EXPLAN: Explaining Black-box Classifiers using Adaptive Neighborhood Generation

1st Peyman Rasouli
Department of Informatics
University of Oslo
Oslo, Norway
peymanra@ifi.uio.no

2nd Ingrid Chieh Yu
Department of Informatics
University of Oslo
Oslo, Norway
ingridcy@ifi.uio.no

***Abstract*—Defining a representative locality is an urgent challenge in perturbation-based explanation methods, which influences the fidelity and soundness of explanations. We address this issue by proposing a robust and intuitive approach for *EXPL*aining black-box classifiers using *A*daptive *N*eighborhood generation (EXPLAN). EXPLAN is a module-based algorithm consisted of dense data generation, representative data selection, data balancing, and rule-based interpretable model. It takes into account the adjacency information derived from the black-box decision function and the structure of the data for creating a representative neighborhood for the instance being explained. As a local model-agnostic explanation method, EXPLAN generates explanations in the form of logical rules that are highly interpretable and well-suited for qualitative analysis of the model's behavior. We discuss fidelity-interpretability trade-offs and demonstrate the performance of the proposed algorithm by a comprehensive comparison with state-of-the-art explanation methods LIME, LORE, and Anchor. The conducted experiments on real-world data sets show our method achieves solid empirical results in terms of fidelity, precision, and stability of explanations.**

***Index Terms*—XAI, Interpretable Machine Learning, Perturbation-based Explanation Methods, Data Sampling**

I. Introduction

Big data has led to the emergence of sophisticated Machine Learning (ML) models that are widely used in industrial, research, and personal applications [23], [38]. Their applications range from safety-critical systems such as self-driving cars and health aid software to personalized systems including movie recommendation systems and smart home appliances. Deep Neural Networks (DNN) [38] and Random Forests (RF) [6] are prominent examples of sophisticated ML models that are highly accurate for many applications, however, their complexity limits their interpretability, and hence they are treated as black-boxes [9].

A black-box model hides its internal behavior and reasons behind its decisions to the user. This lack of transparency is problematic in the sense of both applicability and ethics. For the former, there is a degradation of their applicability due to the need for explainability and understanding the logic of the model as well as the need for performing robustness analysis of the model [3]. For the latter, as many data-driven black-box models are created based on human-generated data, the resultant model is likely to inherit human biases and prejudices, unconsciously making unfair and incorrect decisions [14]. The importance of decision transparency is further emphasized by the GDPR regulation in the European Union that states individuals have the right to receive *meaningful information about the logic involved* when automated decision-making takes place [3]. Opening black-box models using a faithful explanation method is considered as an effective approach to elucidate and address these concerns. In addition to the quantitative analysis of ML models using standard metrics like accuracy, interpretability can assist in qualitative analysis such as fairness, privacy, and reliability of the models.

Explanation methods interpret black-box ML algorithms such as Support Vector Machine (SVM), Neural Network (NN), and Random Forests (RF) [6], [7], [37]. These approaches imitate the behavior of a model locally or globally, to provide explanations for a specific decision or to reveal the overall behavior of the model, respectively. *Fidelity* is an essential criterion in explanation methods. By fidelity, we mean to which extent the interpretable model is able to accurately imitate a black-box prediction and is often measured in terms of the F1-score and Mean Squared Error (MSE) [15]. The *local fidelity* concerns the behavior of the model in the vicinity of a specific instance being explained. Whereas, the *global fidelity* is about the overall behavior of the model (given all inputs). It is worth noting that global fidelity would imply local fidelity, however, identifying globally faithful explanations remains a challenge for complex models. For an explanation to be meaningful, it must at least be locally faithful [25], despite that local fidelity does not imply global fidelity. The main concern in creating an explanation method is to establish a trade-off between *fidelity* and *interpretability*. To increase the dependability of explanations, an explanation method, therefore, must be accurate enough to avoid generating explanations based on incorrect predictions. There is a wide range of research works that aim to address the described issue [3], [12], [14], [33].

In the perturbation-based local explanation methods, the locality created based on a data sampling procedure influences the degree of local fidelity. There are different strategies for creating perturbed neighborhood samples in the proximity of an instance of interest, such as using the distribution of training data, a Gaussian distribution, or evolutionary algorithms. Although these techniques are widely used in many state-of-the-art solutions, there are general challenges associated

with these techniques. Using the distribution of training data merely for creating neighborhood samples may fail to define a compact locality for the instance of interest. This issue can be even severe if imbalanced data sets being used. Generating perturbed samples using a Gaussian distribution is problematic as it disregards the interaction between features and may produce unlikely data points (i.e., outliers). Although evolutionary algorithms, such as the Genetic algorithm, may guarantee to produce a compact neighborhood, they may neglect the diversity of samples in favor of minimizing the cost function (or maximizing the fitness function) [19].

In our opinion, several factors should be considered during the neighborhood generation. First, the created data points need to be similar to the real data that is used for training or testing the black-box model. Second, the neighborhood should contain equal proportion of samples per class to guarantee an unbiased local interpretable model. Third, the diversity of samples should be kept as it has an important role in creating a reliable interpretable model and its produced explanations. Finally, to obtain precise and insightful explanations, it is necessary to have compact data around the instance of interest.

In this paper, we introduce EXPLAN, a rule-based method for black-box outcome explanation problem. It uses a combination of supervised and unsupervised learning for defining the neighborhood of a given instance in an adaptive manner. The main intuition behind EXPLAN is to create neighborhood data by looking at locality from two different but complementary viewpoints, i.e., the decision function of the black-box model and the structure/distribution of the data. As each view provides a different insight into the relationship of data points, we leverage them to generate a more expressive and informative data set. Overall, EXPLAN considers class balance, compactness, and diversity aspects without compromising the computational cost and implementation overhead.

EXPLAN starts by generating data points in close proximity of the instance to be explained using the proposed method in our previous study [24]. To find representative data points per class, an adaptive procedure based on Agglomerative Clustering [20] is devised. Then, SMOTE [8], an effective oversampling technique, is adopted to resolve the class imbalance problem. The result of this workflow is a compact, balanced neighborhood that is used as training data for constructing an interpretable model. Here, we employ decision trees to explain the decision made by the black-box for the given instance. EXPLAN is a local model-agnostic explanation method applicable to tabular data classification problems. We conducted experiments on several classification data sets and compared the results with state-of-the-art methods LIME [25], and rule-based explanation methods LORE [13] and Anchor [26]. The evaluations demonstrate significant results in terms of fidelity, precision, and stability of the generated explanations.

