk to understand the nature of acceptable explanations for loan applicants. Holzinger *et al.* [178] emphasized that the field of XAI has a high potential to contribute towards a better understanding of diseases in the medical field, which can lead to more accurate diagnoses and rational disease prevention strategies as well as better treatment selection. To this end, learning data from different sources and modalities can substantially outperform traditional methods, which work on just one type of data. The authors motivated the need for a novel, holistic approach to an automated medical decision pipeline building on state-of-the-art ML research while integrating the human in the loop. They demonstrated how this can be achieved using an interactive and exploration-based explainability technique called ''counterfactual graphs''. Finally, the authors emphasized the need for multimodality in every stage of this integrated approach since medical decisions are mostly directed by various influence factors stemming from a multitude of underlying signals and knowledge bases.

D. XAI INITIATIVES BY GOVERNMENT AND INDUSTRY

To stimulate the development of XAI techniques, the Defense Advanced Research Projects Agency (DARPA) launched an XAI program in 2017 [223], [224], the goal of which is to develop new techniques capable of making intelligent systems explainable. The program involves 11 teams of researchers from around the world, working towards three main strategic domains: (i) deep explanations, the aim of which is to modify the deep learning models in ways that would make them explainable; (ii) building more interpretable models, the aim of which is to build transparent models by supplementing deep learning with other AI models that are inherently explainable without greatly sacrificing the performance of AI; and (iii) model induction, the aim of which is to treat the model as a black-box and experiment with it to explain its behavior. The initiative on all three domains is focused on end-users, who are not necessarily machine learning experts.

The interest towards XAI also increases in the industry. FICO, which is the leading provider of analytics and decision management technology in the US [225], launched the Explainable Machine Learning Challenge in 2018. The goal of the challenge was to create machine learning models with both high accuracy and explainability for credit

risk assessment. In the scope of this challenge, the winner model was proposed by Chen *et al.* [226], which was a two-layer additive risk model for credit risk assessment. Although the model uses linear modelling, it ensures all the non-linearities are transparent and interpretable. Another example of model interpretability on the industrial level is Microsoft's Azure [227], which utilises three techniques: (i) SHAP, which is used to generate local interpretations; (ii) Mimic, which is used to mimic the underlying black-box models and provide global interpretations; and (iii) Permutation feature importance, which is used to explain the overall behavior of the underlying model. Another XAI platform is proposed by Kyndi [228], which is an artificial intelligence software company providing solutions for government, financial services, and healthcare through natural language processing and graph-based techniques.

IX. CONCLUSION

The Explainable Artificial Intelligence (XAI) literature is rapidly expanding due to the growing interest in the subject. As such, navigating this literature is challenging without a map that charts the XAI problems being addressed, the models being explained, and the techniques being used to provide such explanations. To the best of our knowledge, no such map exists to date for researchers interested in tabular data, which is rather surprising given the tremendous importance and popularity of such data. This article provides exactly such a map, covering a wide variety of very recent XAI studies in the context of tabular data. More specifically, we covered three fundamental problems of XAI. The first is *model explanation*, which requires explaining the underlying logic behind a black-box model. The second is *model inspection*, which requires providing visual or textual explanations of certain properties of the underlying model or its outcome, and the third is *outcome explanation*, which requires explaining the model's outcome given an instance of interest to justify the model's decision. For each of these fundamental problems, we covered two categories of techniques that are proposed in the literature. The first category consists of *model-specific techniques*, which exploit the parameters and features of the model they are designed to explain, but cannot readily be generalised to other models. The second consists of *model-agnostic techniques*, which in principle can be used on any machine learning model, but cannot take advantage of model internals, since they are only capable of analysing input-out pairs. We provided a comprehensive and up-to-date survey of these techniques, especially in the context of tabular data. Careful attention has been given to ensure that the intuitions behind the different techniques are presented in a clear manner along with illustrated examples whenever possible to make the paper accessible to readers who have a limited background in machine learning and artificial intelligence. Finally, we discussed various applications, limitations, and future directions of XAI. We hope this survey will help researchers working on tabular data sets to navigate this exciting and rapidly growing area of research.

REFERENCES

[1] K. Jensen, C. Soguero-Ruiz, K. O. Mikalsen, R.-O. Lindsetmo, I. Kouskoumvekaki, M. Girolami, S. Olav Skrovseth, and K. M. Augestad, "Analysis of free text in electronic health records for identification of cancer patient trajectories," *Sci. Rep.*, vol. 7, no. 1, May 2017, Art. no. 46226.

[2] M. K. Lodhi, R. Ansari, Y. Yao, G. M. Keenan, D. Wilkie, and A. A. Khokhar, "Predicting hospital re-admissions from nursing care data of hospitalized patients," in *Proc. Ind. Conf. Data Mining*, 2017, pp. 181–193.

[3] N. M. Davies, M. Dickson, G. D. Smith, G. J. van den Berg, and F. Windmeijer, "The causal effects of education on health outcomes in the UK biobank," *Nature Hum. Behav.*, vol. 2, no. 2, pp. 117–125, Feb. 2018.

[4] Y.-Q. Chen, J. Zhang, and W. W. Y. Ng, "Loan default prediction using diversified sensitivity undersampling," in *Proc. Int. Conf. Mach. Learn. Cybern. (ICMLC)*, Jul. 2018, pp. 240–245.

[5] A. Byanjankar, M. Heikkila, and J. Mezei, "Predicting credit risk in peer-to-peer lending: A neural network approach," in *Proc. IEEE Symp. Ser. Comput. Intell.*, Dec. 2015, pp. 719–725.

[6] M. Ala'raj and M. F. Abbod, "Classifiers consensus system approach for credit scoring," *Knowl.-Based Syst.*, vol. 104, pp. 89–105, Jul. 2016.

[7] M.-C. Tsai, S.-P. Lin, C.-C. Cheng, and Y.-P. Lin, "The consumer loan default predicting model—An application of DEA–DA and neural network," *Expert Syst. Appl.*, vol. 36, no. 9, pp. 11682–11690, Nov. 2009.

[8] Z. Zhang and D. B. Neill, "Identifying significant predictive bias in classifiers," 2016, *arXiv:1611.08292*. [Online]. Available: http://arxiv.org/abs/1611.08292

[9] N. Hester and K. Gray, "For black men, being tall increases threat stereotyping and police stops," *Proc. Nat. Acad. Sci. USA*, vol. 115, no. 11, pp. 2711–2715, Mar. 2018.

