# PDTX: A Novel Local Explainer Based on the Perceptron Decision Tree

Samara Silva Santos
*Graduate Program in Electrical Engineering*
*Universidade Federal de Minas Gerais*
*Av. Antônio Carlos 6627, 31270-901*
Belo Horizonte, MG, Brazil
samarass@ufmg.br

Marcos Antonio Alves
*Graduate Program in Electrical Engineering*
*Universidade Federal de Minas Gerais*
*Av. Antônio Carlos 6627, 31270-901*
Belo Horizonte, MG, Brazil
marcosalves@ufmg.br

Leonardo Augusto Ferreira
*Graduate Program in Electrical Engineering*
*Universidade Federal de Minas Gerais*
*Av. Antônio Carlos 6627, 31270-901*
Belo Horizonte, MG, Brazil
leauferreira@cpdee.ufmg.com

Frederico Gadelha Guimarães
*Machine Intelligence and Data Science Laboratory (MINDS)*
*Department of Electrical Engineering*
*Universidade Federal de Minas Gerais*
31270-901, Belo Horizonte, Brazil
fredericoguimaraes@ufmg.br

***Abstract*—Artificial Intelligence (AI) approaches that achieve good results and generalization are often opaque models and the decision-maker has no clear explanation about the final classification. As a result, there is an increasing demand for Explainable AI (XAI) models, whose main goal is to provide understandable solutions for human beings and to elucidate the relationship between the features and the black-box model. In this paper, we introduce a novel explainer method, named PDTX, based on the Perceptron Decision Tree (PDT). The evolutionary algorithm jSO is employed to search for the weights of the PDT to approximate the predictions of the black-box model with high fidelity. The PDTX was tested in 10 different datasets from a public repository as an explainer for three classifiers: Multi-Layer Perceptron, Random Forest and Support Vector Machines. Decision-Tree and LIME were used as baselines. The results showed remarkable performance in the majority of the experiments, achieving $87.34\%$ of average accuracy, against $64.23\%$ and $37.44\%$ from DT and LIME, respectively. The PDTX can be used for black-box classifier explanations, for local instances and it is model-agnostic.**

***Index Terms*—Explainable AI, Interpretability, Machine Learning, Local explanations, xAI**

## I. Introduction

Recently, Machine Learning (ML) and Deep Learning (DL) techniques have grown enormously and achieved promising results in the most diverse fields, such as computer vision, speech recognition, robot control, medicine applications, credit card transactions, and others [1], [2]. However, some of these methods are deemed as black-box models, which means that there is no clear relation between its input and output. Consequently, it is not possible to easily understand the decisions made by those systems. Because of this lack of transparency, the demand for approaches that make these methods more comprehensible also emerged.

In [3], transparency is understood when the model that extracts parameters from the training data and generates labels for the test can be described and motivated by the model design. In other words, transparency is the way used by expert systems to explain how that result was achieved. Following the idea, interpretability is related to some properties of an ML model that make it understandable to humans. However, there is a gap between accuracy and transparency since high-precision techniques are usually opaque. A known strategy is to approximate the predictions of a black-box model by an interpretable one, such as a Decision Tree (DT).

In some fields, errors can lead to critical consequences. In medicine, for example, the utilization of some ML/DL approaches is still sensitive because the decisions might affect people's lives and health. Hence, the clinicians must be able to understand why the model made such a prediction [4]. Besides that, there is also a movement of the governments in the creation of new rules and laws to protect user's data and measure the impacts and consequences of AI-based decisions. The European Union, for example, promulgated in 2019 the European Union's General Data Protection Regulation (GDPR), which defines the "*right to explanations*" for its members. The definition in the art. 22 and in Recital 71 EU GDPR says: "*The data subject should have the right not to be subject to a decision, which may include a measure, evaluating personal aspects relating to him or her which is based solely on automated processing and which produces legal effects*

S.S. Santos would like to thank the Federal Center for Technological Education of Minas Gerais (CEFET-MG). M.A. Alves declares that this work has been supported by the Brazilian agency CAPES. Also, this work has been supported by the Brazilian agencies (i) National Council for Scientific and Technological Development (CNPq); (ii) Coordination for the Improvement of Higher Education (CAPES) and (iii) Foundation for Research of the State of Minas Gerais (FAPEMIG, in Portuguese).

MINDS Laboratory – https://minds.eng.ufmg.br/

*concerning him or her or similarly significantly affects him or her, such as automatic refusal of an online credit application or e-recruiting practices without any human intervention*" [5].

Thus, explainability is a hot topic. It presents fundamental relevance for the scientific community since the results of the AI-based approaches become more understandable to analysts and experts. For this reason, the outputs of these systems still lack the utilization of communicable representations through mathematical, logical, linguistic, or visual resources. Therefore, the eXplainable AI (XAI) aims to develop more explainable models and with high accuracy levels and, at the same time, allowing humans not just to understand, but also to trust and manage these kinds of systems [6]. As pointed out by Rudin [7], the best strategy is to elaborate methods that are transparent for default. Nevertheless, although some ML methods are mathematically interpretable, such as Decision Tree and Logistic Regression [8] in some kind of data most of them are not efficient as the black-box ones. Furthermore, building new methods that are transparent in their formulation and outputs and also able to provide high accuracy is still a hard task. On the other side, some authors are collaborating with new interpreters (also called explainers) such as: (i) Local Interpretable Model-Agnostic Explanations (LIME) [9], SHapley Additive exPlanations (SHAP) [10] and Genetic Programming Explainer (GPX) [11] to cite a few.