The rest of the paper is organized as follows. A concrete definition of the local explanation problem and rule-based explanations are provided in Section II. Related works are discussed in Section III. The proposed explanation method, EXPLAN, is introduced in Section IV. The experiments and achieved results are described in Section V. Finally, we conclude the paper and identify future work in Section VI.

II. The Local Explanation Problem

In this section, we recall the basic definitions of tabular data classification, black-box predictor, and interpretable predictor. Subsequently, we define the black-box outcome explanation problem and introduce the concept of local explanation for which the solution EXPLAN is proposed.

Classification, Black-box predictor, and Interpretable predictor. Classification predictive modeling is the task of approximating a mapping function $f : \mathcal{X}^m \rightarrow \mathcal{Y}$, where $\mathcal{X}^m$ is a set of input instances consisting of m features and $\mathcal{Y}$ is the target set. The features m can correspond to any basic data type like integers, reals, booleans, and strings. On the other hand, $\mathcal{Y}$ contains different labels (classes or outcomes) for each input which determines a semantic concept; it can be a set of booleans, integers, or strings. Given an instance $x \in \mathcal{X}^m$, the predictor f can be employed to predict the target value y, i.e., $f(x) = y$. We treat f as a black-box predictor where its internal behavior is either opaque or known but uninterpretable. Examples of such black-box predictors include neural networks, random forest, and support vector machines. Similarly, we indicate with $\mathcal{C}$ an interpretable predictor whose internal reasoning that yields a decision/prediction can be represented by a symbolic representation. Examples of interpretable predictors include decision trees, rule-based classifiers, and rational functions that can provide explanations in the form of logical rules, which is more accessible to humans (however, the study of cognitive comprehensibility of explanations is outside the scope of this paper).

Black Box Outcome Explanation. Given a black-box predictor f and an input x, the black-box outcome explanation problem is about providing an explanation for the outcome $f(x) = y$. We address this problem by creating an interpretable predictor $\mathcal{C}$, which returns the prediction of $\mathcal{C}(x) = y$ together with the reasons behind the prediction as an explanation. In other words, the predictor $\mathcal{C}$ mimes the *local* behavior of f for the particular instance x, without aiming to explain the logic of the black-box globally. Assuming the availability of some knowledge about the characteristics of the feature space, the locality of x is generated as part of the explanation process. Let $\mathcal{C} = \psi(f, x)$ be an interpretable predictor derived from applying the explanation function ψ on the black-box f and the instance x. An explanation belonging to an interpretable domain E, i.e., $e \in E$, is obtained from $\mathcal{C}$ by an explanation logic ε such that the explanation $e = \varepsilon(\mathcal{C}, x)$ can be extracted.

Local Explanation. In a decision rule $r = p \rightarrow y$, the decision y is the consequence of the rule, while the premise p is a conjunction of boolean conditions on feature values, i.e., $p_1 \wedge p_2 \wedge \cdots \wedge p_n$. An instance x satisfies r, or r covers x, if the boolean conditions in p evaluate to true for x. We define an explanation $e = p \rightarrow y$, where $e \in E$, as a decision rule describing the reasons for the prediction $\mathcal{C}(x) = y$. Let x be an instance we want to explain its black-box prediction, if x

satisfies p, the rule $p \rightarrow y$ represents the motivation for the decision value, i.e., p locally explains why f returns y.

According to the above-mentioned definitions, a solution to the outcome explanation problem will then consist of two general steps. First, defining the function ψ that creates an interpretable predictor $\mathcal{C}$ for a given black-box predictor f and an instance x. Second, defining the explanation logic ε to derive a local explanation from $\mathcal{C}$ and x.

III. Related Works

In recent years, several explanation methods and strategies have been proposed in the quest to make interpretable Machine Learning (ML). Generally, the introduced methods fall into two main categories: *intrinsic* interpretable models and *post-hoc* explanation methods. The intrinsic interpretable models are ML models that are inherently and intrinsically interpretable, Falling Rule Lists [35] and Bayesian Rule Lists [17] are popular models related to this research domain. A barrier to their adoption is the accuracy-interpretability trade-off as high accuracy is generally achieved by means of complex prediction models. An alternative to achieve interpretability in ML is to create complex black-box models that have high accuracy and subsequently using a post-hoc technique to provide the required explanations. This class of methods makes the ML models interpretable without altering or even knowing the internal behavior of the original black-box model. The scope of interpretability distinguishes the post-hoc explanation methods as *global methods* and *local methods*. The global explanation methods enable understanding the whole logic of the model while the local explanation methods explain the reasons behind a single prediction. In the following, we provide an overview of related global and local explanation methods.

Surrogate Models. A surrogate model is a simple, interpretable ML model which is used to approximate the predictions of a complex ML model and allow us to draw conclusions about the model's behavior. In other word, it addresses the machine learning interpretability by means of other machine learning models. Osbert et al. [4] proposed Model Extraction as a surrogate model based on decision trees to explain a black-box ML model globally. The authors employed active learning to actively sample a large number of training data points to avoid over-fitting in the learning process. TreeView [32] is a visualization technique for explaining complex deep neural networks using a surrogate model. Specifically, it understands the learned features by a DNN through extracting meta-features, which are used in a decision tree to predict the label of an input and the sequence of nodes visited in the tree during the decision making process.

Model Distillation. Distillation is a model compression technique to transfer information from a complex, black-box model (teacher model) to a simple, transparent model (student model) without significant loss in the prediction accuracy. In [30] a transparent model distillation approach is proposed to detect bias in black-box scoring models. Considering the black-box risk score as the teacher model, interpretable generalized additive models (GAMs) [36] are used as student models. By comparing two GAMs created on the risk score and the actual outcome, it reveals feature values with potential biases in the teacher model. Distill-and-Compare [31] is a method for realistic conditions to gain insight into black-box teacher models via transparent student models that are trained on audit data (i.e., the data that is labeled by the scoring model). Authors in [29] proposed a model distillation technique to learn global additive explanations for interpreting the neural networks trained on tabular data. Their framework visualizes the existing trends in feature space, which allows identifying the important features, analyzing the training data, and debugging errors learned by the black-box model.