[10] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. (2016). *Machine Bias: There's Software Used Across the Country to Predict Future Criminals. And it's Biased Against Blacks*. [Online]. Available: https://www.propublica.org/article/machine-bias-risk-assessments-in-cri%minal-sentencing

[11] H. K. Dam, T. Tran, and A. Ghose, "Explainable software analytics," in *Proc. 40th Int. Conf. Softw. Eng., New Ideas Emerg. Results*, May 2018, pp. 53–56.

[12] B. Goodman and S. Flaxman, "European Union regulations on algorithmic decision-making and a 'right to explanation,'" *AI Mag.*, vol. 38, no. 3, pp. 50–57, 2017.

[13] F. K. Dosilovic, M. Brcic, and N. Hlupic, "Explainable artificial intelligence: A survey," in *Proc. 41st Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2018, pp. 210–215.

[14] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artif. Intell.*, vol. 267, pp. 1–38, Feb. 2019.

[15] Z. C. Lipton, "The mythos of model interpretability," *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[16] F. Hayes-Roth, "Rule-based systems," *Commun. ACM*, vol. 28, no. 9, pp. 921–932, 1985.

[17] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Mach. Intell.*, vol. 1, no. 5, pp. 206–215, 2019.

[18] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA*, vol. 79, no. 8, pp. 2554–2558, 1982.

[19] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[20] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–42, Jan. 2019.

[21] C. Molnar. (2019). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. [Online]. Available: https://christophm.github.io/interpretable-ml-book

[22] C. Molnar. (2021). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. https://christophm.github.io/interpretable-ml-book

[23] B. Poulin, R. Eisner, D. Szafron, P. Lu, R. Greiner, D. S. Wishart, A. Fyshe, B. Pearcy, C. MacDonell, and J. Anvik, "Visual explanation of evidence with additive classifiers," in *Proc. 18th Conf. Innovative Appl. Artif. Intell. (IAAI)*, vol. 2, 2006, pp. 1822–1829.

[24] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?': Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1135–1144.
[25] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 1527–1535.
[26] M. R. Zafar and N. M. Khan, "DLIME: A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems," 2019, *arXiv:1906.10263*. [Online]. Available: http://arxiv.org/abs/1906.10263
[27] R. ElShawi, Y. Sherif, M. Al-Mallah, and S. Sakr, "ILIME: Local and global interpretable model-agnostic explainer of black-box decision," in *Proc. Eur. Conf. Adv. Databases Inf. Syst.*, 2019, pp. 53–68.
[28] S. M. Shankaranarayana and D. Runje, "ALIME: Autoencoder based approach for local interpretability," in *Proc. Int. Conf. Intell. Data Eng. Automated Learn.*, 2019, pp. 454–463.
[29] C. Panigutti, A. Perotti, and D. Pedreschi, "Doctor XAI: An ontology-based approach to black-box sequential data classification explanations," in *Proc. Conf. Fairness, Accountability, Transparency*, Jan. 2020, pp. 629–639.
[30] G. Plumb, D. Molitor, and A. S. Talwalkar, "Model agnostic supervised local explanations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 2515–2524.
[31] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, and F. Giannotti, "Local rule-based explanations of black box decision systems," 2018, *arXiv:1805.10820*. [Online]. Available: http://arxiv.org/abs/1805.10820
[32] J. Lei, M. G'Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman, "Distribution-free predictive inference for regression," *J. Amer. Stat. Assoc.*, vol. 113, no. 523, pp. 1094–1111, Jul. 2018.
[33] R. Turner, "A model explanation system," in *Proc. IEEE 26th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Sep. 2016, pp. 1–6.
[34] M. M.-C. Vidovic, N. Görnitz, K.-R. Müller, and M. Kloft, "Feature importance measure for non-linear learning algorithms," 2016, *arXiv:1611.07567*. [Online]. Available: http://arxiv.org/abs/1611.07567
[35] E. Strumbelj and I. Kononenko, "An efficient explanation of individual classifications using game theory," *J. Mach. Learn. Res.*, vol. 11, pp. 1–18, Jan. 2010.
[36] E. Štrumbelj and I. Kononenko, "Explaining prediction models and individual predictions with feature contributions," *Knowl. Inf. Syst.*, vol. 41, no. 3, pp. 647–665, Dec. 2014.
[37] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4765–4774.
[38] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 3145–3153.
[39] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable ai for trees," *Nature Mach. Intell.*, vol. 2, no. 1, pp. 56–67, 2020.
[40] A. Saabas. (2019). *Treeinterpreter Python Package*. [Online]. Available: https://github.com/andosa/treeinterpreter
[41] S. Haufe, F. Meinecke, K. Görgen, S. Dähne, J.-D. Haynes, B. Blankertz, and F. Bießmann, "On the interpretation of weight vectors of linear models in multivariate neuroimaging," *NeuroImage*, vol. 87, pp. 96–110, Feb. 2014.
[42] I. Mollas, N. Bassiliades, I. Vlahavas, and G. Tsoumakas, "LionForests: Local interpretation of random forests," 2019, *arXiv:1911.08780*. [Online]. Available: http://arxiv.org/abs/1911.08780
[43] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5188–5196.
[44] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.
[45] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS ONE*, vol. 10, no. 7, Jul. 2015, Art. no. e0130140.
[46] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3429–3437.
[47] M. Pawelczyk, K. Broelemann, and G. Kasneci, "Learning model-agnostic counterfactual explanations for tabular data," in *Proc. Web Conf.*, Apr. 2020, pp. 3126–3132.
[48] D. Madigan, K. Mosurski, and R. G. Almond, "Graphical explanation in belief networks," *J. Comput. Graph. Statist.*, vol. 6, no. 2, pp. 160–181, Jun. 1997.
[49] M. Možina, J. Demšar, M. Kattan, and B. Zupan, "Nomograms for visualization of naive Bayesian classifier," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discovery*, 2004, pp. 337–348.
[50] I. Kononenko, "Inductive and Bayesian learning in medical diagnosis," *Appl. Artif. Intell.*, vol. 7, no. 4, pp. 317–337, Oct. 1993.
[51] M. Robnik-Šikonja and I. Kononenko, "Explaining classifications for individual instances," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 5, pp. 589–600, May 2008.
[52] D. Alvarez-Melis and T. S. Jaakkola, "On the robustness of interpretability methods," 2018, *arXiv:1806.08049*. [Online]. Available: http://arxiv.org/abs/1806.08049
[53] G. Visani, E. Bagli, F. Chesani, A. Poluzzi, and D. Capuzzo, "Statistical stability indices for LIME: Obtaining reliable explanations for machine learning models," 2020, *arXiv:2001.11757*. [Online]. Available: http://arxiv.org/abs/2001.11757
[54] G. Visani, E. Bagli, and F. Chesani, "OptiLIME: Optimized LIME explanations for diagnostic computer algorithms," 2020, *arXiv:2006.05714*. [Online]. Available: http://arxiv.org/abs/2006.05714
[55] M. T. Ribeiro, S. Singh, and C. Guestrin, "Model-agnostic interpretability of machine learning," 2016, *arXiv:1606.05386*. [Online]. Available: http://arxiv.org/abs/1606.05386
[56] M. T. Ribeiro, S. Singh, and C. Guestrin, "Nothing else matters: Model-agnostic explanations by identifying prediction invariance," 2016, *arXiv:1611.05817*. [Online]. Available: http://arxiv.org/abs/1611.05817
[57] S. Singh, M. T. Ribeiro, and C. Guestrin, "Programs as black-box explanations," 2016, *arXiv:1611.07579*. [Online]. Available: http://arxiv.org/abs/1611.07579
[58] L. S. Shapley, "A value for n-person games," in *Contributions to Theory Games*, vol. 2. Princeton, NJ, USA: Princeton Univ. Press, 1953, pp. 307–317.
[59] S. Maleki, L. Tran-Thanh, G. Hines, T. Rahwan, and A. Rogers, "Bounding the estimation error of sampling-based Shapley value approximation," 2013, *arXiv:1306.4265*. [Online]. Available: http://arxiv.org/abs/1306.4265
[60] S. Lipovetsky and M. Conklin, "Analysis of regression in game theory approach," *Appl. Stochastic Models Bus. Ind.*, vol. 17, no. 4, pp. 319–330, 2001.
[61] A. Keinan, B. Sandbank, C. Hilgetag, I. Meilijson, and E. Ruppin, "Fair attribution of functional contribution in artificial and biological networks," *Neural Comput.*, vol. 16, no. 9, pp. 1887–1915, Sep. 2004.
[62] S. Cohen, G. Dror, and E. Ruppin, "Feature selection via coalitional game theory," *Neural Comput.*, vol. 19, no. 7, pp. 1939–1961, Jul. 2007.
[63] E. Štrumbelj, I. Kononenko, and M. R. Šikonja, "Explaining instance classifications with interactions of subsets of feature values," *Data Knowl. Eng.*, vol. 68, no. 10, pp. 886–904, Oct. 2009.
[64] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, "Not just a black box: Learning important features through propagating activation differences," 2016, *arXiv:1605.01713*. [Online]. Available: http://arxiv.org/abs/1605.01713
[65] S. M. Lundberg, G. G. Erion, and S.-I. Lee, "Consistent individualized feature attribution for tree ensembles," 2018, *arXiv:1802.03888*. [Online]. Available: http://arxiv.org/abs/1802.03888
[66] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *Harvard J. Law Technol.*, vol. 31, no. 2, pp. 841–887, 2018.
[67] R. K. Mothilal, A. Sharma, and C. Tan, "Explaining machine learning classifiers through diverse counterfactual explanations," in *Proc. Conf. Fairness, Accountability, Transparency*, Jan. 2020, pp. 607–617.
[68] B. Ustun, A. Spangher, and Y. Liu, "Actionable recourse in linear classification," in *Proc. Conf. Fairness, Accountability, Transparency*, Jan. 2019, pp. 10–19.
[69] S. Joshi, O. Koyejo, W. Vijitbenjaronk, B. Kim, and J. Ghosh, "Towards realistic individual recourse and actionable explanations in black-box decision making systems," 2019, *arXiv:1907.09615*. [Online]. Available: http://arxiv.org/abs/1907.09615