In this scenario, this paper presents a new method for the local explanations named Perceptron Decision Tree Explainer (PDTX). It is based on the Perceptron Decision Tree (also known as PDT or Oblique Decision Tree) [12]. PDT divides the features space by considering combinations of the attribute values, whether linear or nonlinear. Since that each internal node evaluates a linear combination of the attributes, they are equivalent to hyperplanes at an oblique orientation to the axes [13]. Furthermore, the tree-form representation of a PDT naturally yields an analytic expression that gives a local explanation of the proportional contribution of each feature to the prediction. Therefore, using this approach, we can approximate the predictions of black-box models and also providing interpretability for classification problems with structured data.

To do so, this paper is organized as follows. Section II presents the XAI terminologies and briefly discuss some existing interpretable models. Section III present the process that are part of the proposed PDTX and how it works. Section IV presents the experimental setup used to validate the PDTX. Section V presents the results, some discussion as well as future extensions/applications. Section VI concludes the paper.

## II. Explainable AI

### A. The black-box problem

A black-box method can be addressed as one in which the input and output are known, however, the internal process is not available or, when it is, it has high complexity to be understood. In this sense, even if the weights and parameters are accessible, it is not possible to obtain a clear relationship between the features and the result. Consider a neural network, for instance: the knowledge of all the weights of the model does not help explaining why a given decision was made. In this way, the XAI provides directions to attend to this demand.

### B. Terminologies

The literature has presented a set of terminologies associated with the XAI field, some of them consolidated in the works of [4], [6], [11], [14].

- **Explainability:** it is related to the notion of explanation as an interface between humans and a decision-maker that is both an accurate proxy of the decision-maker and understandable to humans.
- **Interpretability:** is to be able to explain or provide meaning in a fashion understandable to a human being. Or still, the degree to which a person can understand the reason for an outcome.
- **Understandability:** it refers to the human knowledge of system functions without explanations about its intern functionalities.
- **Comprehensibility:** it is the ability of a learning system to show its learned knowledge in a human intelligible way.
- **Transparency:** a model is transparent if it is self-understandable. It regards the necessity to describe, inspect and reproduce the mechanisms through which AI systems make decisions and learn to adjust to their environment and the governance of the data used created.
- **Complexity:** is the level of effort required by a user to understand an explanation taking into account the user's training or any time constraints required for understanding.
- **Responsible AI:** is AI that is mindful of social values and moral and ethical concerns.

### C. Interpretable Models

Some ML models are inherently transparent and interpretable, such as DT, Linear Regression, Logistic Regression, and Naive Bayes. Due to this, there is a mathematical and/or logical explanation about why certain results were achieved. A model is better interpretable if its decisions are easier for a human being to understand than other models [15]. Despite that, black-box models, especially those related to DL, are more accurate for some specific tasks as image recognition.

In that context, the interpretations can be addressed in two ways: explanation by design and black-box explanation [16]. The former is about building solutions that already produce explanations in decision-making and, the latter is to produce explanations from the ready-made black-box system.

Thereby, explainability methods can be roughly split into global and local explanations [4], [14]. Global ones make it easy to understand the entire logic of a model and follow the reasoning that leads to the different solutions. In contrast, explaining the reasons for a particular decision or a single case implies that interpretability is occurring over only this sample. This case is invoked to produce an individual explanation, usually to justify why the model took a specific decision. It

is argued that analyzing all the solutions is difficult to reach in practice, especially for models with many parameters [14]. Local interpretability may be easier to understand by humans, since understanding the context of just one sample is easier than an entire set. Additionally, there is another classification that can be applied to explanation models, according to their applicability. It is named model-agnostic whether can be used in any black-box ML model or model-specific, if it can be applied only for a single type or class of algorithm.

Also, fidelity may be used as a measure of quality. According to [15], it means how well the explanation approximates the prediction of the black-box model. Thus, high fidelity is preferable since an explanation with low fidelity is essentially useless.

In this work, we can classify the PDTX into the three categories aforementioned: (i) used for black-box explanation, for local instances, and model-agnostic, since it can be extended for any kind of black-box classification model.

## III. Evolving Explanations

In this section, we present the entire process that is part of the proposed PDTX. In Section III-A we explain the fundamentals of the PDT, and how it works in Section III-B through the approximation with the outputs of the black-box ML models. And finally in Section III-C the extracted information that composes the PDTX.

### A. Perceptron Decision Tree

In DT, each internal node indicates a test on an attribute and the leaf nodes correspond to a class, and in this way, the tests associated with each node are equivalent to axis-parallel hyperplanes in an input space [13]. In a PDT, the internal nodes test a linear combination over the attributes that divides the input space by hyperplanes at general positions. Here, we define a PDT as [17] by testing the internal nodes with the equation $wx + \theta = z$, where $w$ is a weight vector, $x$ are the features and $\theta$ is a constant. Thus the search space is divided by hyper-planes in general positions, which are not necessarily axis parallel. To illustrate, an example is presented in Figure 1, with depth equal to 3. The root node tests the inequality $x - y < 0$, where $[1, -1]$ is the $w$ vector, $\theta = 0$, and they are the coefficients of the linear equation that defines the first partition of the search space.