Feature Importance. Quantifying the contribution of each input feature to the outcome of a black-box model is a popular explanation mechanism. In this way, we discover the reasons for a specific prediction by observing the importance degree of each feature. Ribeiro et al. presented LIME [25], as a local model-agnostic explanation method that explains a given instance by creating an interpretable model (a linear regression) based on the neighborhood of the instance. LIME determines the locality via a kernel function, defined on the distance of randomly generated data points to the instance of interest. For a specific instance, the explanation is derived in the form of feature importance where the number of desired features is determined by a hyper-parameter. SHAP is an explanatory approach based on coalitional game theory [18]. The authors use Shapley value as the average contribution of a feature value to a prediction in different coalitions. In SHAP, an explanation is represented as an additive feature attribution method (a linear regression) that is straightforward to extract the importance of each input feature. Eliana et al. proposed LACE [21], a local model-agnostic explanation method. The locality of a sample is formed using the K-nearest neighbor algorithm on the training data. Through a rule-based classifier created on the defined neighborhood, LACE uses a quantity called prediction difference to identify the contribution of features to the prediction of the instance.

Rule-based Methods. These type of techniques gain insight into a black-box model through decision rules. A decision rule consists of a single or several IF-THEN statements that is used for making a prediction [14]. Anchors is a local rule-based explanation method based on reinforcement learning and graph search techniques [26]. It explains a specific instance using a decision rule that "anchors" the black-box prediction. A rule containing some predicates/features anchors a prediction if changes in the value of other features do not influence the prediction. Anchor provides coverage and precision as supplementary information for each explanation. Riccardo et al. proposed LORE [13] to explain the outcome of any black-box model under the tabular data classification setting. LORE assumes a higher availability of clear and simple decision boundary in the neighbourhood of a data point rather than the whole feature space. Therefore, it creates a balanced, compact locality for a given sample using an ad-hoc genetic algorithm. LORE provides decision rules and counter-factual rules to explain an instance of interest. In [16], authors introduced

BETA, a global explanation method based on a multi-objective optimization framework. In this work, unambiguity, fidelity, and interpretability are used as optimization goals. An explanation is generated in the form of several decision sets (sets of IF-THEN rules), each of which captures the behavior of a black-box model in certain parts of the feature space.

IV. EXPLAN Explanation Method

Creating a sound locality for the instance to be explained is a prerequisite for having a faithful local explanation. In this section, we introduce EXPLAN that consists of an adaptive neighborhood generation pipeline to derive a representative locality for the instance being explained. Utilizing a decision tree as the interpretable model, explanations are extracted in the form of decision rules. The algorithm of EXPLAN is described in Algorithm 1 that consists of four main procedures explained in the following sections.

Algorithm 1 EXPLAN Explanation Method

Input: $\{x, f, \mathcal{D}, \mathcal{N}, \tau\}$
/ x: instance to explain, f: black-box model, $\mathcal{D}$: distribution of training data, $\mathcal{N}$: # initial neighborhood samples, τ: # minimum samples per class */*
Output: $\{\mathcal{C}, e\}$
/ $\mathcal{C}$: interpretable model, e: explanation of x */*

1: **function** EXPLAN($x, f, \mathcal{D}, \mathcal{N}, \tau$)
2: $\quad\mathcal{Z} \leftarrow$ DATAGENERATION($x, f, \mathcal{D}, \mathcal{N}$)
3: $\quad\mathcal{Z}' \leftarrow$ DATASELECTION($x, f, \mathcal{Z}, \tau$)
4: $\quad\mathcal{X} \leftarrow$ DATABALANCING($f, \mathcal{Z}'$)
5: $\quad\mathcal{C}, e \leftarrow$ INTERPRETABLEMODEL($x, f, \mathcal{X}$)
6: $\quad$**return** $\mathcal{C}, e$

A. Dense Data Generation

As we are interested to explain x locally, there must be adequate samples in the proximity of x in order to generate reliable explanations. Although it may not impact the fidelity of the interpretable model $\mathcal{C}$ if we use distant samples, the provided explanations may not be a representation of the neighborhood of x. To achieve a dense locality, we employ the data sampling technique introduced in our previous study [24] that generates a compact neighborhood for an instance of interest. This technique is summarized in Algorithm 2.

Data generation starts with RANDOMDATAGENERATION phase that draws $\mathcal{N}$ perturbed samples from the distribution of training data $\mathcal{D}$, denoted by $\mathcal{S}$. Using the distribution of training data leads to create likely random data points that are similar to the original data set in terms of both feature values and class balance. In SURROGATEMODELCONSTRUCTION phase, a random forest $\mathcal{T}$ using $(\mathcal{S}, f(\mathcal{S}))$ as the training data is created to mime f globally. The reason is to leverage the created surrogate model with *TreeInterpreter* technique [28] to achieve observation-level feature importance in the subsequent CONTRIBUTIONEXTRACTION phase. Let $\mathcal{L}$ be the set of labels of the data. Given a trained random forest $\mathcal{T}$ and a sample $s, s \in \mathcal{S}$, TreeInterpreter decomposes each prediction $\mathcal{T}_l(s), l \in \mathcal{L}$ into a bias value and a vector containing the contribution of each feature in the prediction $\mathcal{T}_l(s)$. The output of CONTRIBUTIONEXTRACTION procedure is the aggregation of the achieved contribution vectors, denoted by $\mathcal{V}$. Finally, the randomly generated samples in S are made closer to the instance of interest x through SAMPLEMANIPULATION phase, as features in $s, s \in \mathcal{S}$, that have different values than features in x, but expressing similar contribution to the mutual target classes ($\mathcal{T}_l(s)$ and $\mathcal{T}_l(x)$), are flipped to the feature values of x. The discrete versions of $\mathcal{S}$ and $\mathcal{V}$, obtained using a Quantile-based discretization method [10], are used for the comparison.

Algorithm 2 Dense Data Generation

procedure DATAGENERATION($x, f, \mathcal{D}, \mathcal{N}$)
  procedure RANDOMDATAGENERATION($\mathcal{D}, \mathcal{N}$)
    $\mathcal{S} \leftarrow$ *DataSampling*($\mathcal{D}, \mathcal{N}$)
    return $\mathcal{S}$ */* random data points */*
  procedure SURROGATEMODELCONSTRUCTION($f, \mathcal{S}$)
    $\mathcal{T} \leftarrow$ *RandomForestConstructor*($\mathcal{S}, f(\mathcal{S})$)
    return $\mathcal{T}$ */* RF surrogate model */*
  procedure CONTRIBUTIONEXTRACTION($x, \mathcal{S}, \mathcal{T}$)
    $\mathcal{V}(x) \leftarrow$ *TreeInterpreter*($\mathcal{T}, x$)
    for all $s \in \mathcal{S}$ **do**
      $\mathcal{V}(s) \leftarrow$ *TreeInterpreter*($\mathcal{T}, s$)
    return $\mathcal{V}$ */* feature importance */*
  procedure SAMPLEMANIPULATION($x, \mathcal{S}, \mathcal{T}, \mathcal{V}$)
    $l_x \leftarrow \mathcal{T}(x)$
    $\mathcal{Z} \leftarrow \{\}$
    for all $s \in \mathcal{S}$ **do**
      $l_s \leftarrow \mathcal{T}(s)$
      for $j \leftarrow 1, \mathcal{F}$ **do** */* $\mathcal{F}$: feature dimension */*
        if $(s_j \neq x_j)$ **then**
          if $(\mathcal{V}^{l_x}_{s_j} = \mathcal{V}^{l_x}_{x_j}) \wedge (\mathcal{V}^{l_s}_{s_j} = \mathcal{V}^{l_s}_{x_j})$ **then**
            $s_j \leftarrow x_j$
      $\mathcal{Z} \leftarrow \mathcal{Z} \cup s$
    return $\mathcal{Z}$ */* meaningful dense data w.r.t x */*
  return $\mathcal{Z}$