[70] A. Van Looveren and J. Klaise, "Interpretable counterfactual explanations guided by prototypes," 2019, *arXiv:1907.02584*. [Online]. Available: http://arxiv.org/abs/1907.02584

[71] C. Russell, "Efficient search for diverse coherent explanations," in *Proc. Conf. Fairness, Accountability, Transparency*, Jan. 2019, pp. 20–28.

[72] D. Martens and F. Provost, "Explaining data-driven document classifications," *MIS Quart.*, vol. 38, no. 1, pp. 73–100, 2014.

[73] B. Kment, "Counterfactuals and explanation," *Mind*, vol. 115, no. 458, pp. 261–310, Apr. 2006.

[74] R. M. Byrne, "Precis of the rational imagination: How people create alternatives to reality," *Behav. Brain Sci.*, vol. 30, nos. 5–6, p. 439, 2007.

[75] S. Verma, J. Dickerson, and K. Hines, "Counterfactual explanations for machine learning: A review," 2020, *arXiv:2010.10596*. [Online]. Available: http://arxiv.org/abs/2010.10596

[76] R. Mc Grath, L. Costabello, C. Le Van, P. Sweeney, F. Kamiab, Z. Shen, and F. Lecue, "Interpretable credit application predictions with counterfactual explanations," 2018, *arXiv:1811.05245*. [Online]. Available: http://arxiv.org/abs/1811.05245

[77] A.-H. Karimi, G. Barthe, B. Balle, and I. Valera, "Model-agnostic counterfactual explanations for consequential decisions," in *Proc. Int. Conf. Artif. Intell. Statist.*, Jun. 2020, pp. 895–905.

[78] A. White and A. d'Avila Garcez, "Measurable counterfactual local explanations for any classifier," 2019, *arXiv:1908.03020*. [Online]. Available: http://arxiv.org/abs/1908.03020

[79] T. Laugel, M.-J. Lesot, C. Marsala, X. Renard, and M. Detyniecki, "The dangers of post-hoc interpretability: Unjustified counterfactual explanations," 2019, *arXiv:1907.09294*. [Online]. Available: http://arxiv.org/abs/1907.09294

[80] S. Sharma, J. Henderson, and J. Ghosh, "CERTIFAI: Counterfactual explanations for robustness, transparency, interpretability, and fairness of artificial intelligence models," 2019, *arXiv:1905.07857*. [Online]. Available: http://arxiv.org/abs/1905.07857

[81] R. Poyiadzi, K. Sokol, R. Santos-Rodriguez, T. De Bie, and P. Flach, "FACE: Feasible and actionable counterfactual explanations," in *Proc. AAAI/ACM Conf. AI, Ethics, Soc.*, Feb. 2020, pp. 344–350.

[82] S. Dandl, C. Molnar, M. Binder, and B. Bischl, "Multi-objective counterfactual explanations," in *Proc. Int. Conf. Parallel Problem Solving From Nature*. Leiden, The Netherlands: Springer, 2020, pp. 448–469.