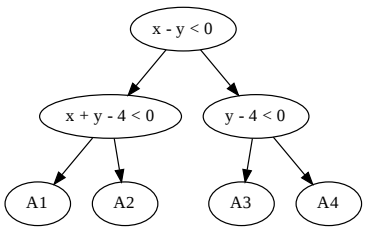


Fig. 1. Illustration of a PDT of depth equal to 3

The corresponding areas of this tree are then presented in Figure 2. The region where the inequality $x - y < 0$ is $True$ lies above the diagonal, covering $A1$ and $A2$. The region where the inequality is $False$ covers both $A3$ and $A4$. Again, the space is divided for $x + y - 4 < 0$ and $y - 4 < 0$ to obtain the other partitions. Following this idea, it is easy to visualize that a PDT can partition the search space recursively, splitting the space into the corresponding classes.

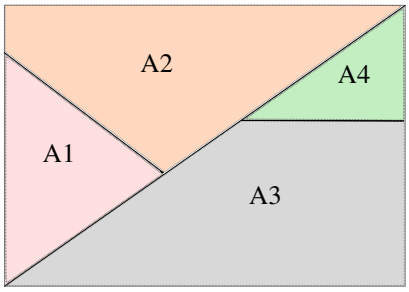


Fig. 2. Example of the partition made by PDT of height 3.

In this work, similar to [17], each PDT is a complete binary tree with linear classifiers in the internal nodes, according to the previous definition, being $w$ a vector of length $n$ and the constant $\theta$. $n$ is the number of attributes of the classification problem. PDT has also $2^{h-1} - 1$ leaf nodes, being $h$ the depth. The learning process occurs in a recursive manner beginning in the root node where the linear classifier is applied. In the case of whether it is considered a matrix structure to represent the binary tree, after using the linear classifier at position $i$ of $n$, the classifier at position $2i$ is used if the output is lower than 0 (left node) or the classifier $2i + 1$ is used otherwise (right node). The space of the $w$ and $\theta$ is continuous and that of the other is discrete.

To induce PDTs and to produce the explainer, it is necessary to find the values of $w$ and $\theta$ that divide the feature space in the best way. Heuristic techniques may be applied to minimize the classification errors. In this paper, Differential Evolution (DE) algorithm, a population-based evolutionary algorithm, was used. It starts with a random population defined in the search space and for each individual $p_i$ generates a mutant vector $m_i$ which is used to create a new individual from a crossover operation with $p_i$. The new individual is compared with the current one, and, if it is better, it is replaced for it. The original DE was introduced in 1995, and up to now many works have used either the classical method or improved versions. One of them is the jSO approach [18] that presents an automatic adaptation of the parameters showing promising results on solving numerical unconstrained optimization problems.

In this work, where each individual of the population is a PDT, we applied jSO to determine the weights and the constant values of each PDT and, then to approximate the PDT outputs to the predictions of the black-box classifier. For the leaf nodes, a random configuration is applied from the possible values of classes that the problem has. Also, local search has been used to achieve the best value of the leaves

in some elements in each generation.

### B. PDT for Local Explanations

This section describes how the PDTX was used to obtain local interpretability for ML models. For sake of simplicity, we also present an example to clarify the manner that the problem was solved.

The problem was modeled similar to [11] as follows: Let $x \in \mathbb{R}$ an input to a pre-trained black-box model, $k$ points are generated in the vicinity of $x$, or noise set $\eta$, from a multivariate Gauss distribution centered at $x$ with covariance matrix, $\sum = I_n \times \sigma$, where $I_n$ is the squared identity of order $n$ and $\sigma$ is measured on the training data.

PDTX aims to obtain a function that simulates the behavior of the original ML model. It was done by minimizing the prediction error between the output obtained by the interpreter and that generated by the ML model. This concept is called fidelity, and it is defined below:

$$w^*, \theta^* = \arg\min \sum_{s_i \in \eta} err(s_i), \tag{1}$$

where

$$err(s_i) = \begin{cases} 1 & \text{if } f(s_i)) \neq g(s_i) \\ 0 & \text{if } f(s_i)) = g(s_i) \end{cases} \tag{2}$$

and $\{s_1, s_2, ..., s_k\}$ are the samples from the noise set $\eta$, $f(s_i)$ the prediction generated by a given individual in the population and $g(s_i)$ that given by the black-box ML model for a $s_i$ sample.

Consider a classification problem in which there are $1,000$ samples in training data divided into two classes. Class $0$ are represented by the green points and Class $1$ the black ones, as illustrated in Figure 3.