Without affecting the class balance of samples, the procedure transforms the randomly generated data into a compact data that contains samples close to the instance of interest. Compared to nearest neighbor search techniques, which find adjacent samples for a data point [1], this method works on feature values to make a dense data in the vicinity of the instance being explained. The output of DATAGENERATION is $\mathcal{Z}$, a meaningful dense data w.r.t. to x.

B. Representative Data Selection

The objective of DATAGENERATION, is to generate dense samples w.r.t. the sample being explained. In this step (i.e., DATASELECTION), our goal is to select representative data points from the pool using unsupervised learning that provides useful information about the structure of the data and the distribution of samples. Given a specified threshold for the

minimum number of samples, the devised procedure adaptively selects appropriate data points for x, which accordingly determines the density of its neighborhood.

It is noteworthy that the nearest neighbor methods (e.g., KNN) [1] require specifying an exact and equal number of adjacent samples for every instance being explained. Nevertheless, it is uncertain whether the specified number of samples would be appropriate (i.e., sufficient and representative) for any particular instance. In contrast, our selection procedure adaptively selects a representative number of samples per class w.r.t an instance of interest. The proposed procedure is summarized in Algorithm 3.

Algorithm 3 Representative Data Selection

procedure DATASELECTION($x, f, \mathcal{Z}, \tau$)
  $n_c \leftarrow 2$ /* n_c: number of clusters */
  $\mathcal{Z}' \leftarrow \{\}$
  for all $l \in \mathcal{L}$ **do** /* $\mathcal{L}$: set of labels */
    $\mathcal{G}_l \leftarrow \{z \in Z \mid f(z) = l\}$
    $\mathcal{G}_l \leftarrow x \cup \mathcal{G}_l$
    while `True` **do**
      $c_x, c_{\neg x} \leftarrow$ *AgglomerativeClustering*$(\mathcal{G}_l, n_c)$
      if $|c_x| \geq \tau$ **then**
        $\mathcal{G}_l \leftarrow c_x$
      else
        break
    $\mathcal{Z}' \leftarrow \mathcal{Z}' \cup \mathcal{G}_l$
  return $\mathcal{Z}'$ /* representative data set */

The procedure is as follows. Let $\mathcal{L}$ be the set of labels of the data. For each $l \in \mathcal{L}$, the samples in $\mathcal{Z}$ that are labeled with l (using the black-box f) constitute a sample set $\mathcal{G}_l$. Afterwards, x is added to each sample set, i.e., $\mathcal{G}_l = \{x \cup \mathcal{G}_l\}, l \in \mathcal{L}$. Given a minimum number of data points per class, i.e., τ, Agglomerative clustering is applied iteratively on each $\mathcal{G}_l, l \in \mathcal{L}$, until suitable clusters containing at least τ samples are achieved, i.e., $|\mathcal{G}_l| \geq \tau$. For the concrete implementation, we use Ward's linkage with the number of clusters set to $n_c = 2$, which is the default value for the algorithm [20], [22]. Specifically, in each iteration of the proposed procedure, a group $\mathcal{G}_l$ is divided into n_c clusters, and data points that are in the same cluster as x, denoted by c_x, are kept for the next iteration and $\mathcal{G}_l$ is updated. When the data is no longer divisible (i.e., $c_x = x \rightarrow |c_x| = 1$) or the threshold τ is exceeded, the procedure terminates and returns $\mathcal{G}_l$. Eventually, we obtain a representative sample set for class l in the proximity of x. The result of this step is the union of the achieved sample sets, i.e., $\mathcal{Z}' = \bigcup_{l \in \mathcal{L}} \mathcal{G}_l$.

C. Data Balancing

When the case is an imbalanced data set, generating samples based on its distribution will be problematic. It may result in a sparse neighborhood for any data point belonging to the minority class. Therefore, the chance of having a compact, balanced neighborhood for the instance of interest is reduced, which affects the fidelity of the explanation method. More severely, it leads to the generation of explanations that are not representative and provide the user with incorrect information.

To mitigate the class imbalance problem, we introduce DATABALANCING that generates new samples in $\mathcal{Z}'$ for the under-represented classes. According to the comprehensive analysis reported in [39], SMOTE [8] is an effective over-sampling algorithm compared to Random oversampling and ADASYN [39]. Without losing useful information, SOMTE over-samples the minority class by interpolating new synthetic samples. It also overcomes the over-fitting problem caused by random oversampling. Given a sample x_i, a new sample x_{new} is generated based on a point x_{zi} in its nearest neighborhood using the following equation:

$$x_{new} = x_i + \lambda \times (x_{zi} - x_i)$$

where λ is a random number in the range $[0, 1]$. This interpolation will create a sample between x_i and x_{zi}. In this phase, SMOTE is applied on $\mathcal{Z}'$ to balance the number of instances in each class which results in a new data set $\mathcal{X}$. This is the final data set that is used for training the local rule-based interpretable model.

D. Rule-based Interpretable Model

The final step of the explanation methodology is to build an interpretable classifier $\mathcal{C}$ trained on $\mathcal{X}$ labeled with the black-box decision function $f(\mathcal{X})$. Such a predictor is able to mime the behavior of the black box f within $\mathcal{X}$. Since $\mathcal{C}$ is an interpretable machine learning model, an explanation for the prediction $f(x)$ can be directly extracted. We employ YaDT implementation of the C4.5 decision tree induction algorithm to generate the interpretable model [27]. This technique is computationally cheap and decision rules can be derived from root-leaf paths in the constructed tree. An explanation $e = p \rightarrow y$ is a rule where p contains the split conditions from the root-leaf path that is satisfied by the instance x and $\mathcal{C}(x) = y$. By construction, the rule e is consistent with $\mathcal{C}$ and satisfied by x. The output of INTERPRETABLEMODEL is the local rule-based interpretable model $\mathcal{C}$ and the explanation e.