[83] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, "How to explain individual classification decisions," *J. Mach. Learn. Res.*, vol. 11, no. 6, pp. 1803–1831, 2010.

[84] P. Cortez and M. J. Embrechts, "Opening black box data mining models using sensitivity analysis," in *Proc. IEEE Symp. Comput. Intell. Data Mining (CIDM)*, Apr. 2011, pp. 341–348.

[85] P. Cortez and M. J. Embrechts, "Using sensitivity analysis and visualization techniques to open black box data mining models," *Inf. Sci.*, vol. 225, pp. 1–17, Mar. 2013.

[86] A. Datta, S. Sen, and Y. Zick, "Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 598–617.

[87] G. Hooker, "Discovering additive structure in black box functions," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2004, pp. 575–580.

[88] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin, "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation," *J. Comput. Graph. Statist.*, vol. 24, no. 1, pp. 44–65, 2015.

[89] J. Krause, A. Perer, and K. Ng, "Interacting with predictions: Visual inspection of black-box machine learning models," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, May 2016, pp. 5686–5697.

[90] G. Casalicchio, C. Molnar, and B. Bischl, "Visualizing the feature importance for black box models," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, pp. 655–670, 2018.

[91] P. Adler, C. Falk, S. A. Friedler, T. Nix, G. Rybeck, C. Scheidegger, B. Smith, and S. Venkatasubramanian, "Auditing black-box models for indirect influence," *Knowl. Inf. Syst.*, vol. 54, no. 1, pp. 95–122, Jan. 2018.

[92] J. Adebayo and L. Kagal, "Iterative orthogonal feature projection for diagnosing bias in black-box models," 2016, *arXiv:1611.04967*. [Online]. Available: http://arxiv.org/abs/1611.04967

[93] R. Shwartz-Ziv and N. Tishby, "Opening the black box of deep neural networks via information," 2017, *arXiv:1703.00810*. [Online]. Available: http://arxiv.org/abs/1703.00810

[94] J. J. Thiagarajan, B. Kailkhura, P. Sattigeri, and K. N. Ramamurthy, "TreeView: Peeking into deep neural networks via feature-space partitioning," 2016, *arXiv:1611.07429*. [Online]. Available: http://arxiv.org/abs/1611.07429

[95] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 3319–3328.

[96] J. D. Olden and D. A. Jackson, "Illuminating the 'black box': A randomization approach for understanding variable contributions in artificial neural networks," *Ecolog. Model.*, vol. 154, nos. 1–2, pp. 135–150, 2002.

[97] A. Palczewska, J. Palczewski, R. M. Robinson, and D. Neagu, "Interpreting random forest classification models using a feature contribution method," in *Integration of Reusable Systems*. Cham, Switzerland: Springer, 2014, pp. 193–218.

[98] L. Auret and C. Aldrich, "Interpretation of nonlinear relationships between process variables by use of random forests," *Minerals Eng.*, vol. 35, pp. 27–42, Aug. 2012.

[99] S. H. Welling, H. H. F. Refsgaard, P. B. Brockhoff, and L. H. Clemmensen, "Forest floor visualizations of random forests," 2016, *arXiv:1605.09196*. [Online]. Available: http://arxiv.org/abs/1605.09196

[100] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[101] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. *Modeling Wine Preferences by Data Mining From Physicochemical Properties*. Accessed: Jul. 6, 2021. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Wine+Quality

[102] A. Saltelli, S. Tarantola, F. Campolongo, and M. Ratto, *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*, vol. 1. Hoboken, NJ, USA: Wiley, 2004.

[103] P. Cortez, J. Teixeira, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Using data mining for wine quality assessment," in *Proc. Int. Conf. Discovery Sci. (DS)*, 2009, pp. 66–79.

[104] J. Krause, A. Perer, and E. Bertini, "Using visual analytics to interpret predictive machine learning models," 2016, *arXiv:1606.05685*. [Online]. Available: http://arxiv.org/abs/1606.05685

[105] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 259–268.

[106] B. Noble and J. W. Daniel, *Applied Linear Algebra*, vol. 3. Upper Saddle River, NJ, USA: Prentice-Hall, 1988.

[107] J. A. Adebayo, "FairML: ToolBox for diagnosing bias in predictive modeling," M.S. thesis, Dept. Elect. Eng. Comput., Massachusetts Inst. Technol., Cambridge, MA, USA, 2016.

[108] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Apr. 2015, pp. 1–5.

[109] Y. Lou, R. Caruana, and J. Gehrke, "Intelligible models for classification and regression," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2012, pp. 150–158.

[110] Y. Lou, R. Caruana, J. Gehrke, and G. Hooker, "Accurate intelligible models with pairwise interactions," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2013, pp. 623–631.

[111] A. Henelius, K. Puolamäki, H. Boström, L. Asker, and P. Papapetrou, "A peek into the black box: Exploring classifiers by randomization," *Data Mining Knowl. Discovery*, vol. 28, nos. 5–6, pp. 1503–1529, Sep. 2014.

[112] S. Krishnan and E. Wu, "PALM: Machine learning explanations for iterative debugging," in *Proc. 2nd Workshop Hum. Loop Data Anal.*, May 2017, pp. 1–6.

[113] A. Zien, N. Krämer, S. Sonnenburg, and G. Rätsch, "The feature importance ranking measure," in *Proc. 2009 Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, pp. 694–709, 2009.

[114] O. Bastani, C. Kim, and H. Bastani, "Interpreting blackbox models via model extraction," 2017, *arXiv:1705.08504*. [Online]. Available: http://arxiv.org/abs/1705.08504

[115] H. Lakkaraju, E. Kamar, R. Caruana, and J. Leskovec, "Faithful and customizable explanations of black box models," in *Proc. AAAI/ACM Conf. AI, Ethics, Soc.*, Jan. 2019, pp. 131–138.

[116] T. J. Hastie, "Generalized additive models," in *Statistical Models in S*. Evanston, IL, USA: Routledge, 2017, pp. 249–307.