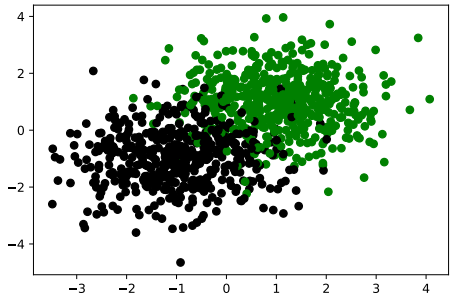

Fig. 3. Data generated with the *2dnormals* function from mlbench representing a 2-dimensional Gaussian problem [19]

Being SVM the $g(.)$ black-box model previously trained with training data, a noise set $\eta$ with $200$ samples is built from a single sample $x$ from the testing data, as showed in Figure 4. For better visualization of the results, the $x$ (red point) was chosen between the two classes.

Figure 5 shows a single PDT that represents the best individual from jSO. This experimental result using the SVM achieved $99\%$ of accuracy (fidelity). It means that the PDT has successfully approximated the predictions of the black-box model.

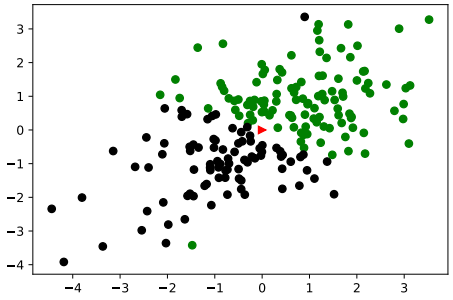

Fig. 4. Noise set generated in the vicinity of the (red) sample

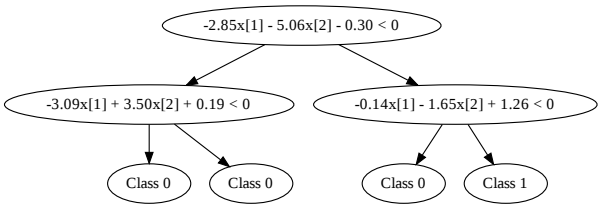


Fig. 5. PDT generated from the jSO showing the sample partitions in the search space

The separation of the classes is presented in Figure 6. The root node divided the data points with the red line straight. Note that this single line is clear enough to divide well the two classes. Next, the blue straight line divides the areas colored in green and blue, and the orange one divides the classes in yellow and red. With these 3 partitions, it is clear that a better classification was provided. It is also worth mentioning that it is possible to both increase the depth of the tree in search of a higher classification rate (Figures 5 and 6) and also to have other trees that may present the same accuracy.

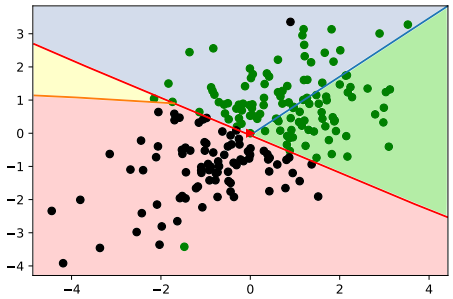

Fig. 6. Separation of classes made by PDT

The deeper the PDT is, the more divisions are going to

be made in the search space and, consequently, the greater the separability among the samples. It works well when the data is not linearly separable. However, if more divisions are made than necessary, the pruning is useful and some nodes may be discarded. In Figure 6, the division made by the blue straight line is unnecessary. For cases such as this one, the pruning operation is performed, and the correspondent node is removed at the explaining step.

### C. Fundamentals and Operation of the PDTX

After the approximation of PDT with the noise set, the local explanation for a specific sample can be reached with the steps described below.

1) Obtain the path (inequalities) followed to classify the input sample;
2) Apply the pruning operation to discard the unnecessary inequalities;
3) Over the remaining inequalities, calculate the hyper-planes that separate the samples;
4) Calculate the feature importance from the coefficients of the hyper-plane closest to $x$. This hyper-plane allows the user to compute the absolute value of partial derivatives of it and evaluate the local importance of each feature.

With this information, it is possible to provide an important local explanation for the problem. The rules extracted from the PDT when all features are kept fixed and only one is changed are shown in the tree of Figure 7 and Figure 8. Figure 7 is obtained by identifying all possible paths of the tree to each leaf, when all attributes, except one, are kept fixed.

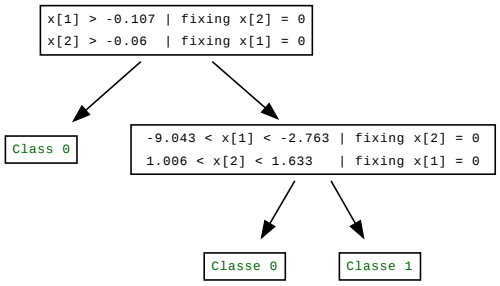


Fig. 7. Rules extracted from PDT

The PDTX output has the following information:

1) Local fidelity of the generated model;
2) Predicted class by the PDTX for the sample;
3) Real predict probabilities (taken from black-box model);
4) Hyper-planes directly obtained from the PDT;
5) Feature importance regarding predicted class, obtained from coefficients of the closest to $x$ hyper-plane ;
   - The absolute value of the coefficient denotes the importance of the feature for Class prediction;
   - Fixing the all the features, we obtain the down/upper bounds of $x_i$.
6) Another way to analyze a PDT is to identify the feature importance for each split. An evaluation method using average feature values and coefficients of a given split is proposed in [20], which consists of multiplication of the average feature value for coefficients and scaled-down the result to the range of $[-1, 1]$. The larger the result is, the more that feature contributes to the split node as can be seen in the last chart of Figure 8.