V. EXPERIMENTS AND RESULTS

In this section, EXPLAN is evaluated with respect to several classification data sets and black-box models. The main goal of this work is to devise a faithful and stable explanation method. By faithful, we mean the ability of the method to accurately imitate the black-box behavior and by stable we mean the ability of the method to be robust against the variation of data sampling. We benchmarked EXPLAN against state-of-the-art LIME, LORE, and Anchor explanation methods. The evaluation results are reported in three parts: (*i*) fidelity comparison, (*ii*) neighborhood analysis, and (*iii*) explanation comparison.

Experimental Setup. The proposed explanation method has been developed in Python programming language and the experiments were run on a system with Intel Core i7-8620HQ processor and 32GB of memory. We used `scikit-learn`

library for implementing the machine learning and data mining algorithms [22]. Source code for replicating our experiments is available at: https://github.com/peymanras/EXPLAN.

In the experiments, three tabular classification data sets including *Adult*, *German credit* [1], and *COMPAS*[2] were used. Details about each data set is described in Table I. Each data set was split into 80% *train set* and 20% *test set*. Half of the samples in the test set were used for evaluating the explanation methods. The number of initial data points and the minimum number of samples per class in EXPLAN were set to $\mathcal{N} = 3000$ and $\tau = 250$, respectively. The default hyper-parameter settings for LIME, LORE, and Anchor were used during the experiments. A Neural Network classifier (**NN**) [37], a Logistic Regression classifier (**LR**) [5], and a Gradient Boosting classifier (**GB**) [11] with the default hyper-parameters specified in the `scikit-learn` library were employed as black-box models.

TABLE I
DESCRIPTION OF THE DATA SETS.

Data set	# Instances	# Features	Class imbalance
Adult	49K	14	<=50K: 76% - >50K: 24%
German	1K	20	Good: 70% - Bad: 30%
COMPAS	7K	52	Medium-Low: 72% - High: 28%

A. Fidelity Comparison

In this section, we compare the fidelity of EXPLAN with state-of-the-art techniques LIME and LORE. Explanations of Anchor are by construction faithful [26], therefore we exclude Anchor for this evaluation. The following properties are used for evaluating the fidelity of the explanation methods in miming the local behavior of the black-boxes. The notations y and Y represent the ground-truth labels and $\hat{y}$ and $\hat{Y}$ are the predicted labels of an individual sample and the entire training data set, respectively:

- $\boldsymbol{fidelity_x}(\boldsymbol{y}, \boldsymbol{\hat{y}}) \in [0, 1]$. It compares the prediction of $\mathcal{C}$ and black box f on the instance of interest x using F1-score.
- $\boldsymbol{fidelity_{\mathcal{X}}}(\boldsymbol{Y}, \boldsymbol{\hat{Y}}) \in [0, 1]$. It compares the predictions of $\mathcal{C}$ and black box f on the training samples $\mathcal{X}$ using F1-score.

The $\boldsymbol{fidelity_x}$ measures the difference between the prediction of the interpretable and the black-box models for any instance x. This is the main metric for determining the faithfulness of an explanation method. The $\boldsymbol{fidelity_{\mathcal{X}}}$ describes how good the interpretable model $\mathcal{C}$ is at imitating the behavior of the black-box decision function in the locality $\mathcal{X}$. In other word, it indicates whether the explanations are derived from a faithful local interpretable model. The desired value for $\boldsymbol{fidelity_x}$ and $\boldsymbol{fidelity_{\mathcal{X}}}$ is 1. Tables II and III present the average and the standard deviation of the results.

[1]Data sets are available at: https://archive.ics.uci.edu/ml/datasets/

[2]Data set is available at: https://www.kaggle.com/danofer/compass

TABLE II
COMPARISON OF $\boldsymbol{fidelity_x}$ SCORES.

Data set	Black-box	EXPLAN	LIME	LORE
Adult	**GB**	0.994±.1	0.838±.4	0.980±.1
	LR	0.992±.1	0.940±.2	0.989±.1
	NN	0.992±.1	0.859±.3	0.977±.2
German	**GB**	1.000±.0	0.910±.3	0.950±.2
	LR	0.990±.1	0.940±.2	0.910±.3
	NN	1.000±.0	0.930±.3	0.990±.1
COMPAS	**GB**	1.000±.0	0.911±.3	0.999±.0
	LR	1.000±.0	0.925±.3	0.981±.1
	NN	0.999±.0	0.915±.3	0.986±.1

For LIME, each sample was explained using $K = \{2, .., 10\}$ features and the result of K with the highest performance was considered. According to Tables II and III, EXPLAN outperforms LIME for all data sets and black-box models concerning both $\boldsymbol{fidelity_x}$ and $\boldsymbol{fidelity_{\mathcal{X}}}$ measures. The main cause of low $\boldsymbol{fidelity_{\mathcal{X}}}$ is the class imbalance of the generated neighborhood for the interpretable model and we observe that LIME is in particular prone to this problem. In comparison with LORE, EXPLAN has a better $\boldsymbol{fidelity_x}$ performance. This efficacy is due to the generation of balanced, representative samples in the locality of x that lead to the rigorous separation of classes. As a result, the created interpretable model $\mathcal{C}$ demonstrates high classification accuracy for the instance x and its neighborhood samples $\mathcal{X}$. Regarding the $\boldsymbol{fidelity_{\mathcal{X}}}$, EXPLAN and LORE demonstrate comparable performance. It is worth noting that the results of EXPLAN have a low variation for the mentioned scores, hence there is more stability in its predictions.

TABLE III
COMPARISON OF $\boldsymbol{fidelity_{\mathcal{X}}}$ SCORES.