[117] 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 *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 1721–1730.
[118] A. Henelius, K. Puolamäki, I. Karlsson, J. Zhao, L. Asker, H. Boström, and P. Papapetrou, "Goldeneye++: A closer look into the black box," in *Proc. Int. Symp. Stat. Learn. Data Sci. (SLDS)*, 2015, pp. 96–105.
[119] S. Sonnenburg, A. Zien, P. Philips, and G. Rätsch, "POIMs: Positional oligomer importance matrices—Understanding support vector machine-based signal detectors," *Bioinformatics*, vol. 24, no. 13, pp. i6–i14, 2008.
[120] H. Lakkaraju, E. Kamar, R. Caruana, and J. Leskovec, "Interpretable & explorable approximations of black box models," 2017, *arXiv:1707.01154*. [Online]. Available: http://arxiv.org/abs/1707.01154
[121] M. Craven and J. W. Shavlik, "Extracting tree-structured representations of trained networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 1996, pp. 24–30.
[122] R. Krishnan, G. Sivakumar, and P. Bhattacharya, "Extracting decision trees from trained neural networks," *Pattern Recognit.*, vol. 32, no. 12, pp. 1999–2009, Dec. 1999.
[123] G. P. J. Schmitz, C. Aldrich, and F. S. Gouws, "ANN-DT: An algorithm for extraction of decision trees from artificial neural networks," *IEEE Trans. Neural Netw.*, vol. 10, no. 6, pp. 1392–1401, Nov. 1999.
[124] O. Boz, "Extracting decision trees from trained neural networks," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2002, pp. 456–461.
[125] U. Johansson and L. Niklasson, "Evolving decision trees using Oracle guides," in *Proc. IEEE Symp. Comput. Intell. Data Mining*, Mar. 2009, pp. 238–244.
[126] Z. Che, S. Purushotham, R. Khemani, and Y. Liu, "Interpretable deep models for ICU outcome prediction," in *Proc. AMIA Annu. Symp.*, 2016, pp. 371–380.
[127] M. Wu, M. C. Hughes, S. Parbhoo, M. Zazzi, V. Roth, and F. Doshi-Velez, "Beyond sparsity: Tree regularization of deep models for interpretability," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, 2018, pp. 1670–1678.
[128] H. A. Chipman, E. I. George, and R. E. McCulloh, "Making sense of a forest of trees," in *Proc. 30th Symp. Interface*, vol. 29, S. Weisberg, Ed. Fairfax Station, VA, USA: Interface Foundation of North America, Aug. 1998, pp. 84–92.
[129] P. Domingos, "Knowledge discovery via multiple models," *Intell. Data Anal.*, vol. 2, no. 3, pp. 187–202, Jul. 1998.
[130] 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," *J. Clin. Psychiatry*, vol. 74, no. 7, pp. 669–674, 2013.
[131] Y. Zhou and G. Hooker, "Interpreting models via single tree approximation," 2016, *arXiv:1610.09036*. [Online]. Available: http://arxiv.org/abs/1610.09036
[132] S. Tan, M. Soloviev, G. Hooker, and M. T. Wells, "Tree space prototypes: Another look at making tree ensembles interpretable," 2016, *arXiv:1611.07115*. [Online]. Available: http://arxiv.org/abs/1611.07115
[133] 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 Trans. Inf. Technol. Biomed.*, vol. 11, no. 3, pp. 312–319, May 2007.
[134] S. Hara and K. Hayashi, "Making tree ensembles interpretable," 2016, *arXiv:1606.05390*. [Online]. Available: http://arxiv.org/abs/1606.05390
[135] M. W. Craven and J. W. Shavlik, "Using sampling and queries to extract rules from trained neural networks," in *Proc. 11th Int. Conf. Mach. Learn. (ICML)*, 1994, pp. 37–45.
[136] I. A. Taha and J. Ghosh, "Symbolic interpretation of artificial neural networks," *IEEE Trans. Knowl. Data Eng.*, vol. 11, no. 3, pp. 448–463, May 1999.
[137] M. G. Augasta and T. Kathirvalavakumar, "Reverse engineering the neural networks for rule extraction in classification problems," *Neural Process. Lett.*, vol. 35, no. 2, pp. 131–150, Apr. 2012.
[138] H. Nunez, C. Angulo, and A. Catala, "Support vector machines with symbolic interpretation," in *Proc. VII Brazilian Symp. Neural Netw. (SBRN)*, Nov. 2002, pp. 142–147.
[139] H. Nú nez, C. Angulo, and A. Català, "Rule extraction from support vector machines," in *Proc. 10th Eurorean Symp. Artif. Neural Netw. (ESANN)*, 2002, pp. 107–112.
[140] Y. Zhang, H. Su, T. Jia, and J. Chu, "Rule extraction from trained support vector machines," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2005, pp. 61–70.
[141] G. Fung, S. Sandilya, and R. B. Rao, "Rule extraction from linear support vector machines," in *Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2005, pp. 32–40.
[142] N. H. Barakat, N. H. Barakat, A. P. Bradley, and A. P. Bradley, "Rule extraction from support vector machines: A sequential covering approach," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 6, pp. 729–741, Jun. 2007.
[143] N. Barakat and J. Diederich, "Eclectic rule-extraction from support vector machines," *Int. J. Comput. Intell.*, vol. 2, no. 1, pp. 59–62, 2005.
[144] X. Fu, C. Ong, S. Keerthi, G. Guang Hung, and L. Goh, "Extracting the knowledge embedded in support vector machines," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul. 2004, pp. 291–296.
[145] A. C. F. Chaves, M. M. B. R. Vellasco, and R. Tanscheit, "Fuzzy rule extraction from support vector machines," in *Proc. 5th Int. Conf. Hybrid Intell. Syst. (HIS)*, Nov. 2005, pp. 1–6.
[146] Z. Chen, J. Li, and L. Wei, "A multiple kernel support vector machine scheme for feature selection and rule extraction from gene expression data of cancer tissue," *Artif. Intell. Med.*, vol. 41, no. 2, pp. 161–175, Oct. 2007.
[147] A. Navia-Vázquez and E. Parrado-Hernández, "Support vector machine interpretation," *Neurocomputing*, vol. 69, nos. 13–15, pp. 1754–1759, Aug. 2006.
[148] H. Deng, "Interpreting tree ensembles with inTrees," *Int. J. Data Sci. Anal.*, vol. 7, no. 4, pp. 277–287, Jun. 2019.
[149] M. W. Craven and J. W. Shavlik, "Learning symbolic rules using artificial neural networks," in *Proc. 10th Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 73–80.
[150] L. Fu, "Rule learning by searching on adapted nets," in *Proc. 9th AAAI Conf. Artif. Intell. (AAAI)*, vol. 91, 1991, pp. 590–595.
[151] G. G. Towell and J. W. Shavlik, "Extracting refined rules from knowledge-based neural networks," *Mach. Learn.*, vol. 13, no. 1, pp. 71–101, 1993.
[152] S. I. Gallant, *Neural Network Learning and Expert Systems*. Cambridge, MA, USA: MIT Press, 1993.
[153] U. Johansson, L. Niklasson, and R. König, "Accuracy vs. comprehensibility in data mining models," in *Proc. 7th Int. Conf. Inf. Fusion (FUSION)*, vol. 1, 2004, pp. 295–300.
[154] U. Johansson, R. König, and L. Niklasson, "The truth is in there-rule extraction from opaque models using genetic programming," in *Proc. 17th Int. Florida Artif. Intell. Res. Society Conf. (FLAIRS)*, 2004, pp. 658–663.
[155] A. D. Arbatli and H. L. Akin, "Rule extraction from trained neural networks using genetic algorithms," *Nonlinear Anal., Theory, Methods Appl.*, vol. 30, no. 3, pp. 1639–1648, Dec. 1997.
[156] Z. Zhou, Y. Jiang, and S. Chen, "Extracting symbolic rules from trained neural network ensembles," *AI Commun.*, vol. 16, no. 1, pp. 3–15, May 2003.
[157] H. Tsukimoto, "Extracting rules from trained neural networks," *IEEE Trans. Neural Netw.*, vol. 11, no. 2, pp. 377–389, Mar. 2000.
[158] R. Setiono and W. K. Leow, "FERNN: An algorithm for fast extraction of rules from neural networks," *Appl. Intell.*, vol. 12, nos. 1–2, pp. 15–25, 2000.
[159] R. Setiono and H. Liu, "Understanding neural networks via rule extraction," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, vol. 1, 1995, pp. 480–485.
[160] H. Núñez, C. Angulo, and A. Català, "Rule-based learning systems for support vector machines," *Neural Process. Lett.*, vol. 24, no. 1, pp. 1–18, Aug. 2006.
[161] C. Rudin and J. Radin, "Why are we using black box models in AI when we don't need to? A lesson from an explainable AI competition," *Harvard Data Sci. Rev.*, vol. 1, no. 2, pp. 1–9, 2019.
[162] D. A. Melis and T. Jaakkola, "Towards robust interpretability with self-explaining neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7775–7784.
[163] T. Laugel, X. Renard, M.-J. Lesot, C. Marsala, and M. Detyniecki, "Defining locality for surrogates in post-hoc interpretablity," 2018, *arXiv:1806.07498*. [Online]. Available: http://arxiv.org/abs/1806.07498
[164] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju, "Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods," 2019, *arXiv:1911.02508*. [Online]. Available: http://arxiv.org/abs/1911.02508
[165] B. Mittelstadt, C. Russell, and S. Wachter, "Explaining explanations in AI," in *Proc. Conf. Fairness, Accountability, Transparency*, Jan. 2019, pp. 279–288.