Figure 8 shows the PDTX output for this example. The model reached $99\%$ of fidelity, and the classification for $x = (0, 0)$ was Class 0, which is following the black-box predict probabilities. From the hyper-plane, the rules generated to obtain this result were: $x_1$ is greater than $-0.107$ and $x_2$ greater than $-0.06$, when $x_2$ and $x_1$ were kept fixed. Both features had positive importance for the prediction, and the impact of $x_2$ being greater than $x_1$. Since only one hyper-plane was necessary for predicting $x$, the calculation was done only on it, which confirms the bigger importance of $x_2$ over $x_1$.

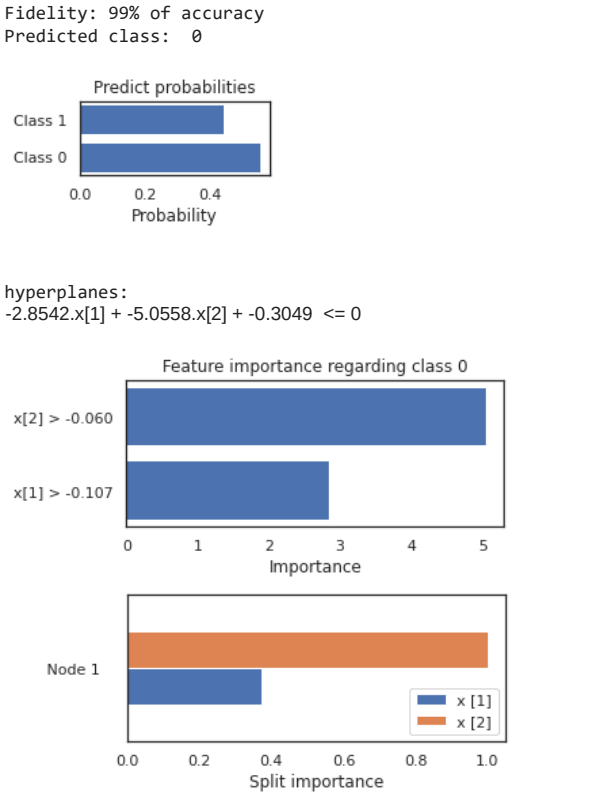


Fig. 8. Explanations taken from PDTX

The local approximation can be verified from small changes around $x$, while the impact in the prediction probability determined for the $g(.)$ was obtained by Sklearn package [21].

In Table I, the first column corresponds to the points in the vicinity of $x$, the probability column contains the black-box

model predictions, and the last column the PDTX predictions. The first row shows that the predicted class was 0 (0.556) corresponding to the same one made by the PDTX. Adding 0.1 to each of the attributes $x_1$ and $x_2$, separately, we obtain an increase in the classification probability for Class 0, described in rows 2 and 3. It is noteworthy that this impacts more $x_2$ than $x_1$, that is, adding 0.1 in $x_1$ the probability of prediction for Class 0 varies from 0.556 to 0.633, while adding 0.1 in $x_2$ increases this probability from 0.556 to 0.688. This effect was expected since the importance of $x_2$ is greater than $x_1$, as shown in Figure 8. By adding 0.1 on both attributes simultaneously, the black-box's probability of ranking to Class 0 increases to 0.753. On the other hand, points outside the region determined in the explanations change the classification to Class 1, as shown in the rows 5 and 6. Lastly, row 7 exemplifies another region that Class 0 is also obtained (which corresponds to the yellow region of Figure 6 and it follows the rules on the right side of the tree of Figure 7). An example of a point in this region is the point $[-3, 1.5]$.

TABLE I
CHECKING THE LOCAL APPROXIMATION

| Vicinity of x | Probability | | PDTX prediction |
|---|---|---|---|
| | Class 0 | Class 1 | |
| $[0, 0]$ | 0.556 | 0.444 | Class 0 |
| $[0.1, 0]$ | 0.633 | 0.367 | Class 0 |
| $[0, 0.1]$ | 0.688 | 0.312 | Class 0 |
| $[0.1, 0.1]$ | 0.753 | 0.247 | Class 0 |
| $[-0.2, 0]$ | 0.395 | 0.605 | Class 1 |
| $[-0.1, 0]$ | 0.416 | 0.584 | Class 1 |
| $[-3, 1.5]$ | 0.576 | 0.424 | Class 0 |

For sake of simplicity, an overview of the PDTX explainer is shown in Figure 9.

## IV. EXPERIMENTAL METHODOLOGY

### A. Datasets

For this work, we used ten datasets from Penn Machine Learning Benchmarks (PMLB) [22] repository, which provides a set of curated benchmark datasets for evaluating and comparing the supervised ML algorithms. This selection was based on the variability of the number of features, samples, and classes.