Data set	Black-box	EXPLAN	LIME	LORE
Adult	**GB**	0.971±.0	0.738±.0	0.996±.0
	LR	0.990±.0	0.793±.1	0.995±.0
	NN	0.980±.0	0.804±.0	0.993±.0
German	**GB**	0.942±.0	0.223±.1	0.979±.0
	LR	0.972±.0	0.179±.1	0.944±.2
	NN	0.981±.0	0.037±.1	0.987±.0
COMPAS	**GB**	0.984±.0	0.897±.0	0.982±.1
	LR	0.988±.0	0.919±.0	0.975±.1
	NN	0.988±.0	0.896±.0	0.974±.1

B. Neighborhood Analysis

In the neighborhood analysis, we investigate precision, coverage, sample variance, and describe some statistics related to the generated neighborhood of the explanation methods. We compare EXPLAN with LORE and Anchor because neighborhood generation is the key feature of these methods, and they all use decision rules as their explanation strategy. Anchor is a high precision explanation method that guarantees a desired level of precision defined by the user (the default precision threshold is 0.95) and defines successful interpretable explanation as the one that has both high coverage and high

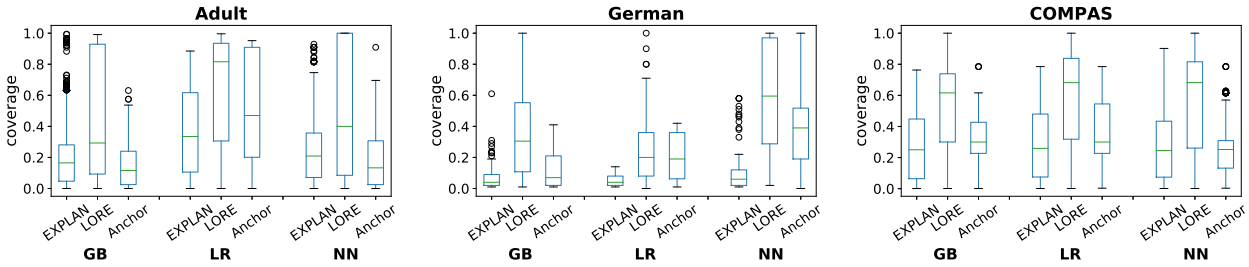

Fig. 1. Comparison of rule coverage.

TABLE IV
COMPARISON OF $\boldsymbol{precision_e}$ SCORES.

Data set	Black-box	EXPLAN	LORE	Anchor
Adult	**GB**	0.924±.1	0.852±.2	0.980±.1
	LR	0.966±.1	0.894±.2	0.963±.0
	NN	0.895±.2	0.815±.2	0.971±.1
German	**GB**	0.897±.2	0.816±.2	0.984±.0
	LR	0.937±.1	0.835±.3	0.994±.0
	NN	0.950±.1	0.879±.2	0.976±.1
COMPAS	**GB**	0.914±.2	0.855±.2	0.963±.0
	LR	0.912±.2	0.862±.2	0.963±.0
	NN	0.898±.2	0.851±.2	0.979±.0

precision [26]. By coverage we mean the *number* of data points that can be covered by the rule. The precision gives the *fraction* of data points in the neighborhood of x in the original data set that are covered and classified correctly by the rule. By benchmarking EXPLAN against Anchor, we will have a reliable comparison in terms of coverage and precision.

Fig. 1 depicts the coverage of the different explanation methods. Results state a comparable performance between EXPLAN and Anchor while a superior coverage for LORE. It is noted that the coverage of an explanation rule is influenced by the diversity of the neighborhood data. We will further demonstrate how the explanation methods perform from this perspective through feature frequency variance evaluation.

Precision indicates the ability of the explanation method in creating a representative locality for the instance being explained. The $\boldsymbol{precision_e}$ measures the accuracy of the generated rule e in classifying the neighborhood samples in the original data. The results of the $\boldsymbol{precision_e}$ score is given in Table IV. According to Table IV, EXPLAN outperforms LORE regarding precision viewpoint. An influential factor for the superiority of EXPLAN is the diversity of neighborhood samples that is inherited from the original data in the initial sample generation phase. Compared to Anchor, EXPLAN has lower but close precision values.

Fig. 2 illustrates a visualization of EXPLAN's neighborhood generation process for an instance from *Adult* data set using t-SNE technique [34]. Fig. 2 clearly shows that the locality in the initial step has been refined during sequential steps for the given sample. The representative data selection phase plays an important role in constraining the width of the neighborhood. It provides an appropriate, limited feature space for the balancing phase by removing the distant and unrelated data points. Finally, through the data balancing step, an equal number of samples per class is created that guarantees an unbiased interpretable model.

Feature frequency variance is a useful metric for measuring the diversity of a neighborhood data, which to the best of our knowledge we are the first paper that performs this analysis. Diversity affects fidelity, coverage, and precision of an explanation method. We calculate the variation between the frequency distribution of features to determine the diversity of a sampled data. A low variation refers to a diverse data set in which all features are equally distributed, whereas a high variation indicates a uniform data set in which one or a few unfair features are highly distributed. Having similar samples as the neighborhood data may lead to the deficiency of the interpretable model in capturing the local behavior of the black-box model. To calculate feature frequency variance, we use the *Coefficient of Variation* [2] which is the standard deviation relative to the mean. Compared to standard deviation which measures the variability for a single data set, the coefficient of variation allows us to compare the standard deviations of different data sets. Considering $\mathcal{CV}$ as the coefficient of variation function, we compute $\rho = \mathcal{CV}(\mathcal{CV}(\mathcal{X}))$, as it first measures the coefficient of variation of each feature in the data that results in a vector, then it calculates the coefficient of variation of the vector to determine the variation between the frequency distribution of features. A low ρ-value (i.e., $\rho \approxeq 0$) indicates a high diversity for all features in the data, whereas a high ρ-value indicates a high diversity for only one or a few specific features, which implicitly refers to similar samples in the locality. Neighborhood data generated by different explanation methods have varied sizes.To perform an unbiased measurement of ρ-value, an equal number of samples are randomly selected from the neighborhoods. Table V shows the mean and standard deviation of ρ-value for different scenarios with EXPLAN demonstrating a generally better performance than LORE and Anchor. It can be seen that LORE has a high ρ-value with considerable deviation values in all cases which shows its tendency for creating a locality just by changing a few specific features. While, EXPLAN and Anchor achieve similarly low ρ-values, reflecting the used samples for the neighborhood are highly diverse and representative.

The number of neighborhood samples for different methods varies considerably. For example, EXPLAN, on average, explains an instance using 1052 ± 281 samples, while LORE and Anchor need 1038 ± 25 and 3578 ± 1368 samples, respectively.