[166] G. A. F. Seber and A. J. Lee, *Linear Regression Analysis*, 2nd ed. Hoboken, NJ, USA: Wiley, 2003.
[167] D. W. Hosmer Jr., S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, 3rd ed. Hoboken, NJ, USA: Wiley, 2013.
[168] H. Lakkaraju, S. H. Bach, and J. Leskovec, "Interpretable decision sets: A joint framework for description and prediction," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1675–1684.
[169] G. M. Koop, *Bayesian Econometrics*. Hoboken, NJ, USA: Wiley, 2003.
[170] H. Liu and S. Teck Tan, "X2R: A fast rule generator," in *Proc. IEEE Int. Conf. Syst., Man Cybern. Intell. Syst. 21st Century*, Oct. 1995, pp. 1631–1635.
[171] X. Yin and J. Han, "CPAR: Classification based on predictive association rules," in *Proc. SIAM Int. Conf. Data Mining*, May 2003, pp. 331–335.
[172] F. Wang and C. Rudin, "Falling rule lists," in *Proc. 18th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2015, pp. 1013–1022.
[173] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan, "Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model," *Ann. Appl. Statist.*, vol. 9, no. 3, pp. 1350–1371, 2015.
[174] T. Wang, C. Rudin, F. Doshi-Velez, Y. Liu, E. Klampfl, and P. MacNeille, "A Bayesian framework for learning rule sets for interpretable classification," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 2357–2393, 2017.
[175] M. S. Hossain, G. Muhammad, and N. Guizani, "Explainable AI and mass surveillance system-based healthcare framework to combat COVID-I9 like pandemics," *IEEE Netw.*, vol. 34, no. 4, pp. 126–132, Jul. 2020.
[176] A. Singh, S. Sengupta, and V. Lakshminarayanan, "Explainable deep learning models in medical image analysis," *J. Imag.*, vol. 6, no. 6, p. 52, Jun. 2020.
[177] A. Y. Zhang, S. S. W. Lam, M. E. H. Ong, P. H. Tang, and L. L. Chan, "Explainable AI: Classification of MRI brain scans orders for quality improvement," in *Proc. 6th IEEE/ACM Int. Conf. Big Data Comput., Appl. Technol. (BDCAT)*, 2019, pp. 95–102.
[178] A. Holzinger, B. Malle, A. Saranti, and B. Pfeifer, "Towards multi-modal causability with graph neural networks enabling information fusion for explainable AI," *Inf. Fusion*, vol. 71, pp. 28–37, Jul. 2021.
[179] S. S. Samuel, N. N. B. Abdullah, and A. Raj, "Interpretation of SVM using data mining technique to extract syllogistic rules," in *Proc. Int. Cross-Domain Conf. Mach. Learn. Knowl. Extraction (CD-MAKE)*, 2020, pp. 249–266.
[180] W. Hryniewska, P. Bombiński, P. Szatkowski, P. Tomaszewska, A. Przelaskowski, and P. Biecek, "Checklist for responsible deep learning modeling of medical images based on COVID-19 detection studies," 2020, *arXiv:2012.08333*. [Online]. Available: http://arxiv.org/abs/2012.08333
[181] S. N. Payrovnaziri, Z. Chen, P. Rengifo-Moreno, T. Miller, J. Bian, J. H. Chen, X. Liu, and Z. He, "Explainable artificial intelligence models using real-world electronic health record data: A systematic scoping review," *J. Amer. Med. Inform. Assoc.*, vol. 27, no. 7, pp. 1173–1185, Jul. 2020.
[182] L. M. Demajo, V. Vella, and A. Dingli, "Explainable AI for interpretable credit scoring," 2020, *arXiv:2012.03749*. [Online]. Available: http://arxiv.org/abs/2012.03749
[183] D. Cirqueira, D. Nedbal, M. Helfert, and M. Bezbradica, "Scenario-based requirements elicitation for user-centric explainable AI," in *Proc. Int. Cross-Domain Conf. Mach. Learn. Knowl. Extraction (CD-MAKE)*, 2020, pp. 321–341.
[184] R. Walambe, A. Kolhatkar, M. Ojha, A. Kademani, M. Pandya, S. Kathote, and K. Kotecha, "Integration of explainable AI and blockchain for secure storage of human readable justifications for credit risk assessment," in *Proc. Int. Adv. Comput. Conf. (IACC)*, 2021, pp. 55–72.
[185] M. Nassar, K. Salah, M. H. U. Rehman, and D. Svetinovic, "Blockchain for explainable and trustworthy artificial intelligence," *WIREs Data Mining Knowl. Discovery*, vol. 10, no. 1, Jan. 2020, Art. no. e1340, doi: 10.1002/widm.1340.
[186] J. V. D. Burgt, "Explainable AI in banking," *J. Digit. Banking*, vol. 4, no. 4, pp. 344–350, 2020.
[187] O. Loyola-González, "Understanding the criminal behavior in Mexico City through an explainable artificial intelligence model," in *Proc. Mexican Int. Conf. Artif. Intell.*, 2019, pp. 136–149.
[188] H. Zhong, Y. Wang, C. Tu, T. Zhang, Z. Liu, and M. Sun, "Iteratively questioning and answering for interpretable legal judgment prediction," in *Proc. 34th AAAI Conf. Artif. Intell. (AAAI)*, 2020, pp. 1250–1257.
[189] K. Atkinson, T. Bench-Capon, and D. Bollegala, "Explanation in AI and law: Past, present and future," *Artif. Intell.*, vol. 289, Dec. 2020, Art. no. 103387.
[190] A. Deeks, "The judicial demand for explainable artificial intelligence," *Columbia Law Rev.*, vol. 119, no. 7, pp. 1829–1850, 2019.
[191] C. Yang, M. Chen, and Q. Yuan, "The application of XGBoost and SHAP to examining the factors in freight truck-related crashes: An exploratory analysis," *Accident Anal. Prevention*, vol. 158, Aug. 2021, Art. no. 106153.
[192] A. Sargsyan, A. Karapetyan, W. L. Woon, and A. Alshamsi, "Explainable AI as a social microscope: A case study on academic performance," in *Proc. 6th Int. Conf. Mach. Learn., Optim., Data Sci. (LOD)*, pp. 257–268, 2020.
[193] S. Khedkar, P. Gandhi, G. Shinde, and V. Subramanian, "Deep learning and explainable AI in healthcare using EHR," in *Deep Learning Techniques for Biomedical and Health Informatics*. Springer, 2020, pp. 129–148.
[194] D. Dave, H. Naik, S. Singhal, and P. Patel, "Explainable AI meets healthcare: A study on heart disease dataset," 2020, *arXiv:2011.03195*. [Online]. Available: http://arxiv.org/abs/2011.03195
[195] L. M. Thimoteo, M. M. Vellasco, J. M. do Amaral, K. Figueiredo, C. L. Yokoyama, and E. Marques, "Interpretable machine learning for COVID-19 diagnosis through clinical variables," *Anais da Sociedade Brasileira de Automática*, vol. 2, no. 1, 2020. [Online]. Available: https://www.sba.org.br/open_journal_systems/index.php/sba/article/view/1590, doi: 10.48011/asba.v2i1.1590.
[196] C.-A. Hu, C.-M. Chen, Y.-C. Fang, S.-J. Liang, H.-C. Wang, W.-F. Fang, C.-C. Sheu, W.-C. Perng, K.-Y. Yang, K.-C. Kao, C.-L. Wu, C.-S. Tsai, M.-Y. Lin, and W.-C. Chao, "Using a machine learning approach to predict mortality in critically ill influenza patients: A cross-sectional retrospective multicentre study in Taiwan," *BMJ Open*, vol. 10, no. 2, Feb. 2020, Art. no. e033898.
[197] A. Adadi and M. Berrada, "Explainable AI for healthcare: From black box to interpretable models," in *Embedded Systems and Artificial Intelligence*. Singapore: Springer, 2020, pp. 327–337.
[198] U. Pawar, D. O'Shea, S. Rea, and R. O'Reilly, "Explainable AI in healthcare," in *Proc. Int. Conf. Cyber Situational Awareness, Data Anal. Assessment (CyberSA)*, Jun. 2020, pp. 1–2.
[199] M. A. Ahmad, A. Teredesai, and C. Eckert, "Interpretable machine learning in healthcare," in *Proc. IEEE Int. Conf. Healthcare Informat. (ICHI)*, Jun. 2018, pp. 559–560.
[200] S. Sachan, J.-B. Yang, D.-L. Xu, D. E. Benavides, and Y. Li, "An explainable AI decision-support-system to automate loan underwriting," *Expert Syst. Appl.*, vol. 144, Apr. 2020, Art. no. 113100.
[201] J. J. Ohana, S. Ohana, E. Benhamou, D. Saltiel, and B. Guez, "Explainable AI (XAI) models applied to the multi-agent environment of financial markets," in *Proc. 3rd Int. Workshop Explainable, Transparent Auton. Agents Multi-Agent Syst. (EXTRAAMAS)*, in Lecture Notes in Computer Science, vol. 12688, D. Calvaresi, A. Najjar, M. Winikoff, and K. Främling, Eds. Cham, Switzerland: Springer, May 2021, pp. 189–207.
[202] H. Yoon, "A machine learning evaluation of the COMPAS dataset," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, p. 5474.
[203] A. Kuppa and N.-A. Le-Khac, "Black box attacks on explainable artificial Intelligence(XAI) methods in cyber security," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.
[204] S. Wachter, B. Mittelstadt, and C. Russell, "Why fairness cannot be automated : Bridging the gap between EU non-discrimination law and AI," *Comput. Law Secur. Rev.*, vol. 41, 2021. Art. no. 105567.
[205] K. O'Hara, "Explainable AI and the philosophy and practice of explanation," *Comput. Law Secur. Rev.*, vol. 39, Nov. 2020, Art. no. 105474.
[206] *The Impact of the General Data Protection Regulation (GDPR) on Artificial Intelligence*. Accessed: Jul. 7, 2021. [Online]. Available: https://www.europarl.europa.eu/RegData/etudes/STUD/2020/641530/EPRS_STU%(2020)641530_EN.pdf
[207] D. Wang, Q. Yang, A. Abdul, and B. Y. Lim, "Designing theory-driven user-centric explainable AI," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, May 2019, pp. 1–15.
[208] S. Coppers, J. Van den Bergh, K. Luyten, K. Coninx, I. van der Lek-Ciudin, T. Vanallemeersch, and V. Vandeghinste, "Intellingo: An intelligible translation environment," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, Apr. 2018, pp. 1–13.