TABLE II
CLASSIFICATION DATASETS

| **Datasets** | Features | Samples |
|---|---|---|
| Breast | 10 | 699 |
| Car | 6 | 1728 |
| Diabetes | 8 | 768 |
| Ecoli | 7 | 327 |
| Glass | 9 | 205 |
| Hill Valley without noise | 100 | 1212 |
| Iris | 4 | 150 |
| Ionosphere | 34 | 34 |
| Phoneme | 5 | 5404 |
| Wine recognition | 13 | 178 |

### B. Black-box Machine Learning Models

As previously explained, a model is considered a black box when there is no logical/mathematical correlation between input and output. In this paper, we used: Support Vector Machines (SVM), Random Forest (RF), and Artificial Neural Networks (ANN) of the type Multilayer Perceptron (MLP). There is a vast of works in the literature that show promising results with these methods, see for instance [4].

### C. Competing Explainers and Quality Measure

The competing methods are the traditional Decision Tree and LIME [9]. They were chosen for their simplicity and for being widely used in the literature. In DT, each node represents a feature, each branch is a decision, and the leaf nodes are the classes of the classification problem. The path from the root node until the leaf provides an interpretation for the prediction made. Following this path, it is possible to obtain the logical interpretation for the instance analyzed. LIME, in its turn, uses a linear local interpretable model in the vicinity of the predicted instance. This approach is a local surrogate model that generates an explainer worth of resources based on a linear least-squares method with $l_2$-norm regularization [9], [11].

We are going to use the fidelity of the model as a quality measure. Fidelity, in this case, is understood as the accuracy obtained by the approximation method with the black-box, as described in Equation (3). Each explainer has its own approximation method, which is: Linear approximation in the case of LIME, PDT in the case of the proposed PDTX and the DT is itself.

$$acc(f) = \frac{1}{|\eta|} \sum_{s_i \in \eta} h(s_i) \tag{3}$$

where,

$$h(s_i) = \begin{cases} 1 & \text{if } f(s_i)) = g(s_i) \\ 0 & \text{if } f(s_i)) \neq g(s_i) \end{cases} \tag{4}$$

where $h$ represents the result of the prediction from the approximation method.

The process for evaluating each method is described in Section IV-D. The steps are done for the PDT and competing methods, which aim to approximate the black-box methods.

### D. Experimental setup

Here is presented the experimental setup used to compare the proposed explainer PDTX, with the competing DT and LIME. The assumed configurations are listed below:

1) The aforementioned datasets are normalized (z-score) and divided into training and test data, in the proportion of 80% and 20%, respectively.
2) The black-box models are employed in the training data;
3) A random sample (to be explained) is selected from the test set. A noise set is created in the vicinity of this instance and the predictions from the black-box model are obtained.

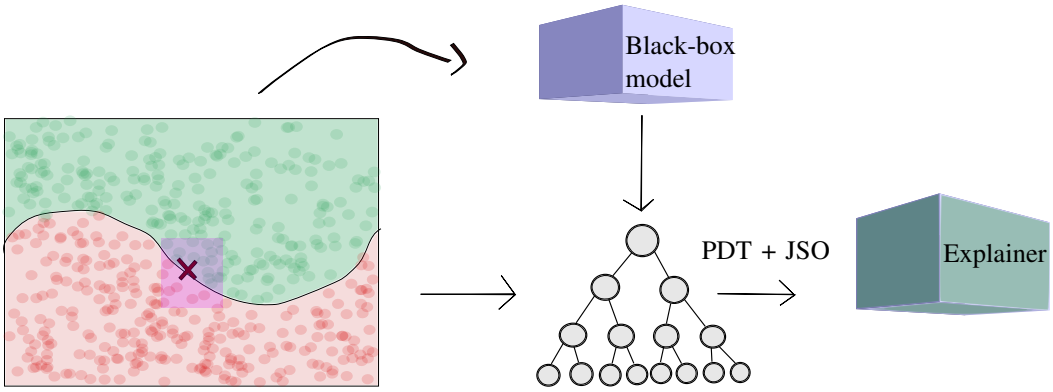


Fig. 9. Overview of the proposed PDTX considering the (i) generation of a vicinity around a noise sample to be explained, (ii) to obtain the predictions from the ML black-box model for the noise set, and (iii) approximation of the PDT predictions with the jSO. With this tree, it is possible to explain the test case.

4) Apply the explainer methods, which are PDTX, LIME, and DT, to get their predictions. Since we are using 10 datasets, 100 samples, 3 ML black-box models, and 3 explaining methods, we have nine thousand predictions.

The noise set was generated using Numpy library [23], centered in a sample selected randomly from the test set and covariance calculated in the training set. For the black-box models, it was used the Sklearn [21], with the following setup: (i) Random Forest: number of trees in the forest equals 100; Iii) SVM: "sigmoid" kernel; (iii) ANN: max_iter equal 3000. (iv) Default values for another parameters.

Finally, the experimental setup applied to PDTX and the competing methods was: (i) Stop criteria: Maximum number of objective function evaluated equal 2000; (ii) Population size: 150 individuals; (iii) Depth of tree: 5; (iv) To define the tree weights by JSO: adaptative rate of mutation and crossover, as defined in JSO work [18]; (v) To define the tree leaves: random configuration with mutation rate chosen randomly between 0.6 and 1 and local search for 10% of individuals in each generation; (vi) Search interval: $[-100, 100]$.