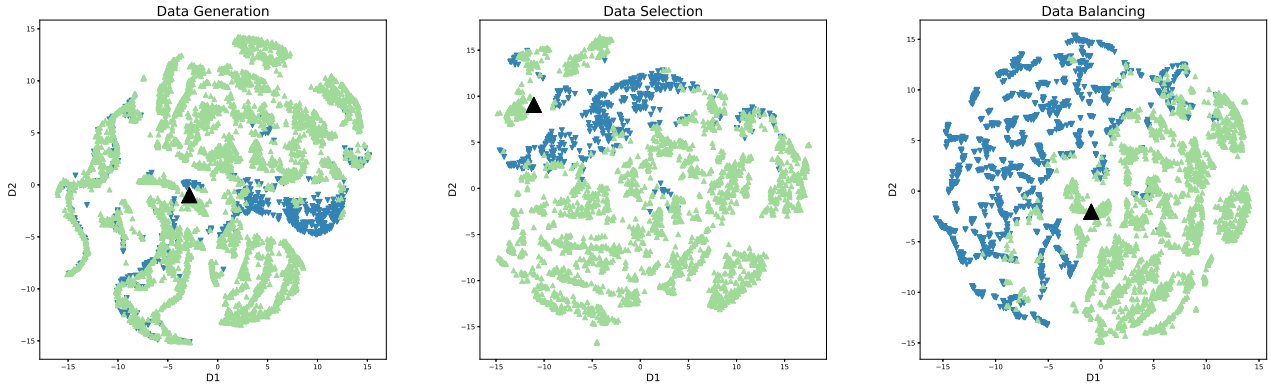

Fig. 2. Visualization of EXPLAN's neighborhood construction process.

TABLE V
COMPARISON OF FEATURE FREQUENCY VARIANCE.

Data set	Black-box	EXPLAN	LORE	Anchor
Adult	**GB**	1.191±.2	2.263±.5	1.497±.1
	LR	1.087±.2	2.259±.5	1.523±.1
	NN	1.174±.5	2.273±.5	1.509±.1
German	**GB**	0.469±.0	2.296±.6	0.472±.0
	LR	0.513±.0	2.293±.7	0.469±.0
	NN	0.467±.0	2.334±.6	0.468±.0
COMPAS	**GB**	0.529±.1	1.342±.4	0.682±.1
	LR	0.535±.1	1.335±.4	0.678±.1
	NN	0.533±.1	1.314±.4	0.681±.1

By measuring the distance between neighborhood points and the instance of interest we can approximately decide about the breadth of the locality. Although finding a sufficient width for the locality is a challenge, neighborhoods with remarkably small distances are not favored, as they indicate a hyper-local area in the decision boundary. The average neighborhood distance for EXPLAN, LORE, and Anchor are 5449609, 1930185, and 10129203, respectively. The class balance rate has a direct effect on the faithfulness of the interpretable model, and accordingly on the generated explanations. The average class balance rate for EXPLAN, LORE, and Anchor are 0.995 ± 0.0, 0.972 ± 0.0, and 0.659 ± 0.0, respectively.

C. Explanation Comparison

An explanation e generated by EXPLAN for an input x from *Adult* data set and **GB** black-box model is given below:

> $\boldsymbol{x} = \{$**age:** 30; **workclass:** Private; **education:** 11th; **marital-status:** Never-married; **occupation:** Prof-specialty; **relationship:** Unmarried; **race:** White; **sex:** Male; **capital-gain:** 0; **capital-loss:** 0; **hours-per-week:** 40; **native-country:** United-States — **class:** $\leq 50K\}$
>
> $\boldsymbol{e}$ = {**age:** $\leq$ 30 $\wedge$ **capital-gain:** $\leq$ 0 $\wedge$ **hours-per-week:** $\leq 44\}$ $\rightarrow$ **class:** $\leq 50K$

In this example, EXPLAN explains the instance using 3 features that are locally important for the instance to be classified as "$\leq 50K$". The aim of the provided explanation is merely to illustrate the structure of rule-based explanations. Comparing the validity of explanations is a qualitative analysis task that requires corresponding domain knowledge. We leave this part of experiments to future work.

Apart from the importance of fidelity, consistency in generating explanations is another desired property of explanation methods. By this, we mean how stable the method is in explaining a particular instance with the same explanation independent of the runs of the method. More specifically, for the input x given above, we want to derive the same explanation e in every run of EXPLAN. The Jaccard coefficient [15] is a way to calculate the similarity between explanations, and we used it to measure and compare the stability of EXPLAN, LORE, and Anchor. In this experiment, we conducted 5 runs for every instance and computed the similarity between the predicates of the rules in terms of Jaccard values which are reported in Table VI. The result shows that EXPLAN is comparable to Anchor, and it is significantly more stable than LORE.

The average time required by EXPLAN to explain an instance is 1.99 ± 0.5 seconds, while LORE and Anchor need 5.51 ± 1.2 and 0.45 ± 0.2 seconds, respectively. LIME explains an instance in 1.91 ± 0.4 seconds. According to the execution time, our proposed algorithm is computationally efficient. It explains an instance almost within the same amount of time needed by LIME, and it is computationally less-intensive than LORE. The principal advantage of the execution efficiency is the feasibility of deriving global explanations from local explanations, especially for large-scale data sets. Explanation size is the number of predicates in an explanation rule e. It depends on the dimensionality of feature space, the complexity of the decision function, and the characteristics of the created neighborhood. The average length of explanations in EXPLAN is 3.08 ± 0.4, while it is 1.78 ± 0.4 and 2.35 ± 0.5 for LORE and Anchor, respectively. LIME, on average, explains a particular instance with 8.38 ± 0.8 features. There is a trade-off between comprehensibility and interpretability of explanations. Comprehensibility indicates informative and semantic explanations, while interpretability refers to simple and understandable explanations. As a result, comprehensible explanations with reasonable length are favored. It is noteworthy that having a reliable measurement of the comprehensibility demands the corresponding domain knowledge or domain expert.

TABLE VI
COMPARISON OF JACCARD MEASURE OF STABILITY.

Data set	Black-box	EXPLAN	LORE	Anchor
Adult	**GB**	0.827±.1	0.821±.1	0.755±.1
	LR	0.859±.1	0.799±.1	0.671±.1
	NN	0.856±.1	0.728±.1	0.744±.2
German	**GB**	0.702±.1	0.694±.2	0.754±.1
	LR	0.729±.1	0.698±.2	0.819±.2
	NN	0.846±.1	0.779±.1	0.884±.1
COMPAS	**GB**	0.888±.1	0.858±.2	0.859±.1
	LR	0.886±.1	0.859±.2	0.854±.1
	NN	0.848±.1	0.807±.2	0.822±.1

VI. CONCLUSIONS AND FUTURE WORKS

In this paper, we introduced EXPLAN, a novel local model-agnostic explanation method for tabular classifiers. The main feature of the proposed algorithm is an adaptive neighborhood generation mechanism that defines an appropriate locality for the instance of interest. Compared to the baseline approaches, EXPLAN is a computationally efficient explanation method with significant fidelity, precision, and stability properties. In future work, we will focus our efforts on qualitative analysis of the explanations by incorporating domain knowledge into our methodology. Furthermore, we aim to validate the robustness of the black-box model utilizing the explanations.