[209] Q. V. Liao, D. Gruen, and S. Miller, "Questioning the AI: Informing design practices for explainable AI user experiences," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, Apr. 2020, pp. 1–15.
[210] J. Rebanal, J. Combitsis, Y. Tang, and X. Chen, "XAlgo: A design probe of explaining Algorithms' internal states via question-answering," in *Proc. 26th Int. Conf. Intell. User Interface*, Apr. 2021, pp. 329–339.
[211] A. M. Antoniadi, Y. Du, Y. Guendouz, L. Wei, C. Mazo, B. A. Becker, and C. Mooney, "Current challenges and future opportunities for XAI in machine learning-based clinical decision support systems: A systematic review," *Appl. Sci.*, vol. 11, no. 11, p. 5088, May 2021.
[212] S. Bruckert, B. Finzel, and U. Schmid, "The next generation of medical decision support: A roadmap toward transparent expert companions," *Frontiers Artif. Intell.*, vol. 3, Sep. 2020, Art. no. 507973.
[213] A. Holzinger, A. Carrington, and H. Müller, "Measuring the quality of explanations: The system causability scale (SCS)," *KI-Künstliche Intelligenz*, vol. 34, pp. 193–198, Jan. 2020.
[214] A. Holzinger, "Explainable AI and multi-modal causability in medicine," *i-com*, vol. 19, no. 3, pp. 171–179, Jan. 2021.
[215] S. I. Lee and S. J. Yoo, "Multimodal deep learning for finance: Integrating and forecasting international stock markets," *J. Supercomput.*, vol. 76, no. 10, pp. 8294–8312, Oct. 2020.
[216] A. Peña, I. Serna, A. Morales, and J. Fierrez, "FairCVtest demo: Understanding bias in multimodal learning with a testbed in fair automatic recruitment," in *Proc. Int. Conf. Multimodal Interact.*, Oct. 2020, pp. 760–761.
[217] G. Joshi, R. Walambe, and K. Kotecha, "A review on explainability in multimodal deep neural nets," *IEEE Access*, vol. 9, pp. 59800–59821, 2021.
[218] D. H. Park, L. A. Hendricks, Z. Akata, A. Rohrbach, B. Schiele, T. Darrell, and M. Rohrbach, "Multimodal explanations: Justifying decisions and pointing to the evidence," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8779–8788.
[219] R. Srinivasan, A. Chander, and P. Pezeshkpour, "Generating user-friendly explanations for loan denials using GANs," 2019, *arXiv:1906.10244*. [Online]. Available: http://arxiv.org/abs/1906.10244
[220] H. Lee, S. T. Kim, and Y. M. Ro, "Generation of multimodal justification using visual word constraint model for explainable computer-aided diagnosis," in *Interpretability of Machine Intelligence in Medical Image Computing and Multimodal Learning for Clinical Decision Support*. Shenzhen, China: Springer, 2019, pp. 21–29.
[221] K. Alipour, A. Ray, X. Lin, J. P. Schulze, Y. Yao, and G. T. Burachas, "The impact of explanations on AI competency prediction in VQA," in *Proc. IEEE Int. Conf. Humanized Comput. Commun. with Artif. Intell. (HCCAI)*, Sep. 2020, pp. 25–32.
[222] J. Wu, L. Chen, and R. J. Mooney, "Improving VQA and its explanations by comparing competing explanations," 2020, *arXiv:2006.15631*. [Online]. Available: http://arxiv.org/abs/2006.15631
[223] D. Gunning and D. Aha, "DARPA's explainable artificial intelligence (XAI) program," *AI Mag.*, vol. 40, no. 2, pp. 44–58, Jun. 2019.
[224] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, "XAI—Explainable artificial intelligence," *Sci. Robot.*, vol. 4, no. 37, 2019, Art. no. eaay7120.
[225] *Explainable Machine Learning Challenge*. Accessed: Jul. 6, 2021. [Online]. Available: https://community.fico.com/s/explainable-machine-learning-challenge
[226] C. Chen, K. Lin, C. Rudin, Y. Shaposhnik, S. Wang, and T. Wang, "An interpretable model with globally consistent explanations for credit risk," 2018, *arXiv:1811.12615*. [Online]. Available: http://arxiv.org/abs/1811.12615
[227] *Model Interpretability in Azure Machine Learning*. Accessed: Jul. 6, 2021. [Online]. Available: https://docs.microsoft.com/en-us/azure/machine-learning/how-to-machine-%learning-interpretability
[228] *Providing Explainable AI Through Advanced Knowledge Graphs*. Accessed: Jul. 18, 2021. [Online]. Available: https://kyndi.com/wp-content/uploads/2018/03/Cognilytica-Briefing-Note_%Kyndi.pdf