Also, for LIME and DT, the rest of the configurations were kept the default values. For statistical analysis, each experiment was run 30 times.

## V. Results and Discussion

The PDTX was employed in the 10 datasets previously mentioned and the average accuracy for the approximation are summarized in Table III.

As can be observed, the PDTX obtained the best results for the majority of the experiments regarding all the black-box methods (MLP, RF, and SVM). To verify if the observed differences are statistically significant, the Wilcoxon Signed-Rank test [24] was performed and the p-values described in Table IV confirm the differences.

The good performance of the PDTX may be due, mainly, to the linear combination of all attributes. The DT, for instance, takes into consideration only the most important feature per node, based on a cost function (usually entropy), to do the split. This method has simplicity in a logical interpretation,

TABLE III
Average accuracy for the approximation of the proposed PDTX and the competing methods (DT and LIME)

| Model | Dataset | PDTX | DT | LIME |
|---|---|---|---|---|
| MLP | Breast | **0.979 ± 0.018** | 0.500±0.141 | 0.336±0.260 |
| MLP | Car | **0.857 ± 0.082** | 0.794±0.103 | 0.044±0.044 |
| MLP | Diabetes | **0.931 ± 0.045** | 0.572±0.095 | 0.343±0.246 |
| MLP | Ecoli | **0.843 ± 0.077** | 0.655±0.148 | 0.278±0.249 |
| MLP | Glass | **0.713 ± 0.078** | 0.424±0.148 | 0.495±0.195 |
| MLP | Iris | **0.907 ± 0.051** | 0.872±0.074 | 0.346±0.136 |
| MLP | Wine rec. | **0.893 ± 0.052** | 0.789±0.127 | 0.280±0.211 |
| MLP | Ionosphere | **0.890 ± 0.054** | 0.686±0.165 | 0.538±0.286 |
| MLP | Phoneme | **0.895 ± 0.062** | 0.656±0.106 | 0.232±0.150 |
| MLP | Hill V. | **0.780 ± 0.105** | 0.772±0.099 | 0.577±0.217 |
| MLP Mean | | 0.869 ± 0.076 | 0.672±0.141 | 0.347±0.158 |
| RF | Breast | **0.978 ± 0.014** | 0.834±0.063 | 0.426±0.297 |
| RF | Car | 0.853 ± 0.099 | **0.881 ± 0.061** | 0.029±0.024 |
| RF | Diabetes | **0.918 ± 0.050** | 0.640±0.101 | 0.307±0.218 |
| RF | Ecoli | **0.851 ± 0.097** | 0.729±0.099 | 0.237±0.227 |
| RF | Glass | **0.767 ± 0.114** | 0.677±0.151 | 0.545±0.226 |
| RF | Iris | 0.907 ± 0.049 | **0.924 ± 0.049** | 0.324±0.088 |
| RF | Wine rec. | **0.888 ± 0.049** | 0.816±0.104 | 0.448±0.323 |
| RF | Ionosphere | **0.879 ± 0.083** | 0.743±0.107 | 0.475±0.319 |
| RF | Phoneme | **0.865 ± 0.070** | 0.696±0.077 | 0.281±0.183 |
| RF | Hill V. | 0.759 ± 0.076 | **0.850 ± 0.055** | 0.601±0.163 |
| RF Mean | | 0.867±0, 066 | 0.792±0.095 | 0.367±0.167 |
| SVM | Breast | **0.976 ± 0.019** | 0.484±0.144 | 0.405±0.272 |
| SVM | Car | **0.842 ± 0.089** | 0.539±0.139 | 0.010±0.025 |
| SVM | Diabetes | **0.867 ± 0.052** | 0.527±0.097 | 0.395±0.222 |
| SVM | Ecoli | **0.879 ± 0.086** | 0.339±0.271 | 0.202±0.218 |
| SVM | Glass | **0.787 ± 0.069** | 0.328±0.183 | 0.385±0.226 |
| SVM | Iris | **0.901 ± 0.070** | 0.306±0.270 | 0.285±0.162 |
| SVM | Wine rec. | **1.000 ± 0.000** | 0.541±0.268 | 0.364±0.306 |
| SVM | Ionosphere | **0.887 ± 0.053** | 0.634±0.111 | 0.624±0.241 |
| SVM | Phoneme | **0.881 ± 0.072** | 0.525±0.073 | 0.268±0.207 |
| SVM | Hill V. | **0.830 ± 0.106** | 0.536±0.268 | 0.741±0.273 |
| SVM Mean | | 0.885±0, 064 | 0.475±0.111 | 0.409±0.224 |
| **Overall Mean** | | 0.873±0, 067 | 0.642±0.171 | 0.374±0.181 |

but it can vary depending on the depth of the tree. The higher the number of features, the deeper the DT is. In datasets with many features, the visualization can be affected. In PDT, the depth is fixed. In our empirical experiments, it showed good results with a fixed height of 5. The LIME, in its turn, is well known in the scientific community, easy to apply, and may

TABLE IV
RESULTS FOR THE WILCOXON SIGNED-RANK TEST

| Competing Models | p-value |
|---|---|
| PDTX vs. DT | 6.654e-06 |
| PDTX vs. LIME | 1.824e-06 |
| DT vs. LIME | 8.070e-06 |

be one of the pioneers in the context of local explanation. However, the method requires careful application on complex real data, see for instance the discussion presented in [25].