REFERENCES

[1] Mohammad Reza Abbasifard, Bijan Ghahremani, and Hassan Naderi, "A survey on nearest neighbor search methods," International Journal of Computer Applications, vol. 95(25), 2014.

[2] Herv e Abdi, Coefficient of variation, Encyclopedia of research design, vol. 1, pp. 169–171, 2010.

[3] Amina Adadi and Mohammed Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," IEEE Access, vol. 6, pp. 52138–52160, 2018.

[4] Osbert Bastani, Carolyn Kim, and Hamsa Bastani, "Interpretability via model extraction," arXiv preprint arXiv:1706.09773, 2017.

[5] Dankmar Bohning, "Multinomial Logistic Regression Algorithm," Ann. Inst. Stat. Math., vol. 44(1), pp. 197–200, 1992.

[6] Leo Breiman, "Random Forests," Mach. Learn., vol. 45, pp. 5–32, 2001.

[7] Richard G Brereton and Gavin R Lloyd, "Support vector machines for classification and regression," Analyst, vol. 135(2), pp. 230–267, 2010.

[8] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," Journal of artificial intelligence research, vol. 16, pp. 321–357, 2002.

[9] Finale Doshi-Velez and Been Kim, "Towards a rigorous science of interpretable machine learning," arXiv preprint arXiv:1702.08608, 2017.

[10] James Dougherty, Ron Kohavi, and Mehran Sahami, "Supervised and unsupervised discretization of continuous features," in Machine Learning Proceedings 1995, pp. 194–202, Elsevier, 1995.

[11] Jerome H Friedman, "Greedy Function Approximation: A Gradient-Boosting Machine," Ann. Stat., vol. 29(5), pp. 1189–1232, 2001.

[12] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael-Specter, and Lalana Kagal, "Explaining explanations: An overview of interpretability of machine learning," in 5th International Conference on data science and advanced analytics (DSAA), pp. 80–89. IEEE, 2018.

[13] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti, "Local rule-based explana-tions of black box decision systems," arXiv preprint arXiv:1805.10820, 2018.

[14] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi, "A survey of methods for explaining black box models," ACM computing surveys (CSUR), vol. 51(5), pp. 93, 2019.

[15] Jiawei. Han, Micheline. Kamber, and Jian. Pei, Data mining : concepts and techniques, Elsevier Science, 2011.

[16] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec, "Interpretable explorable approximations of black box models," arXiv preprint arXiv:1707.01154, 2017.

[17] Benjamin Letham, Cynthia Rudin, Tyler H Mccormick, and David Madigan, "Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model," The Annals of Applied Statistics, vol. 9(3), pp. 1350–1371, 2015.

[18] Scott M Lundberg and Su-In Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems, pp. 4765–4774, 2017.

[19] Christoph Molnar, Interpretable Machine Learning: A Guide for Making Black Box Models Explainable, 2019.

[20] Fionn Murtagh and Pierre Legendre, "Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion?," Journal of Classification, vol. 31(3), pp. 274–295, October 2014.

[21] Eliana Pastor and Elena Baralis, "Explaining black box models by means of local rules," in Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pp. 510–517. ACM, 2019.

[22] F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, M Blondel, P Prettenhofer, R Weiss, V Dubourg, J Vanderplas, A Passos, D Cournapeau, M Brucher, M Perrot, and E Duchesnay, "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[23] Junfei Qiu, Qihui Wu, Guoru Ding, Yuhua Xu, and Shuo Feng, "A survey of machine learning for big data processing," EURASIP Journal on Advances in Signal Processing, vol. 67(2016), May 2016.

[24] Peyman Rasouli and Ingrid Chieh Yu, "Meaningful Data Sampling for a Faithful Local Explanation Method," in 20th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2019), LNCS, vol. 11871, pp. 28–38. Springer, 2019.

[25] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, ""Why Should I Trust You?" Explaining the Predictions of Any Classifier," KDD '16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144, 2016.

[26] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, "Anchors: High-precision model-agnostic explanations," in Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[27] Salvatore Ruggieri, "Yadt: Yet another decision tree builder," in 16th IEEE International Conference on Tools with Artificial Intelligence, pp.260–265. IEEE, 2004.

[28] Ando Saabas. Interpreting Random Forests, Available at: http://blog.datadive.net/interpreting-random-forests/, [Last Accessed 15 Nov 2019], 2014.

[29] Sarah Tan, Rich Caruana, Giles Hooker, Paul Koch, and Albert Gordo, "Learning global additive explanations for neural nets using model distillation," arXiv preprint arXiv:1801.08640, 2018.

[30] Sarah Tan, Rich Caruana, Giles Hooker, and Yin Lou, "Detecting bias in black-box models using transparent model distillation," arXiv preprintarXiv:1710.06169, 2017.

[31] Sarah Tan, Rich Caruana, Giles Hooker, and Yin Lou, "Distill-and-compare: auditing black-box models using transparent model distillation," in Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, pp. 303–310. ACM, 2018.

[32] Jayaraman J Thiagarajan, Bhavya Kailkhura, Prasanna Sattigeri,and Karthikeyan Natesan Ramamurthy, "Treeview: Peeking into deep neural networks via feature-space partitioning," arXiv preprint-arXiv:1611.07429, 2016.

[33] Erico Tjoa and Cuntai Guan, "A survey on explainable artificial intelligence (xai): Towards medical xai," arXiv preprint arXiv:1907.07374, 2019.

[34] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-SNE," J. Mach. Learn. Res., vol. 9(Nov), pp. 2579–2605, 2008.

[35] Fulton Wang and Cynthia Rudin, "Falling rule lists," in Artificial Intelligence and Statistics, pp. 1013–1022, 2015.

[36] Simon N Wood, Generalized additive models: an introduction with R, Chapman and Hall/CRC, 2017.

[37] Guoqiang Peter Zhang, "Neural networks for classification: A survey," IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, vol. 30(4), pp. 451–462, November 2000.

[38] Qingchen Zhang, Laurence T Yang, Zhikui Chen, and Peng Li, "A survey on deep learning for big data," Information Fusion, vol. 42, pp. 146–157, 2018.

[39] Dattagupta, Samrat Jayanta. A performance comparison of oversampling methods for data generation in imbalanced learning tasks. Diss., 2018.