MARIA SAHAKYAN received the Ph.D. degree in interdisciplinary engineering from Khalifa University, United Arab Emirates, in 2020, where her research was focused on explainable artificial intelligence. She is currently a Postdoctoral Associate at New York University Abu Dhabi, United Arab Emirates. Her current research interests include explainable artificial intelligence, machine learning, and data analytics

ZEYAR AUNG (Senior Member, IEEE) received the Ph.D. degree in computer science from the National University of Singapore, in 2006. From 2006 to 2010, he worked as a Research Fellow at the Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore. In 2010, he joined Masdar Institute, which later became a part of Khalifa University, as an Assistant Professor. He is currently an Associate Professor with the Department of Electrical Engineering and Computer Science, Khalifa University. His past research interests include bioinformatics and chemoinformatics. His current research interests include data analytics, machine learning, and their applications in various domains like cyber security, social media, renewable energy, and power systems.

TALAL RAHWAN received the Ph.D. degree in computer science from the University of Southampton, U.K., in 2007. He is currently an Associate Professor of computer science at New York University Abu Dhabi, United Arab Emirates. His research interests include data science, computational social science, game theory, and artificial intelligence. He received the Dean's Award for Early Career Researcher from the University of Southampton. His Ph.D. thesis earned the British Computer Society's Distinguished Dissertation Award, which annually recognizes the most outstanding Ph.D. thesis in computer science, U.K. He was selected by the IEEE Computer Society as one of the ten most promising, young artificial intelligence (AI) researchers in the world. His work appeared in major academic journals, including *Nature Communications*, *Nature Human Behaviour*, and *Nature Machine Intelligence*.