The good performance of PDTX concerning DT and LIME is expected since the complexity of the former is greater than these both competitors. Thus, the great advantage of the PDTX over them is that it achieves high fidelity to the black-box models while maintaining its explanatory potential.

## VI. CONCLUSION

The massive use of computational intelligence tools in diverse areas of knowledge, together with the new regulations for human rights regarding automatic decisions which can affect our lives, increases the need for explainable methods. It allows the analyst to better understand the black-box models and taking decisions based on these explications.

In this context, we presented a new method for local interpretability focused, at this time, in classification tasks, based on PDT. The PDTX enables the users to collectively analyze the features of the problem. This explainer is model-agnostic, which allows it to be used with any ML method with structured data. Besides that, it combines the advantages of the traditional DT and LIME. It presents a tree structure similar to a flowchart, being easy to read and understand and providing information from the general equation of a hyperplane. Different from global explainer models, local ones aim to rather explain single predictions by interpretable models than the whole black box model at once.

The experimental setup performed resulted in a general measure of fidelity of $87.34\%$ for PDTX, $64.23\%$ for traditional DT, and $37.44\%$ for LIME. This work contributes by providing a new method that has a better approximation to a black-box model and gives robust local explanations. Future works include the extension to regression methods and deep learning methods.

## REFERENCES


[1] K. Das and R. N. Behera, "A survey on machine learning: concept, algorithms and applications," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 5, no. 2, pp. 1301–1309, 2017.

[2] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, "A survey of deep learning and its applications: a new paradigm to machine learning," *Archives of Computational Methods in Engineering*, pp. 1–22, 2019.

[3] R. Roscher, B. Bohn, M. F. Duarte, and J. Garcke, "Explainable machine learning for scientific insights and discoveries," *IEEE Access*, vol. 8, pp. 42 200–42 216, 2020.

[4] M. A. Alves, G. Z. Castro, B. A. S. Oliveira, L. A. Ferreira, J. A. Ramírez, R. Silva, and F. G. Guimarães, "Explaining machine learning based diagnosis of covid-19 from routine blood tests with decision trees and criteria graphs," *Computers in Biology and Medicine*, vol. 132, p. 104335, 2021.

[5] R. Viorescu *et al.*, "2018 reform of eu data protection rules," *European Journal of Law and Public Administration*, vol. 4, no. 2, pp. 27–39, 2017.

[6] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020.

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

[8] M.-Y. Chen, "Predicting corporate financial distress based on integration of decision tree classification and logistic regression," *Expert systems with applications*, vol. 38, no. 9, pp. 11 261–11 272, 2011.

[9] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[10] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *arXiv preprint arXiv:1705.07874*, 2017.

[11] L. A. Ferreira, F. G. Guimarães, and R. Silva, "Applying genetic programming to improve interpretability in machine learning models," in *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2020, pp. 1–8.

[12] E. Cantu-Paz and C. Kamath, "Inducing oblique decision trees with evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 1, pp. 54–68, 2003.

[13] S. K. Murthy, S. Kasif, and S. Salzberg, "A system for induction of oblique decision trees," *Journal of artificial intelligence research*, vol. 2, pp. 1–32, 1994.

[14] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (xai)," *IEEE access*, vol. 6, pp. 52 138–52 160, 2018.

[15] C. Molnar, *Interpretable Machine Learning*, 2019, https://christophm.github.io/interpretable-ml-book/.

[16] D. Pedreschi, F. Giannotti, R. Guidotti, A. Monreale, S. Ruggieri, and F. Turini, "Meaningful explanations of black box ai decision systems," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 9780–9784.

[17] R. A. Lopes, A. Freitas, R. P. Silva, and F. G. Guimarães, "Differential evolution and perceptron decision trees for classification tasks," in *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 2012, pp. 550–557.

[18] J. Brest, M. S. Maučec, and B. Bošković, "Single objective real-parameter optimization: Algorithm jso," in *2017 IEEE congress on evolutionary computation (CEC)*. IEEE, 2017, pp. 1311–1318.

[19] F. Leisch, E. Dimitriadou, M. F. Leisch, and Z. No, "Package 'mlbench'," *CRAN*, 2009.

[20] B. M. Zázvorka, "Application of decision trees to failure detection in hvac systems," 2020.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[22] R. S. Olson, W. La Cava, P. Orzechowski, R. J. Urbanowicz, and J. H. Moore, "Pmlb: a large benchmark suite for machine learning evaluation and comparison," *BioData Mining*, vol. 10, no. 1, p. 36, Dec 2017. [Online]. Available: https://doi.org/10.1186/s13040-017-0154-4

[23] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: a structure for efficient numerical computation," *Computing in science & engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[24] R. Woolson, "Wilcoxon signed-rank test," *Wiley encyclopedia of clinical trials*, pp. 1–3, 2007.

[25] C. Molnar, S. Gruber, and P. Kopper, "Limitations of interpretable machine learning methods," 2020.